La técnica del Análisis de la Semántica Latente (LSA/LSI) como modelo informático de la comprensión del texto y el discurso

Una aproximación distribuida al análisis semántico.

Tesis doctoral presentada por

Guillermo de Jorge Botana

Dirigida por el doctor

José Antonio León Cascón

FACULTAD DE PSICOLOGÍA

Departamento Psicología Básica

Programa de Doctorado de Calidad: Comprensión del Texto y del Discurso

Septiembre, 2010

"Los monstruos clásicos: Este título lleno de promesas es el de un libro viejo que hallé al acaso en el taller de un maestro pintor. Sus páginas, ya rancias, reproducen en estampas los monstruos creados por la imaginación de los antiguos. Al hojearlo, yo recordaba cómo en ningún día del mundo pudo el hombre deducir de su mente una sola forma que antes no estuviese ante sus ojos. Puso el asirio las alas del pájaro en el lomo del toro, y el heleno pobló de centauros los bosques mitológicos. Combinaron formas, pero ninguno las creó. La observación es vieja y solamente la saco a memoria para hacer más claro mi pensamiento y llegar a decir cómo algo semejante ocurre con las palabras. El poeta las combina, las ensambla, y con elementos conocidos inventa también un linaje de monstruos: El suyo. Logra así despertar emociones dormidas, pero crearlas nunca."

[La lámpara maravillosa, Ramón Mª del Valle-Inclán]

Puesto que no existe el hombre
sobran los versos humanos
tomemos pues las medidas
fabriquemos aparatos
de palabras, al margen
de ideas y sentimientos
.
.
.
con, de, si, tras, cada, todo
lero, luego, lará, menos.
agite usted la caja de sonidos
y verá cómo acaba por hallarles un sentido.

[Lírica de cámara, Gabriel Celaya]

Agradecimientos:

En primer lugar quiero agradecerle a mi director de tesis, José Antonio León, el gran apoyo intelectual y moral que me ha prestado, incluso en los momentos bajos en los que se hacía difícil continuar. Gracias a su capacidad de motivar y de reunir personas interesadas en una misma temática, ha salido esta tesis adelante. En segundo lugar, quisiera también agradecer a Ricardo Olmos Albacete las aportaciones tan valiosas que me ha hecho. Sus consejos y comentarios han sido enormemente útiles y me han allanado mucho mi labor investigadora. Gracias por esas conversaciones mágicas en torno a la hoguera del LSA, en las que incluso salían infinidad de pavesas filosóficas en torno al lenguaje humano. Me gustaría mencionar también a Javier Sainz, tutor del DEA, enemigo del mito en la ciencia de la psicología y buscador de buenos modelos. También a Ramón Lopez-Higes, que hizo la primera revisión crítica de alguno de estos textos, y a Jesús Sanz y José María Prados Atienza, que desde la complutense me apoyaron con la logística de algunos experimentos y simulaciones. También a Yusef Hassan, investigador del CESIC, por enseñarme a dibujar las palabras y a Gary Cook, por cambiar su idioma por el mío. De los docentes que tuve en licenciatura, mencionar la buena labor de Luis Enrique Lopez-Bascuas, Miguel Ángel Gª-Perez, Salvador Urraca, Gª-Hoz y Luis LLavona (me dejo muchos, aunque no multitud). Es su solvencia la que puede hacer de la psicología una disciplina competitiva.

Por último, dedico esta tesis a Anna, por acompañarme durante tantos años en tantas cosas. Gracias por su cariño constante. Y claro está, también a mis padres, Jaime y Mª José.

Capítulo 1

Introducción

Introducción

Una de las disciplinas que ha tenido mayor número de aportaciones en los últimos tiempos ha sido la Ciencia Cognitiva, cuyo andamiaje ha sido construido por saberes aparentemente tan dispares como la lingüística, la biología, la psicología cognitiva, la psicolingüística, la informática o la inteligencia artificial, etc. Desde la psicología cognitiva y la psicolingüística se han aportado multitud de experimentos y datos empíricos, así como también tentativas de modelos que han sido formalizados con las restricciones que imponían estos mismos datos empíricos. Uno de esos modelos es el llamado Análisis de la Semántica Latente (en inglés *Latent Semantic Analysis* y en adelante LSA) y en él ponemos el foco de atención.

El LSA es una herramienta informática que analiza relaciones semánticas entre diferentes unidades lingüísticas de forma completamente automatizada (Landauer y Dumais, 1997). Según sus creadores, el LSA no sólo es una herramienta de análisis semántico, sino que también puede concebirse como una teoría de adquisición del lenguaje (Landauer y Dumais, 1997; Landauer, Foltz y Laham, 1998). El LSA fue originalmente descrito por Deerwester, Dumais, Furnas, Landauer y Harshman (1990) cómo un método de Recuperación de la Información (*Information Retrieval*). Fueron más tarde Landauer y Dumais (1996; 1997) los que concibieron este modelo como un modelo plausible de la adquisición y la representación del conocimiento. Desde ese momento hasta la fecha podemos encontrar múltiples aplicaciones del LSA en todo tipo de líneas teóricas: por ejemplo, ha sido empleada para modelar algunos fenómenos cognitivos (Kintsch,1998; Landauer, 1999; Kintsch, 2001; Kintsch y Bowles, 2002), se ha utilizado en aplicaciones más directas como la corrección de textos en el ámbito académico (Trusso, 2005), como medida de cohesión y coherencia textual (Graesser ,McNamara, Louwerse and Cai, 2004), como emulación de modelos de usuarios potenciales en usabilidad WEB (Blackmon, Polson, Kitajima, y Lewis, 2002; Blackmon y Mandalia, 2004) o como complemento a las ontologías (Cederberg y Widdows, 2003).

En opinión de Landauer (1999) el LSA ofrece un modelo computacional que simula correctamente muchos fenómenos que tienen que ver con el uso del lenguaje y que tiene mucho que decir sobre la forma en que el ser humano adquiere el lenguaje. Para este autor, el LSA es una herramienta capaz de "aprender" el significado de las palabras a partir de grandes cantidades de lenguaje de manera similar a como lo hacen los humanos. Según Landauer y Dumais (1997), los mecanismos que subyacen al LSA explicarían la adquisición del lenguaje desde una óptica conexionista, argumentando que un sistema como la mente humana capta las micro-contingencias de los datos que se nos presentan y retiran el ruido de las propiedades superfluas. De esta manera, fenómenos empíricos como la coincidencia, co-ocurrencia, contingencia o correlación de hechos sirven de base para explicar el aprendizaje del significado. De alguna forma, estos dos aspectos, captación de micro-contingencias y eliminación de superficialidades, configuran el núcleo de la teoría computacional en la que se basa el LSA. Tanto es así que Landauer y Dumais han propuesto al LSA y las técnicas análogas como manera de fundamentar y resolver la enorme paradoja de "el problema platónico"; es decir, de cómo las personas tenemos más conocimiento del que se podría extraer de la información a la que hemos sido expuestos. O en términos platónicos, "¿Cómo el esclavo puede llegar razonar sin haber sido expuesto a problemas ni información similar?". La solución es porque la arquitectura funcional de los sistemas como el LSA (o la mente), no están dotados de conocimiento innato, sino de mecanismos que inducen el conocimiento general a partir de su entorno. Este conocimiento se extrae indirectamente de las coocurrencias locales de los datos y en el caso del LSA a partir de las microrelaciones de los términos en un corpus increíblemente grande de lenguaje. De esta manera, el LSA, como los niños, suele aprender por instrucción directa mucha menos información que la que puedan llegar a extraer por estos mecanismos de inducción indirecta (Landauer, 2002).

Otra de las virtudes del LSA en cuanto a modelo es que emula el fenómeno de que los conceptos y representaciones mentales no son algo estático e invariable, sino que están en continua interacción con los demás contenidos de la mente y dependen de ellos para su concreción. El

conocimiento no es permanente. El significado de algo viene representado por la porción de la red del conocimiento que está activada en ese instante haciéndolo flexible, intercambiable y temporal. Si algún elemento de la red de conocimiento está activado en la memoria de trabajo (el foco de atención más o menos consciente), otros elementos directamente conectados con él, pueden también ser recuperados (Kintsch, 1998). En el lenguaje natural se producen multitud de fenómenos de estas características: tropos como la metáfora y la metonimia, la ya mencionada homonimia, la sinonimia, procesos de categorización y conceptualización, comprensión de predicaciones, etc. Estos fenómenos y el reconocimiento de las estructuras que se ven involucrados en ellos, requieren de modelos que sean sensibles a la información que aporta el contexto en el que se han extraído. Estos y otros ejemplos ponen de manifiesto que la apuesta teórica que hacen del LSA es fuerte y controvertida. El fuerte de la línea argumentativa de Landauer y Dumais (1997) es sin duda las evidencias empíricas que presentan y los datos que avalan el potencial del LSA. No hay ningún género de dudas de que el LSA resuelve tareas complejas. La cuestión debe centrarse entonces en si el LSA es *fuerza bruta*, un analizador de un colosal saco de términos o si, por el contrario, hay argumentos de peso de que hay algo más depurado y fino tras la herramienta. Para ello conviene saber cómo funciona, *grosso modo,* el LSA.

El LSA comienza procesando un texto de grandes dimensiones. Este texto contiene miles e, incluso, millones de párrafos (o frases). Este texto constituye lo que se conoce como el *corpus lingüístico*. El corpus se representa en una matriz cuyas filas contiene todos los términos distintos del corpus (palabras) y las columnas representen una ventana contextual en la que aparecen esos términos (habitualmente párrafos) (ver figura 1). De este modo, la matriz contiene sencillamente el número de veces que cada término aparece en un documento. Sobre esta matriz de frecuencias se efectúa una ponderación con el objeto de restar importancia a las palabras excesivamente frecuentes y aumentarla a las palabras moderadamente infrecuentes (Nakov, Popova, Mateev, 2001). La razón de esta ponderación es sencilla: las palabras demasiado frecuentes no sirven para discriminar bien la información importante del párrafo y las moderadamente infrecuentes sí. El siguiente paso es someter

esta matriz ponderada a un algoritmo llamado *Descomposición en Valores Singulares* (SVD en adelante por sus siglas en inglés: *Singular Value Decomposition*). SVD es una técnica de reducción de dimensiones como es el análisis factorial (figura 1). El SVD se aplica con la idea de reducir el número de dimensiones de la matriz original en un número mucho más manejable (en torno a 300), sin que se pierda la información sustancial de la matriz original. Lo interesante de esta reducción de dimensiones no es únicamente mejorar el manejo de una matriz tan grande como la original, sino crear un espacio semántico vectorial en el que tanto términos como documentos están representados por medio de vectores que contengan sólo la información sustancial para la formación de conceptos (figura 2). La nueva representación resultante de los términos y documentos en este espacio semántico ha mostrado ser muy exitosa simulando comportamientos humanos (Landauer y Dumais, 1997). La ventaja de representar el lenguaje vectorialmente es que éstos son susceptibles de comparaciones por medio de cosenos, distancias euclídeas u otras medidas de similitud. Además, a partir de las coordenadas de los términos ya representados pueden introducirse en el espacio nuevos vectores que representen textos producidos *a posteriori* y que se suelen llamar pseudodocumentos (Landauer, Foltz y Laham, 1998). Más técnicamente los pseudodocumentos son textos que no aparecen en el corpus lingüístico que el LSA analiza y que se proyectan en el nuevo espacio semántico reducido como un texto más. La ubicación de pseudodocumentos en el espacio semántico latente no precisa de recalcular todo el espacio reducido por medio del SVD. De otra forma sería una técnica inútil. Será a partir de estos textos nuevos (pseudodocumentos) como se lleve a cabo la categorización de términos y textos.

Como se ha dicho, el LSA no surge como un modelo computacional de la mente. Lo hace para solventar las grandes limitaciones que tenían los motores de búsqueda en bases de datos a principios de la década de los 90 del siglo pasado. Entonces, ante la creciente recopilación de documentos digitalizados se crea una enorme demanda de nuevos y eficientes sistemas de búsqueda. Los que entonces hay, es decir, los sistemas tradicionales de recuperación de la información y categorización de los textos, presentaban

serios problemas relacionados con fenómenos del lenguaje natural y la categorización humana (polisemias, lenguaje metafórico, etc). Debido a la carencia de interpretación semántica del contexto, en ocasiones, estos sistemas tradicionales se muestran sumamente ineficientes, ya que recuperan grandes porcentajes de documentos irrelevantes (falsas alarmas) e ignoran documentos relevantes que deberían aparecer como candidatos (Dumais, 2003). Por ejemplo, en el caso de los sinónimos "tumor" y "neoplasma", el sistema al que se le introduce como entrada el término "tumor", puede omitir documentos en los que se encuentre representada la palabra "neoplasma".

Son muchas y variadas las tentativas de solución para este tipo de problemas, todas ellas parcialmente exitosas. Por ejemplo, algunas de ellas se limitan simplemente a reducir la variabilidad del lenguaje lematizando los términos. Otras intentan clonar las relaciones conceptuales que las palabras mantienen creando y marcando artificialmente un vocabulario en forma de red en el que cada palabra-nodo contenga las relaciones jerárquicas, meronímicas, etc., que mantienen con las demás. Esta última orientación está siendo ampliamente desarrollada en el contexto de la *WEB* y tienen la ventaja de que hacen explícitas el tipo de relación que mantienen unas palabras con otras pero tienen como desventaja no tener en cuenta los modelos mentales humanos y los costes que entraña el marcaje "manual" de los componentes. Para crear una ontología hace falta mucho tiempo y la participación de personas que marquen las propiedades de cada clase o palabra. Además, las medidas de similitud que se pueden aplicar a estos modelos son menos versátiles (Dumais, 2003). La técnica de análisis de la semántica latente, además de otras, como los modelos de tópicos (Griffiths & Steyvers, 2004; Steyvers & Griffiths, 2007) representan una buena propuesta para solventar alguna de las desventajas de los métodos citados, a saber, los métodos de lematización, los métodos de búsqueda literal y los modelos de ontologías.

La ventaja de estos modelos es que han mostrado mucha flexibilidad para modelar y simular tareas *muy humanas,* entre las que se cuentan efectos de *priming* semántico, evaluación de textos, establecimiento de coherencia en los textos, la comprensión de metáforas, polisemias, etc (Landauer y Dumais,

1996, Landauer, Foltz y Laham, 1998; Kintsch, 2001; Landauer, 2002). Con el modelo del LSA, enseguida salta a la vista que representa una buena aproximación a un modelo de conocimiento humano capaz de implementarse como sistema de recuperación de información y resolver algunos problemas planteados en el mundo de las tecnologías. Por esta razón este tipo de técnicas ha despertado un incipiente interés en el mundo de la minería de datos o datamining, motores de recuperación de información (Information retrieval), tutores virtuales, sistemas que detectan el plagiarismo, analizadores de estados de ánimo, Sistemas de Reconocimiento de voz (IVR), etc.

El formato de esta tesis consiste en una colección de cuatro manuscritos individuales, que han sido recientemente aceptados o enviados para publicación a revistas internacionales de psicología experimental, lingüística o tecnología. Estos manuscritos son el corazón de esta tesis y representan la parte empírica. Cada manuscrito que puede ser leído independientemente, sin embargo, los cuatro trabajos son complementarios y siguen una lógica argumental. No obstante, la tesis consta de un amplio marco teórico general que sigue un formato convencional, destinado a ofrecer una revisión conceptual más profunda que los capítulos de investigación. En él se profundiza sobre la propia técnica, se presentan las herramientas creadas para dicha investigación y se examinan los casos en donde la técnica se ha aplicado con éxito para resolver problemas prácticos. Además se describen las principales teorías e investigaciones que han planteado el tema del significado y se reflexiona sobre el lugar que ocupa LSA en todas ellas. En una última parte de la tesis se desarrolla una discusión general y un apartado de conclusiones que resumen los logros y limitaciones de los trabajos que se han desarrollado.

Capítulo 2

La técnica

2.1.- Introducción

Como ya comentábamos en la introducción, a principios de la década de los noventa surgieron varios modelos computacionales en el marco de la psicología cognitiva. Uno que ha tenido especial seguimiento por su aplicabilidad es LSA o también llamado LSI (*Latent Semantic Indexing*). La indexación de la semántica latente fue originalmente descrita por Deerwester, Dumais, Furnas, Landauer y Harshman (1990) como un método de recuperación de la información (*Information Retrieval*). Sin embargo, fueron Landauer y Dumais, especialmente en su artículo titulado *A solution to Plato's problem* (1997), los que propusieron al LSA como un modelo plausible de la adquisición y la representación del conocimiento. Posteriormente, fue continuado y perfeccionado por otros autores de manera que acogiese algunos fenómenos empíricos sobre procesamiento del lenguaje (Kintsch, 2001; Landauer, Foltz y Laham, 1998; Quesada, Kintsch y Gomez, 2001; Rehder, Schreiner, Wolfe, Laham, Landauer y Kintsch, 1998; Wolfe, Schreiner, Rehder, Laham, Foltz, Kintsch y Landauer, 1998).

El LSA es un modelo capaz de explicar las relaciones que mantienen unas palabras con otras en diferentes contextos para generar conceptos. Este tipo de relaciones contextuales hacen posible que se pueda llegar a suponer la existencia de bolsas temáticas y redes conceptuales. En otras palabras, LSA aprovecha un fenómeno que se suele cumplirse en el lenguaje natural: las palabras del mismo campo semántico suelen aparecer juntas o en similares contextos (Yu, Cuadrado, Ceglowki y Payne, 2004).

Supongamos que disponemos de una producción verbal con un gran número de frases. Imaginemos también que cuando una frase incluye "mar" también incluye "playa". Este tipo de coincidencias podrían formar un significado emergente y más bien abstracto que significase "sobre la costa" (véase la figura 2.1). Estos significados emergentes podrían generar dimensiones descriptoras de las palabras.

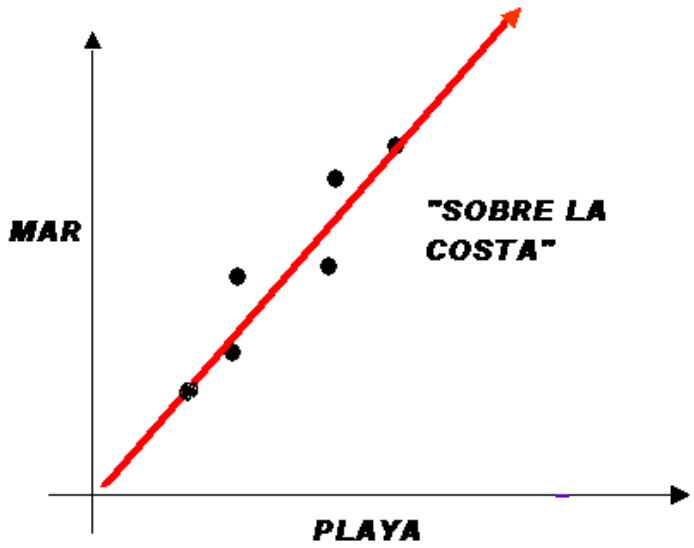


Figura 2.1. La puntuación en las dimensiones “Mar” y “Playa” configura un supuesto grupo temático de mayor abstracción que se podría llamar “Sobre la costa”.

De manera intuitiva, si pudiésemos detectar esta clase de descriptores emergentes podríamos descubrir automáticamente las relaciones entre las palabras. Esta es, precisamente, la base sobre la que se sustenta este tipo de técnicas, la concurrencia de las palabras representadas en documentos, párrafos, frases, páginas WEBs, etc. Sobre esta base, surge la idea tan atractiva como plausible de que palabras que no aparecen en determinados documentos, puedan disponer de una probabilidad *p* de estar en ellos. Esta presencia “virtual” surge por la relación que estas palabras establecen con las que sí se encuentran presentes en los documentos. En otras palabras, si en un documento apareciesen las palabras "playa" y "cangrejo", podría estimarse una cierta probabilidad de que surgiese también la palabra "mar", si esta palabra hubiese aparecido relacionada alguna vez junto a las palabras "cangrejo" o "playa". Toda esta información puede ser susceptible de ser representada matricialmente de manera que se plasmen las ocurrencias de las palabras en los contextos.

Continuando con el ejemplo, tratemos de imaginarnos ahora que las palabras “mar”, “playa”, “cangrejo” y “gaviota” se descubren de un viejo

documento que habla sobre la ría de Vigo y que otras palabras como “meseta”, “encina”, “cereal” y “vino” proceden de una guía sobre Valladolid. Podríamos diseñar una matriz en la que en su parte horizontal se colocasen los documentos referentes a Vigo y también los pertenecientes a la provincia de Valladolid. En la parte vertical se ubicarían los términos que salen de todos estos documentos (“mar”, “encina”, etc.). En cada celdilla consignaríamos las ocurrencias de los términos que aparecen en dichos documentos. Un paso más en la elaboración de esta matriz sería comprobar qué dimensiones de esos términos (los documentos de los que surgen) y qué dimensiones de los documentos (los términos que contienen) son los que tienen más importancia tienen con el objeto de diferenciar a términos y documentos y formar con ellos bolsas o conglomerados semánticos. Es decir, se trataría de buscar las dimensiones que mejor permitan una diferenciación en "bolsas semánticas” en las que los términos participan[1]. Este proceso matricial sería análogo a lo que hace la nariz y la boca del enólogo al diferenciar un vino de otro. No todas las dimensiones olfativas y gustativas servirán para diferenciar los vinos en categorías. Sólo algunas de éstas serán las importantes y el resto serán secundarias o prescindibles. Lo que hará el enólogo es potenciar las importantes y eliminar el ruido que provocasen las demás.

En nuestro caso, habrá dimensiones que no sean importantes para diferenciar unos términos de otros y, sin embargo, habrá otros cuya aportación sí resulte relevante. En el caso de LSA, para la búsqueda de estas dimensiones importantes, se utiliza una técnica que reduce la matriz original en unas cuantas dimensiones importantes, las cuales son nuevos constructos que nada tienen que ver con las dimensiones anteriores. Las matrices son reconstruidas de manera que los términos estén representados por dimensiones que simplemente sirven para resaltar las diferencias de significado. El algoritmo que permite ese tipo de representación se conoce como *descomposición en valores singulares* (SVD) (Berry, 1992). Una vez

[1] Como veremos después, estas dimensiones pueden no identificarse con dimensiones comprensibles del mundo real. Al contrario que en el análisis factorial donde se suele dotar de significado a las dimensiones latentes, en el LSA esto no es así puesto que las dimensiones son abstracciones que contienen múltiples conglomerados de términos inconexos semánticamente. Las dimensiones pueden ser una recolocación arbitraria del espacio y no significar ni “ocurrencia de playa” ni “aparece en guía del Ayto. de Vigo”, etc.

calculado SVD se tomarán sólo las dimensiones que mejor caractericen las "bolsas" semánticas. Inicialmente, se debe buscar un equilibrio en el número de dimensiones final, de manera que éstas no sean escasas o demasiado extensas. Un número demasiado escaso de dimensiones hace que las representaciones semánticas queden muy groseras o vastas. Por el contrario, un número excesivo de dimensiones hace de éstas que sean poco latentes y queden expuestas a un uso subjetivo de cada autor.

Un criterio para determinar el número de dimensiones de la matriz es aproximarse empíricamente a la reducción de dimensiones que llevamos a cabo las personas para desentrañar el mundo que les rodea. Wierzbicka (1996) propone que el significado de las palabras puede ser definido mediante un conjunto acotado de primitivos semánticos que especificarían todos los conceptos. Estos primitivos podrían ser descritos con un metalenguaje semántico. Llamemos ahora "dimensiones más importantes" a lo que la autora denomina primitivos. Si fuésemos capaces de aislar esas dimensiones y desechar las dimensiones irrelevantes, podríamos comparar cualquier unidad lingüística gracias a ellas, al igual que haríamos si conociéramos esos primitivos. LSA es una buena aproximación para encontrar una dimensionalidad eficiente que puedan funcionar como las dimensiones más importantes, como episodios que portan la esencia de la semántica, como abstracciones *latentes* del lenguaje, o como lo define Wierzbicka, como primitivos semánticos.

El corazón del LSA es la representación de los términos y los documentos en un número de dimensiones eficiente para diferenciar semánticamente unos de otros. Al final del proceso, toda esta información se representa de forma vectorial, lo que permite que términos, documentos y nuevos documentos puedan ser comparados entre sí por medio del producto escalar, el coseno o alguna otra medida de semejanza o de similitud. En este sentido, el LSA supone una ampliación del modelo espacio-vectorial de Salton y McGill (1983), aunque con la salvedad de que el modelo derivado del LSA es un modelo refinado, ya que permite comparar textos sólo con dimensiones que marcan diferencias entre las relaciones de los términos y los documentos. El

LSA desdeña las dimensiones que no las remarcan. El problema del anterior modelo, el modelo espacio-vectorial, es que no considera la eliminación del ruido que proviene de la gran variabilidad de términos para designar los mismos referentes. Siguiendo el paralelismo de cómo los humanos hacemos uso del lenguaje, algunos estudios indican que sólo el 20% de las personas usamos las mismas palabras para designar los mismos referentes (Furnas et al., 1982). Esta variabilidad de uso de los términos (la elección de poner unos y no otros) dificulta mucho una estructura semántica del documento (Todd y Berry, 1996). Este inconveniente, este ruido intrínseco a la diversidad de términos y de usos, lo solventa el LSA sometiendo la matriz de frecuencias de términos por documentos a la descomposición en valores singulares (SVD), reduciendo las dimensiones de la matriz a las dimensiones más importantes para captar la semántica del lenguaje.

2.2.- Punto de partida

2.2.1.- El documento como unidad contextual

Llegar al espacio semántico reducido no es una tarea sencilla, ni directa ni inmediata. Como se verá, el LSA obliga a decidir entre un número amplio de pasos antes de alcanzar el producto final. La proliferación de caminos hace difícil que el espacio semántico reducido resultante sea el mismo cuando dos investigadores, de forma independiente, generan cada uno su propio LSA. Pero, ¿cómo comienza LSA su análisis? Como paso previo a cualquier operación, hay que dividir el corpus lingüístico que pretendemos analizar en documentos. Estos documentos serán lo que nosotros consideraremos unidades mínimas de significado contextual, es decir, constituyen nuestra medida del contexto. De las palabras que estén representadas dentro de los documentos se podrá decir que comparten un contexto. Este es el punto de partida de la técnica: palabras que comparten contexto en repetidas ocasiones se les infiere el mismo campo semántico. Como este es el núcleo del LSA, su idea principal, se entiende bien la

importancia de elegir los contextos. Obsérvese que uno podría escoger un capítulo, otro autor una frase, otro investigador un párrafo, todos con la convicción de que eligen el contexto de la forma más apropiada.

2.2.2.- Algunos estudios que examinan el tamaño del documento

En general, la mayoría de los autores consideran el *párrafo* como ventana contextual para representar de una manera significativa las relaciones semánticas que contraen los términos. Landauer y Dumais (1997) emplearon una unidad compuesta por los dos mil primeros caracteres de cada artículo de una enciclopedia para modelar la adquisición y representación del conocimiento. Esta ventana puede resultar arbitraria pero se justifica por el gran volumen del corpus de referencia (30.473 artículos insertos en dicha enciclopedia que contenía conocimiento de carácter general). En estudios posteriores, Landauer (2003) propone que el párrafo representa una buena solución para modelar la adquisición del conocimiento en detrimento de otra propuesta alternativa que sería la frase. Considera que en la frase están representadas de manera muy explícita la influencia de los aspectos sintácticos, propiedades éstas que no tiene oportunidad de analizarse con LSA estándar y sin duda quedará reflejado en el espacio semántico final (Wiemer-Hastings, 2000). Landauer (2003) utiliza como ventana contextual el párrafo como criterio en una aplicación llamada IEA (*Intelligent Essay Assesor*), la cual está destinada a evaluar la calidad de textos escritos por estudiantes en diversas materias. Los párrafos, recomienda, deben tener entre 50 a 300 palabras. Por su parte, Rehder et al., (1998) sugieren que para evaluar el contenido de un párrafo, el LSA ofrece peores rendimientos si dichos párrafos son inferiores a 200 palabras. Estos autores comprobaron que con 200 ó más palabras como criterio de contexto, el LSA explicaba aproximadamente el 60% de la varianza de las evaluaciones de una serie de jueces en un análisis de regresión lineal. Al reducir el contexto a 60 palabras, el LSA sólo pudo dar cuenta del 10% de la varianza, de modo que el LSA se convierte en un

buen predictor de las evaluaciones de los jueces solamente cuando tenemos un mínimo de palabras configurando el contexto*.*

La medida que propuso Landauer, tomar un párrafo de 50 a 300 palabras*,* ha sido asumida como regla general a la hora de diseñar los documentos en las matrices destinadas al análisis de la coherencia de los textos en aplicaciones educativas como *autotutor* o *e-learning* (Trusso et al, 2005) y ha demostrado su superioridad frente a la frase (Wiemer-Hastings, Wiemer-Hastings y Graesser, 1999; Kurby, Wiemer-Hastings, Gandury, Magliano, Millis y McNamara, 2003). No obstante, hay autores que recomiendan en otras ocasiones un nivel de análisis más básico que el párrafo. Por ejemplo, puede haber ocasiones en las que interese tomar una unidad como la frase para indagar sobre el efecto que ejercen las divisiones sintácticas en las propias frases (Wiemer-Hastings, 2000).

2.2.3.- Otros estudios alternativos para delimitar los contextos

Pueden existir otras circunstancias en las que se opte por unidades que no coincidan exactamente con una de las ventanas contextuales antes mencionadas (frases o párrafos), simplemente por requerimientos más prácticos. Conviene mencionar dos ejemplos que se alejan de los estándares, pues el espacio semántico latente que surge del proceso final depende enormemente de qué tipo de contexto se elija para configurarlo. No en vano, cuanto más verosímiles sean en términos psicológicos las distintas ventanas contextuales utilizadas, mayor y mejor será la obtención de un espacio que simule la semántica humana.

El primer ejemplo es el de Schütze (1998) y Cederberg et al, (2003), quienes diseñaron un corpus de contenido general representado por un espacio semántico-vectorial que proviene de una cantidad reducida de documentos. Debido a este número reducido de documentos, éstos tienen que ser lo más representativos posibles del lenguaje de uso general por lo que se forman los documentos de la siguiente manera:

1) Se seleccionan las 1000 palabras más frecuentes según un corpus normativo (no entran dentro de estas palabras las conocidas palabras *stop word,* palabras de función que se ignoran de los análisis por tener sentidos puramente sintácticos como, por ejemplo, las preposiciones).

2) Estas palabras serán la marca de cada documento o contexto.

3) El resto de palabras que componen el corpus serán asignadas a las filas, contándose el número de ocurrencias que tienen estas palabras dentro de la ventana que representa las 15 palabras próximas a cada una de las palabras que marcan un contexto (las 1000 más frecuentes). Dicho de otra forma, el documento que representa cada palabra "marcadora" lo formarán todas las palabras que aparecen en esta ventana de 15 palabras que orbitan sobre la palabra principal.

De esta manera, estos autores se aseguran de que los contextos que representan sus documentos sean contextos de uso frecuente pues tienen como pivote las palabras más frecuentes de la lengua.

El segundo ejemplo de cómo crear documentos para el corpus es la sugerida por Burek, Vargas-Vera y Moreale (2004). Estos autores aprovecharon las ontologías que fueron creadas previamente para analizarlas como si se tratasen de un documento. Es decir, se forma cada documento con cada una de las clases de la ontología. De esta manera no se necesita un corpus de referencia. Esta forma de diseñar matrices puede ser de suma utilidad cuando contemos con restricciones de cálculo en las propias máquinas o constricciones económico-temporales.

2.3.- Tratamiento y eliminación de estructuras del corpus

Conviene referirse, aunque sea brevemente, al proceso de *purga* o *poda* al que se someten los corpus antes de que los analice LSA. Hay estructuras lingüísticas que no aportan nada a la semántica y suelen depurarse antes de proceder con los siguientes pasos. No en vano, Franceschetti, Karnavat, Marineau, McCallie, Olde, Terry y Graesser (2001) han estudiado cómo, en ocasiones, los corpus pequeños pero diseñados siguiendo cuidadosamente ciertos protocolos pueden representar mucho mejor el conocimiento que corpus grandes sin ningún control.

2.3.1.- Eliminación de estructuras atendiendo a la frecuencia

Una primera forma de reducir ruido lingüístico al corpus es eliminando directamente estructuras que no van a aportar ningún beneficio a nuestro análisis semántico. Yu et al. (2004) proponen una seria de directrices a seguir y de esta forma capturar solamente las relaciones semánticas. Según ellos, la mejor forma sería la siguiente:

1. Hacer una lista completa de las palabras que aparecen.
2. Descartar los artículos, preposiciones y conjunciones.
3. Descartar verbos comunes (*saber*, *ver*, *hacer*, *ser*).
4. Descartar los pronombres.
5. Descartar adjetivos comunes (*grande*, *tarde*, *alto*)
6. Descartar adverbios (*de cualquier modo*, *posiblemente*)
7. Descartar las palabras que salgan en todos los documentos.
8. Descartar las palabras que ocurren en un solo documento (a veces llamados ermitaños).

Esta forma de proceder (salvo eliminaciones debidas a la frecuencia de aparición en los documentos, puntos 7 y 8) ha recibido el nombre de *stop word* o *palabras stop* (de hecho se representan como una lista de las palabras que serán debidamente eliminadas de la siguientes fases de análisis). Hay autores que, además, emplean una lista llamada *lista pase* o *go list* que está compuesta por palabras que aunque aparezcan con una mínima frecuencia, se garantice su pervivencia a lo largo de las purgas, pues se sabe de antemano que son grandes portadoras de significado en determinados corpus (Dam y Kaufmann, en prensa). La *lista pase* puede ser útil también para garantizar que los bigramas y trigramas no serán fragmentados (Chu-Carroll y Carpenter, 1999). Por ejemplo, el término “cuenta bancaria” puede ser transformado en “bancaria” al tomar “cuenta” como verbo frecuente. Empleando una *lista pase* se puede garantizar que este tipo de construcciones sobrevivan a lo largo del proceso y su información sea aprovechable. Un caso extremo de *lista pase* es

el que todo el espacio semántico se ha formado con términos provenientes de dicha lista, en ausencia de cualquier otro tipo de técnica de poda o purga.

2.3.2.- Lematización de estructuras

Otra de las cosas que pueden favorecer mucho la eficiencia del análisis son los procesos que transforman en lemas todos los términos que componen el corpus (Denhière, Lemaire, Bellissens y Jhean-Larose, 2007). Haciendo esto, se reduce considerablemente la cantidad de términos que entrarían a formar parte del análisis y existiría una menor variabilidad entre ellos, lo que en algunas ocasiones resultaría beneficioso. No obstante, se han vertido algunas críticas sobre estos procesos y algunos autores han comprobado que en corpus suficientemente grandes no se encuentra diferencia significativa en cuanto a la efectividad (Nakov, Valchanova y Angelova, 2003). Según señalan estos autores, lematizar el corpus no aumenta la efectividad de los modelos LSA. Incluso en corpus específicos de dominio específico y de pequeño tamaño, se ha constatado que después de aplicar en el preproceso la lista de las palabras prohibidas (stop-wors), lematizar produce peores resultados (Wild, Stahl, Stermsek y Neuman, 2006).

En cualquier caso, además de eliminar componentes y de lematizar algunos términos, se puede optar por agrupar estructuras sintácticas y semánticas de manera que se simplifiquen los textos. Un ejemplo de ello es la trasformación de estructuras subordinadas en coordinadas o la resolución y unificación de la anáfora pronominal a lo largo de las frases *(véase* Wiemer-Hastings, 2000; Wiemer-Hastings y Zipitria, 2001). No obstante, para lenguas flexivas[2] como el francés o el español, se ha encontrado beneficio en la lematización de los verbos, aunque no de los nombres (Denhière y Lemaire, 2004). Denhière et al. (2007) recomiendan lematizar los verbos en una única forma, pero no lo promueven en adjetivos y nombres, ya que singulares y

[2] Las **lenguas flexivas** o **sintéticas,** en contraposición a lenguas **analíticas**, son aquellas lenguas que se caracterizan por una tendencia a incluir mucha información en sufijos o prefijos mediante la flexión de algunas palabras (en este caso, los verbos). La flexión se emplea a menudo para diferenciar los casos que acepta la lengua.

plurales, junto con femenino masculino en muchas ocasiones portan distinto significado. En el siguiente apartado expondremos por qué los verbos representan partículas peligrosas al análisis.

2.3.3.- Eliminación de verbos

Una variedad de LSA puede contemplar la posibilidad de eliminar todos los verbos del corpus. Esto puede hacerse necesario cuando el corpus a analizar posea unas dimensiones muy reducidas los verbos no estén lo suficientemente muestreados. En corpus de este tipo es habitual toparse con ejemplos como este con el que nos encontramos en un estudio: El término *Skinner* se relaciona semánticamente con el término *propuso* (reflejado con un coseno amplio). Este efecto se produce porque en algún párrafo (o más de un párrafo) del corpus ambos términos concurren, lo cual propicia que ambas palabras estén cercanas en el espacio semántico. Se concederá que, al menos *a posteriori*, la relación entre *Skinner* y *conductismo* no está dotada de la misma entidad que *Skinner* y *propuso*. La variabilidad en las formas verbales (tiempos, formas, voz, etc.) y la variabilidad de formas estilísticas (*propuso*, *defendió*, *ofreció*, *generó*, etc.), hacen de los verbos partículas "peligrosas" en corpus de reducido tamaño. Dos simples ocurrencias verbales pueden elevar una relación espuria a una fuerte relación semántica que pretende hacer que la forma verbal sea distintiva del término que acompaña (en este caso *Skinner*). *Propuso* ocurre junto a *Skinner* de una manera casi estética, pero sin aportar rasgos distintivos. Se excusa decir que nada hay en *propuso* que nos lleve a relacionarlo distintivamente con *Skinner* ya que el verbo *proponer* es por lo general una acción que puede ser atribuida a todos los científicos e incluso a toda la especie humana. Por regla general, en una muestra amplia del lenguaje, estos efectos se disipan ya que términos como *proponer* salen insertos en documentos que tratan de muy diversos temas, pero es preciso prevenir estos efectos indeseables para algunos corpus. Para evitar que las combinaciones aleatorias de los verbos en los corpus, sean valoradas como verdaderas relaciones semánticas, podemos eliminar, aunque con ciertos costes, todas las ocurrencias de los verbos en el corpus. Retirar los verbos del corpus entraña ciertos problemas que pueden sesgar de una manera excesiva

el análisis semántico, por lo que su eliminación ha de ser acotada a un cierto tipo de corpus y de casuística. La cuestión metodológica de la retirada de los verbos entraña dificultades que son insalvables para cierto tipo de circunstancias (a no ser que se cuente con *parseadores* muy sofisticados que empleen criterios sintácticos y gramaticales). A saber:

a) Se prescinde de la información que los verbos pueden aportar a las relaciones del espacio semántico. Esta consideración encuentra cierto alivio en los trabajos de algunos autores que encuentran que no está clara la importancia de las formas verbales como términos de alta significatividad o peso dentro de los documentos. Por ejemplo, en los estudios efectuados por Gil Leiva y Rodríguez Muñoz (1997), aunque circunscritos a las expresiones clave utilizadas (manualmente) para indizar documentos en español, muestran una escasa presencia de las formas verbales en dichas expresiones clave. Aunque, desde luego, esto no sea directamente extrapolable a la totalidad del documento, sí que puede hacer pensar que la importancia de los verbos pudiera ser menor que la de los sustantivos. En lo que a humanos se refiere, los experimentos que realizaron Kersten y Earles (2004) concluyen que el recuerdo de los verbos es más dependiente del contexto en los que aparecen que la memoria para los nombres. Según estos autores, el significado verbal entraña el conocimiento parcial de los objetos que se ven involucrados lo cual les genera mayor dependencia contextual para su retención. Esto puede ser debido a la estrategia de aprendizaje en edades tempranas de nombres y verbos. Los primeros son aprendidos en base a su apariencia e ignorando las acciones y movimientos. Los segundos son aprendidos a partir de las acciones y movimientos de los nombres (Kersten y Smith, 2002). Atendiendo a la posibilidad de eliminar los verbos del corpus, estas evidencias muestran que los verbos podrían portar menos contenido semántico que diferencie a unos documentos de otros o que este contenido no es tan central ni característico como el de los sustantivos. Así pues, si la retirada de los verbos puede ayudar a reducir la captación de relaciones espurias en cierto tipo de corpus, podemos contar aún con las estructuras que más definen las relaciones semánticas: los sustantivos.

b) El segundo problema que surge al eliminar formas verbales se produce cuando eliminamos términos que no tienen una función gramatical de verbo. Por ejemplo, esto se da con los homónimos (cuyo significado pertenece a otro grupo semántico), como puede ser el término *pienso:* dicho término representa la primera persona del singular del presente de indicativo del verbo pensar o un sustantivo común que hace referencia a la comida del ganado. Esta forma automatizada de adelgazamiento del corpus eliminaría todas las ocurrencias ya que sólo atendería a su forma literal. A su vez, hay formas verbales que son utilizadas también como sustantivos y que en una medida considerable pertenecen a una categoría diferente a la de verbo. Un ejemplo habitual, aunque no único, es el de los participios los cuales pueden tener papel de adjetivos o de sustantivos. Otro es el de algunas personas verbales y su coincidencia grafémica con sustantivos. Mostramos a continuación algunos ejemplos son la palabra *abogado* que puede ser verbo o sustantivo, *bebido* puede ser verbo o adjetivo, *pescado, cena, ducha* o *ahorro.* No hay forma de diferenciar entre las distintas categorías gramaticales de estos términos, con lo que se pierde información que no es puramente verbal al actuar de esta forma.

En general y como recomendación, solamente parece que la eliminación de los tiempos verbales es útil ante condiciones muy concretas, ya que, como se ha visto, el uso de esta forma de depuración pues puede eliminar gran parte información semánticamente relevante (y eso sin contar con la información relevante que poseen los propios verbos). Opciones alternativas y más seguras sería hacer pruebas bajo la condición de eliminación de los verbos y sin dicha condición, estudiar el comportamiento semántico de ambas formas de proceder y valorar, en cualquier caso, si hay mejoría conforme a la utilización de todas las formas verbales o sin ellas. Por supuesto, también existen formas menos automatizas, con la que podrían eliminarse formas verbales indeseables. También es posible el empeoramiento de algún algoritmo de identificación gramatical o incluso implementar una función probabilística empleando la frecuencia de aparición de los términos representando unas formas gramaticales u otras.

En cualquier caso, conviene insistir en que la eliminación de las formas verbales tiene únicamente sentido cuando éstas estén muy poco representadas y reducidas en corpus de pocos documentos (corpus típicamente de dominio específico y no generalistas), donde la carga la lleven los sustantivos y no haya excesivo problema con los sesgos antes expuestos y donde se promuevan relaciones semánticas espurias entre los términos y dichas formas verbales.

2.3.4.- Eliminación de contenido tangencial

Una cuestión que se plantean los investigadores que trabajan con LSA y con corpus de dominio específico (aquellos que tratan sobre una temática restringida como, por ejemplo, física, literatura o psicología, es saber hasta qué punto conviene ceñirse estrictamente al dominio de conocimientos sobre el que se quiere investigar. Sobre este aspecto Olde, Franceschetti, Karnavat y Graesser (2002) realizaron una simulación utilizando cinco corpus de física, manipulándolos de tal manera que cada uno de ellos recortaba información del anterior. Así, el más breve contenía información absolutamente nuclear sobre el dominio de conocimientos mientras que el último portaba, además de la información nuclear, también información de tipo más generalista y alejada de la temática. Esto se hizo para comprobar en qué medida, suprimir información no central iba a repercutir en la efectividad y la calidad de las relaciones semánticas que encuentra LSA. Las conclusiones del estudio fueron que no hay beneficio en eliminar meticulosamente la información tangencial que normalmente se adjunta en los libros de texto. El rendimiento es prácticamente igual para todos los corpus excepto para el más pequeño (es decir, llega un punto en el que la pérdida de la información es negativa para el LSA). La inclusión de información que los expertos consideran tangencial no disminuye la calidad de las relaciones semánticas que el LSA arroja (tampoco la aumenta). En un estudio posterior sobre el mismo corpus y las mismas manipulaciones, Franceschetti, et al.(2001) afinaron aún más las conclusiones: Los textos depurados en cuanto a información tangencial permite que se represente mejor la información central del tópico y que algunos términos se erijan como representantes de los contenidos más importantes. Aún así, continúan los autores, los corpus con información tangencial funcionan mejor a

la hora introducir comparaciones por medio de pseudodocumentos que son introducidas por los usuarios. En un estudio nuestro ampliamos estos hallazgos (Jorge-Botana, León, Olmos y Escudero, 2010) y encontramos que la ausencia de información tangencial, hace que la mera aparición de algunos términos clave infle la valoración que LSA tiene de resúmenes académicos. La ausencia de información tangencial incide en que unos pocos términos clave concentren todo el poder discriminatorio para juzgar la idoneidad de un resumen y que las comparaciones con el coseno sean menos fiables.

También respecto al tamaño de los corpus, Denhière y Lemaire (2004) advierten que en un corpus de dominio general, hay un tamaño crítico por encima del que, añadir más cantidad, no cambia significativamente la representación del conocimiento. Rehder et al. (1998) encuentran similares resultados en cuanto a la inclusión de información no técnica en el análisis. Reanalizan los datos de Wolfe (1998) en los que se diseñaba un método para proporcionar a los alumnos los textos más acordes con su nivel. El corpus de referencia versaba sobre “el corazón y el sistema circulatorio”. Tomando esta experiencia anterior, Rehder et al. (1998) dividieron los términos de dicho corpus entre técnicos (palabras usadas específicamente en descripciones sobre el sistema circulatorio) y no técnicos. En esta clasificación salen 47% de términos técnicos y el resto no-técnicos. Hecho esto, pidió a los estudiantes que escribieran sobre el sistema circulatorio. Por cada ensayo producido por el estudiante, el LSA generaba dos vectores, uno que contiene exclusivamente los términos técnicos y el otro con los no técnicos. Cada uno de estos ensayos más el ensayo que contiene todos términos (el original del alumno) se compara con el texto de referencia obteniéndose los cosenos como medida de similitud. Las tres medidas de similitud, es decir, las medidas de posesión de conocimiento según el análisis se correlacionan con la aptitud de cada alumno en un precuestionario que se les administró previamente. Las correlaciones realizadas entre este precuestionario y los cosenos de los tres tipos de ensayos individuales y el texto de referencia resultaron significativas. Los resultados mostraron cómo los ensayos compuestos con los términos no técnicos correlacionaron igual de bien con los resultados del precuestionario

que los ensayos con términos técnicos. Los ensayos con los términos no técnicos funcionaron como buenos predictores de posesión de conocimiento. Las palabras no técnicas que los alumnos utilizaron para escribir sobre el tema del "corazón y el sistema circulatorio" portaron gran información sobre el conocimiento que estos poseen. Aunque estos autores advierten que los resultados pueden ser debidos a que "la única forma de producir una colección acertada de palabras sobre un tema es redactar un buen ensayo" los autores concluyeron que nada se gana retirando los términos no técnicos de los textos sobre un dominio temático.

2.4.- Extracción de términos y formación de la matriz de ocurrencias

Una vez depurado nuestro corpus y extraídas las estructuras que consideramos no relevantes para el análisis y una vez acometida la división del corpus en documentos, la matriz de ocurrencias se obtiene captando el número de apariciones de cada término, en cada uno de los documentos. Es importante que los documentos configuren una unidad de contenido pues de estas unidades se extraerán los términos que ocuparán las filas de la matriz de ocurrencias. Esa ocurrencia será consignada en cada una de las celdas correspondientes. Generalmente y de manera estándar, se colocan los términos no repetidos (únicos) ocupando las filas de la matriz y cada uno de los documentos en los que hemos dividido el corpus ocupando las columnas (identificados por un número). Decimos términos no repetidos ya que cada fila ha de representar la ocurrencia de un término en cada uno de los documentos pero dos filas no podrán representar el mismo término. Veamos el siguiente ejemplo[3]:

A1:Los archivos planos constituyen la forma más básica de una base de datos #

A2:Los archivos planos incluyen un campo por cada uno de los elementos que se desean contemplar #

A3:La redundancia de elementos es una característica de estos archivos #

A4:La base de datos relacional soluciona la redundancia#

B1:Son frutos largos y con sabor #

[3] Cada una de estas frases configurará un documento en el análisis. Además, en dicho análisis sólo se tendrán en cuenta los sustantivos los cuales aparecen en color rojo y subrayados.

B2:La recogida será buena si ha tenido una buena base como semillero #

B3:Los frutos son de color verde #

B4:En la recogida es parecida a los demás frutos largos #

Supongamos que tenemos los siguientes enunciados y que cada uno de ellos representa un documento. Los documentos –A- representan documentos de una temática reconocida por nosotros a priori. En este caso los documentos -A- representan al tópico “bases de datos”. Por otra parte, los documentos –B- representan el tópico “siembra de frutas”. La matriz de ocurrencias quedaría de la siguiente forma:

	A1	A2	A3	A4	B1	B2	B3	B4
	0	0	0	0	0	0	0	0
archivos	0	1	1	1	0	0	0	0
planos	0	1	1	0	0	0	0	0
base	0	1	0	0	1	0	1	0
datos	0	1	0	0	1	0	0	0
campo	0	0	1	0	0	0	0	0
elementos	0	0	1	1	0	0	0	0
redundancia	0	0	0	1	1	0	0	0
relacional	0	0	0	0	1	0	0	0
frutos	0	0	0	0	0	1	0	1
largos	0	0	0	0	0	1	0	1
sabor	0	0	0	0	0	1	0	0
recogida	0	0	0	0	0	0	1	0
semillero	0	0	0	0	0	0	1	0
color	0	0	0	0	0	0	0	1
verde	0	0	0	0	0	0	0	1

Tabla 2.1.- Matriz de ocurrencias de cada uno de los términos en cada uno de los documentos de los que se compone el texto.

	A1	A2	A3	A4	B1	B2	B3	B4
archivos	1	1	1	0	0	0	0	0
planos	1	1	0	0	0	0	0	0
base	1	0	0	1	0	1	0	0
datos	1	0	0	1	0	0	0	0
elementos	0	1	1	0	0	0	0	0
redundancia	0	0	1	1	0	0	0	0
frutos	0	0	0	0	1	0	1	1
largos	0	0	0	0	1	0	0	1
recogida	0	0	0	0	0	1	0	1

Tabla 2.2.- Matriz de ocurrencias de cada uno de los términos en cada uno de los documentos de los que se compone el texto. La diferencia con la anterior es que esta vez se han purgado los términos que ocurren en un solo documento.

Cada uno de los términos no repetidos se colocaría en la parte vertical de la matriz siendo representadas sus ocurrencias por cada una de los vectores filas. A su vez, los documentos quedarían posicionados en la parte horizontal siendo representados sus términos componentes en cada uno de los vectores columnas. Bastaría entonces con observar el texto y comprobar que la representación de la matriz es correcta. Por ejemplo, el término “base” está representado en los documentos A1, A4, B2. Sobre esta matriz es donde se aplican algunos ajustes lingüísticos opcionales que dan cuenta de un fenómeno bastante intuitivo y es que los términos que se repiten en la mayoría de los documentos proveen de una menor información sobre el tópico sobre el que versa este documento, es decir, discriminan peor y son menos representativos del tema. Pongamos, por ejemplo, el término "cosa" en el lenguaje coloquial. Seguramente, si implementásemos un sistema que se encargase de discriminar el tópico de los documentos en base a los términos que lo componen, no escogeríamos “cosa” como uno de los términos discriminativos, pues se suele repetir en multitud de producciones sean estas del tema que sean. También las palabras de función suelen ser un buen ejemplo de términos que se repiten en casi todos los documentos y que no indican sobre que trata el mismo. Sobre este tipo de ajuste hablaremos en el siguiente apartado.

2.5.- Ajustes lingüísticos a la matriz de coocurrencias.

Una forma muy efectiva de reducir el problema de la representatividad de los términos en los párrafos es la transformación que se realiza sobre la matriz de coocurrencias antes de someterla a la *descomposición del valor singular* (SVD). Gracias a esta transformación, cada celdilla de la matriz expresa la importancia de ese término en ese documento y en qué medida ese término nos aporta información sobre ese documento (Landauer, et al. 1998). En otras palabras, la transformación da cuenta del fenómeno según el cual las palabras cuya ocurrencia se distribuye por casi todos los documentos no portan ninguna información sobre ellos. Véase el caso extremo anteriormente expuesto del término “cosa”. Este suelen salir en casi todos los documentos y, sin embargo, no aportan ninguna información sobre aquellos documentos en

los que aparece. Se trata de un término comodín del lenguaje coloquial que es utilizado en gran cantidad de documentos pero que no aporta ninguna información de ellos.

La motivación de esta transformación es ponderar cada término en base a su capacidad para representar supuestos dominios semánticos. Se infiere que si un término ocurre en un número muy alto de documentos será mal predictor del dominio al que puede pertenecer. Imagínese el lector un término como “dolor” en un corpus basado en una taxonomía médica. Este término no nos ofrecería gran información sobre tipos de enfermedades, no así por ejemplo “inmunodeficiencia”. Esta transformación trata de menguar el influjo de términos muy frecuentes y poco informativos como “dolor”.

Una vez que se ha construido la matriz de ocurrencias y con antelación al SVD, se calculan los pesos locales y globales de cada uno de los vectores de esta matriz. Respecto a los pesos globales, es recomendable calcular uno de estos dos:

(1) Frecuencia del término

$$l_{ij} = tf_{ij}$$

(2) Logaritmo

$$l_{ij} = log(tf_{ij} + 1)$$

Donde tf_{ij} el número de ocurrencias de un término *i* en un documento *j*

Respecto a los pesos globales, la llamada fórmula de la Entropía o IDF (*Inverse Document Frequency*) han sido las más utilizadas por todos los autores.

(3) IDF

$$g_i = log_2 (n / df_i) + 1$$

Dónde $\boldsymbol{df_i}$ es el número de documentos en el que el términos ***i*** ocurre.

(4) Entropía

$g_i = 1 + \Sigma_j (p_{ij} \log(p_{ij}) / \log(n))$

Dónde $p_{ij} = tf_{ij} / gf_i$

Dónde

$\boldsymbol{tf_{ij}}$ es el número de ocurrencias de el término ***i*** en el documento ***j***

$\boldsymbol{gf_i}$ es el número de veces en que ***i*** ocurre a lo largo de todos los documentos

n es el número de documentos

El resultado final es el producto entre el peso local y el peso global ($\boldsymbol{x_{ij} = l_{ij} * g_i}$) que será el nuevo valor para cada celda.

	A1	A2	A3	A4	B1	B2	B3	B4
archivos	0,326943084	0,326943084	0,326943084	0	0	0	0	0
planos	0,46209812	0,46209812	0	0	0	0	0	0
base	0,326943084	0	0	0,326943084	0	0,326943084	0	0
datos	0,46209812	0	0	0,46209812	0	0	0	0
elementos	0	0,46209812	0,46209812	0	0	0	0	0
redundancia	0	0	0,46209812	0,46209812	0	0	0	0
frutos	0	0	0	0	0,326943084	0	0,326943084	0,32694308
largos	0	0	0	0	0,46209812	0	0	0,46209812
recogida	0	0	0	0	0	0,46209812	0	0,46209812

Tabla 2.3.-. Matriz de ocurrencias una vez calculada la función de Log-Entropía.

	A1	A2	A3	A4	B1	B2	B3	B4
archivos	-0,630929754	-0,630929754	-0,630929754	0	0	0	0	0
planos	-1	-1	0	0	0	0	0	0
base	-0,630929754	0	0	-0,630929754	0	-0,630929754	0	0
datos	-1	0	0	-1	0	0	0	0
elementos	0	-1	-1	0	0	0	0	0
redundancia	0	0	-1	-1	0	0	0	0
frutos	0	0	0	0	-0,630929754	0	-0,630929754	-0,630929754
largos	0	0	0	0	-1	0	0	-1
recogida	0	0	0	0	0	-1	0	-1

Tabla 2.4.- Matriz de ocurrencias una vez calculada la función de Log-IDF.

En estas gráficas presentamos la matriz formada a partir de la fórmula (C) o entropía y (B) o IDF como peso global y el logaritmo de la frecuencia como peso local. Nótese como se ponderan las apariciones de los términos en los documentos. Los términos que aparecen en muchos documentos serían ponderados de manera menor que aquellos términos con menor frecuencia. A su vez y aunque aquí no se represente el caso, las ocurrencias del mismo término en un mismo documento no son representadas linealmente (ej. dos

ocurrencias no representarían el doble) sino que se ponderan también logarítmicamente. Esto evita que un documento colapse la información que aporte un término concreto. En ejemplos tan sencillos como este no se puede apreciar en todo su extensión el beneficio que este tipo de cálculos ofrece en el análisis del texto. Imagínese el lector el posible efecto demoledor que produjese una frecuencia inusitada de un término en muchos documentos o la aparición de un término en un solo documento. Estos dos tipos de situación extrema harían que las "unidades de significado" que representan los textos estuvieran "abanderados" por términos que, en realidad, representan muchas temáticas en el primer caso o "pasaba por ahí en ese momento" en el segundo caso. La verdadera esencia de la técnica es conocer en todo momento la naturaleza del texto a introducir y evitar las relaciones espúreas que se pudiesen formar.

2.6.- SVD como procedimiento estándar

2.6.1.- Aplicación de SVD

Una vez obtenida la matriz de ocurrencias después de haberla sometido a los ajustes lingüísticos es cuando se aplica el procedimiento de la *descomposición del valor singular* (SVD son sus siglas en inglés, S*ingular Value Decomposition*). Se trata de una técnica estándar que se aplica en álgebra lineal sobre matrices. Es una forma específica de análisis factorial. En la matriz original, términos y documentos son mutuamente dependientes entre ellos. La técnica SVD devolverá un desglose de las relaciones que se mantienen en la matriz original. En ella, la matriz original **(x)** es descompuesta en el producto de tres matrices. Las matrices resultantes contendrán "vectores singulares" y "valores singulares", estos últimos explicando la variabilidad explicada por cada dimensión a lo largo de términos y documentos.

Una matriz contendrá la representación de los términos **(T),** cuyos componentes o factores serán linealmente independientes de la relación con los documentos en la matriz original. Otra matriz contiene la representación de los documentos **(D)** de la misma forma que la de términos, es decir, como

vectores singulares cuyos componentes son linealmente independientes de la relación con los términos en la matriz original. Por último, una matriz diagonal **(S)** de valores singulares escalados (de mayor a menor aportación para agrupar) y cuya aportación es que la matriz independiente de términos multiplicada por ella y, a su vez, multiplicada por la matriz traspuesta de la matriz independiente de documentos, reconstruyen la matriz general.

$$X = T\,S\,D'$$

Lo relevante es que la matriz original puede ser reconstruida de una manera fidedigna sin emplear todas las dimensiones o factores que agrupaban en torno a si las relaciones de la matriz original Términos-Documentos. En realidad, basta con emplear únicamente las dimensiones que más peso tienen a la hora de formar conglomerados y cuya ponderación está descrita en la matriz diagonal de valores singulares (los factores que más varianza expliquen). Como los valores de la matriz de valores singulares están ordenados de mayor a menor aportación, entonces será fácil tomar sólo los que más ayuden a construir agrupaciones. Se reducen las dimensiones y se vuelven a multiplicar.

$$X_j = T_k\,S_k\,D_k'$$

De esta nueva multiplicación de las tres matrices resultará otra tal que sea parecida a la original, pero con la particularidad de que se ha reducido gran parte del ruido que ejercían dimensiones o factores que no eran del todo relevantes en las relaciones entre términos y documentos. El número de dimensiones (K) que se tomarán en esta nueva multiplicación sigue siendo un tema en estudio. Generalmente, en corpus generalistas (representan un conocimiento general) se suelen tomar entre 200, 300 o 400 dimensiones, pero todo dependerá del tipo de texto que estamos sometiendo al análisis.

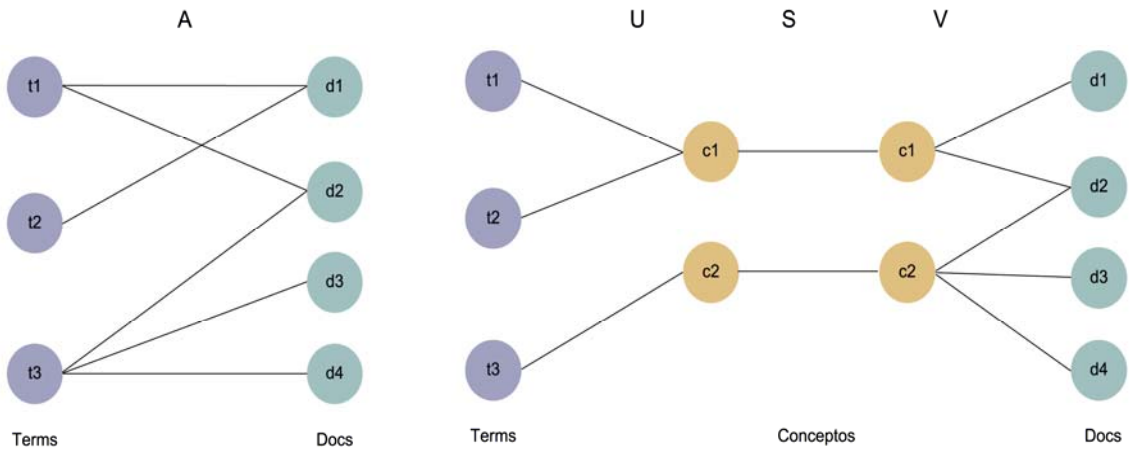


Figura 2.2.- *Representación esquemática de lo que significa la reducción de dimensiones llevada a cabo por medio de SVD. En la figura de la izquierda, cada término está representado por cuatro dimensiones, tantas como párrafos existen en el corpus.{d1,d2,d3,d4}. En la figura de la derecha, los términos pasan a estar representados por dos dimensiones abstractas pero de una mayor utilidad funcional. A cada término se le infiere una probabilidad de estar representado en un concepto. Compruébese por ejemplo que al término t2 se le infiere cierta probabilidad de salir en el párrafo d2 aunque como muestra la figura de la izquierda, esto no se produzca.*

En definitiva, lo que hace es mediante la SVD, buscar las dimensiones que mejor permitan una diferenciación de las “bolsas” semánticas en las que los términos participan. Una vez hecho, elegiremos las dimensiones que mejor permitan esto, sin restringir el número de estas dimensiones tanto, de manera que la representación semántica quede muy grosera ni elegir demasiadas haciendo que las posibles diferencias se difuminen excesivamente (véase la figura 2.2.). Se ha de preservar la mayor información posible de la matriz original pero reduciendo sus dimensiones. En una situación ideal, esta información que se pierde fruto de la reducción de dimensionalidad representa el ruido que no permite ver las diferencias entre los grupos semánticos. Las cosas que empezaban a ser similares mínimamente en la matriz original, aparecen ahora mucho más similares mientras que lo que no era similar permanece distinto. Veamos nuestro ejemplo y cotejemos las matrices en las que se descompone la matriz de ocurrencias. La matriz señalada con {x} se refiere a la matriz de ocurrencias pero después de realizados los cálculos de Entropía.

{X}=

	A1	A2	A3	A4	B1	B2	B3	B4
archivos	0,326943084	0,326943084	0,326943084	0	0	0	0	0
planos	0,46209812	0,46209812	0	0	0	0	0	0
base	0,326943084	0	0	0,326943084	0	0,326943084	0	0
datos	0,46209812	0	0	0,46209812	0	0	0	0
elementos	0	0,46209812	0,46209812	0	0	0	0	0
redundancia	0	0	0,46209812	0,46209812	0	0	0	0
frutos	0	0	0	0	0,326943084	0	0,326943084	0,326943084
largos	0	0	0	0	0,46209812	0	0	0,46209812
recogida	0	0	0	0	0	0,46209812	0	0,46209812

{T}=

-0,4502742	-0,076894667	-0,282356921	-0,109020956	-0,01616935	-0,0082696	-0,278372165	0,788611723
-0,450708	-0,061073536	-0,155135626	-0,593314212	0,077487543	-0,02628386	-0,002484663	-0,400427489
-0,3439955	0,115583795	0,472455017	-0,065549971	-0,22621022	0,158334773	0,716880361	0,225030036
-0,4307413	0,007140497	0,521449303	-0,053635244	0,323185938	-0,09624894	-0,372905208	-0,191972625
-0,3860813	-0,102036233	-0,578626469	0,220468585	-0,18712519	0,04020842	0,264329243	-0,31719447
-0,3661147	-0,033822199	0,09795846	0,760147553	0,058573211	-0,02975666	-0,106091302	-0,108739606
-0,0221698	0,515147791	-0,147801087	0,025013382	0,375367119	0,754411052	-0,039022751	-0,010056651
-0,0286143	0,636903403	-0,17141047	0,026750109	0,340182798	-0,62701248	0,223654484	0,06743772
-0,0763898	0,542557008	0,073845968	-0,033870374	-0,73691421	-0,01114229	-0,369942701	-0,114768262

{S}=

1,10961179	0	0	0	0	0	0	0
0	0,923779623	0	0	0	0	0	0
0	0	0,771762291	0	0	0	0	0
0	0	0	0,662750918	0	0	0	0
0	0	0	0	0,545816223	0	0	0
0	0	0	0	0	0,259442774	0	0
0	0	0	0	0	0	0,145414876	0
0	0	0	0	0	0	0	0,136410336

{D}=

-0,6011085	-0,481152791	-0,445923878	-0,433207924	-0,01844868	-0,13316959	-0,006532261	-0,050261235
-0,0132859	-0,108806143	-0,095174329	0,027560424	0,50091587	0,31230814	0,182320549	0,772316719
0,29986451	-0,558960619	-0,407418748	0,571021762	-0,16524648	0,244362786	-0,062613248	-0,121030682
-0,5371986	-0,313745187	0,629945905	0,460273826	0,030990719	-0,05595246	0,012339406	0,007374854
0,1940326	-0,102506672	-0,118519896	0,187704788	0,512849386	-0,75938479	0,224844331	-0,111035828
-0,0291371	0,014380142	0,008194683	-0,024901416	-0,16609371	0,179683656	0,950689326	-0,185939443
-0,2069921	0,206210224	-0,123030045	0,089644812	0,622990578	0,436195053	-0,08773668	-0,552610095
0,42266727	-0,540872569	0,447237467	-0,479337157	0,204345887	0,150559086	-0,024103398	-0,184438422

Nótese como la matriz diagonal devuelve los valores singulares escalados de mayor a menor. De estos valores, se escogerán los mayores y se reducirán las tres matrices a las mismas dimensiones de la nueva matriz diagonal. Después de reducir el número de factores se vuelven a multiplicar las matrices. De esta nueva multiplicación resultará otra $\mathbf{X_i}$ tal que sea parecida a la original **X** pero con la particularidad de que se ha reducido gran parte del ruido que ejercían factores que no eran del todo relevantes en las relaciones entre términos y documentos. De esta manera se reconstruirá una matriz $\mathbf{X_i}$ pero sin el ruido que suponen los factores que tienen poco peso. En nuestro ejemplo, la reducción de la matriz a dos factores se expresa de la siguiente forma:

$$\mathbf{X_i = Tk\ Sk\ Dk'}$$

La matriz reconstruida quedaría entonces así:

$$\{X_i\}=$$

	A1	A2	A3	A4	B1	B2	B3	B4
archivos	0,301275326	0,248127069	0,229557343	0,21448577	-0,02636442	0,044351052	-0,009687197	-0,029748535
planos	0,301370485	0,246768437	0,228380987	0,215097091	-0,01903453	0,04897961	-0,007019395	-0,01843675
base	0,228025417	0,172039069	0,160047663	0,16829884	0,060526656	0,084177403	0,02196046	0,101648098
datos	0,287215581	0,229251995	0,212504054	0,207235981	0,012121814	0,065709221	0,004324762	0,029117033
elementos	0,258767425	0,216381999	0,200004998	0,182988624	-0,0393124	0,027612052	-0,014386928	-0,051265863
redundancia	0,244612521	0,198865557	0,184128065	0,175127514	-0,00815606	0,044341663	-0,003042771	-0,00371208
frutos	0,008464695	-0,039942683	-0,034322162	0,023772413	0,238831199	0,151898104	0,086923948	0,368768843
largos	0,011268777	-0,048740058	-0,04183821	0,029970071	0,295303811	0,187977344	0,107477228	0,455994848
recogida	0,044292864	-0,013750005	-0,009903807	0,05053339	0,252624357	0,167817669	0,09193332	0,391347835

Tabla 2.5. Matriz reconstruida tomando sólo las dimensiones importantes, en este caso las dos primeras dimensiones.

$$\{T_k\}$$

-0,601108498	-0,481152791	-0,44592388	-0,43320792	-0,018448677	-0,133169586	-0,006532261	-0,050261235
-0,013285867	-0,108806143	-0,09517433	0,027560424	0,50091587	0,31230814	0,182320549	0,772316719

$\{S_k\}$

1,10961179	0
0	0,923779623

$\{D_k\}$

-0,4502742	-0,076894667
-0,450708	-0,061073536
-0,3439955	0,115583795
-0,4307413	0,007140497
-0,3860813	-0,102036233
-0,3661147	-0,033822199
-0,0221698	0,515147791
-0,0286143	0,636903403
-0,0763898	0,542557008

De esta manera, volvemos a tener nuestra primera matriz de ocurrencias pero con una diferencia: Después de haber aplicado SVD y la reducción de factores, cada término tiene un probabilidad inferida de estar en un documento al igual que cada documento tiene un probabilidad inferida de contener esos términos (Tabla 2.5). Por eso puede decirse que LSA es más que una técnica de co-ocurrencias, capta también las probabilidades de ocurrir juntos aunque no ocurran.

2.6.2.- ¿En qué proporción participa la SVD en el proceso de análisis de la semántica latente?

La *descomposición del valor singular* (SVD) es una técnica estándar que captura las regularidades de los patrones de ocurrencia entre los términos y documentos. Algunos autores se han preguntado la proporción de la participación de la descomposición del valor singular (SVD) en el proceso completo del análisis de la semántica latente (LSA). Es pertinente recordar que además de la técnica SVD, el análisis cuenta con cálculos de entropía en el preproceso y medidas de similitud en el postproceso. No está claro cuantitativamente cual es su verdadera contribución. En un informe técnico, Wiemer-Hastings (1999) trató de poner luz sobre esta cuestión comparando dos tipos de LSA además de la simple búsqueda literal (*keyword*). Para consignar la ganancia de poder discriminativo que proviene de la descomposición del valor singular (SVD), compara un modelo completo de LSA con otro en el que no se introduce la descomposición del valor singular (SVD). Este último modelo de LSA si posee el cálculo de la entropía y las medidas de similitud. Compara, además, estas dos aproximaciones con la simple búsqueda literal de términos, es decir, en qué medida coinciden literalmente los textos de los usuarios con las respuestas ideales, medido esto según una fórmula descrita en este artículo, si coinciden todos los términos, la coincidencia sería representada con 1, una coincidencia nula sería representada con 0. En la versión literal se introduce también el cálculo de pesos de LSA.

Las comparaciones de los tres modelos entre si se realizan correlacionando cada uno de ellos con diferentes criterios (humanos) que se emiten sobre el conocimiento que representan los textos introducidos por los usuarios del *AutoTutor* (Wiemer-Hastings et al., 1998). AutoTutor es un sistema de tutoría inteligente que valora el conocimiento sobre “sistemas operativos” que representan las respuestas introducidas por los alumnos. A todos estos grupos-criterio (desde expertos hasta estudiantes sin experiencia) se les pide que emitan un juicio del 1 al 6 sobre el nivel de conocimientos que posee un usuario, una vez visto el texto que ha producido (acotado en párrafos). La profundidad con que cada modelo emula el criterio humano estará en función

de su correlación con los grupos-criterios. Los resultados son que la versión completa de LSA consigue un rendimiento máximo de r=0.48, mientras que la versión sin-SVD lo hace de r=0.43. Sin embargo, la versión completa tiene un comportamiento más estable a lo largo de la curva que relaciona el umbral que se elige y la correlación con el criterio humano. Esto significa que los resultados de la versión sin SVD (sólo cálculos de pesos de entropía y comparaciones geométricas) obtienen unos resultados que, aunque más bajos, se aproximan a la versión completa. La ganancia que representa esta diferencia es de un 20%. La versión con búsquedas literales obtiene un rendimiento máximo de umbral r=0.40, correlación parecida a r=0.43 de la versión LSA sin SVD. La conclusión final es que SVD hace el análisis LSA más robusto y capaz de explotar el significado de los textos con fenómenos como la sinonimia y la homonimia. pero también parece observarse que la sola combinación de los cálculos de entropía y las comparaciones geométricas de los vectores son suficientes para producir juicios que se aproximan, aunque no llegan, al rendimiento de la versión completa. Además, dada la diferencia de capacidad y tiempo de cálculo que requieren cada una de las versiones, es posible utilizar versiones reducidas cuando importan más las limitaciones en los recursos de cálculo que la capacidad de juicio (como, por ejemplo, buscadores sencillos). En cualquier caso, como los mismos autores advierten, es posible que las diferencias se hagan mayores si ponemos a prueba las tres versiones en textos de mayor tamaño (Landauer y Dumais, 1997). Recordemos que la técnica LSA es más efectiva en la medida en la que se procesan textos mayores y más representativos. Puede resultar interesante probar este mismo protocolo de evaluación de la aportación de la descomposición del valor singular en un texto mayor y de dominio general de uso. Es importante recalcar aquí que estos autores utilizaron un corpus relativamente pequeño y con un dominio específico de conocimiento como es la informática. Como ha sido experimentado con las herramientas de esta misma tesis, la técnica LSA se comporta como una comparación entre literales si los textos de referencia son de tamaño reducido y sin ningún control previo.

2.6.3.- El número de factores o porcentaje acumulado de valor singular

2.6.3.1.- Significado de los factores

Como ya se dijo en el apartado anterior, una vez realizada la descomposición del valor singular, hay que reconstruir la matriz de nuevo pero esta vez utilizando únicamente los valores singulares que más aporten a la relación de ambas matrices de vectores. Estas dimensiones relevantes serán localizadas a partir de la matriz diagonal de valores singulares. Aún así, las dimensiones del nuevo espacio semántico no son identificables, todo lo más sirven para comparar unos vectores con otros y constatar la cercanía semántica de unos con otros. Kintsch (2001). Nada hay en la dimensión que nos sea transparente en el sentido de encontrar una relación con el mundo real sino que son abstracciones que capta la propia técnica. Kintsch (1998) apunta que en el caso de comparar estas dimensiones con la formación del conocimiento humano, la elección del número de dimensiones que son requeridas, es una cuestión de la experiencia vital y el grado de ajuste que las personas consiguen para discriminar entre representaciones de su mundo. Al igual que el sistema implementado en una computadora, la mente debe explotar las coocurrencias de los eventos y realizar un análisis en el que no haya ni tantas dimensiones para tomar en cuenta información superflua e inútil ni un número de dimensiones tan pequeño que no permita captar o capturar la complejidad de su mundo. Las dimensiones en si no son entendibles sino es por su funcionalidad para discriminar eventos del medio. Aún así, Hu, et al. (2006) propusieron un método para, si cabe, intuir los significados, partiendo de una base de dimensiones cuyo significado es conocido y transformando el espacio semántico entero a esa nueva base.

2.6.3.2.- El número de factores como elemento empírico y a posteriori.

Al estar ordenados de mayor a menor aportación la matriz de valores singulares (la diagonal), tendremos simplemente que decidir qué número de dimensiones vamos a tomar. En otras palabras, donde vamos a poner el umbral de la aportación de cada factor para ser elegido. Idealmente, se pretende tomar suficientes dimensiones para capturar toda la estructura real de la matriz de términos–documentos, pero sin introducir ruido de factores o detalles irrelevantes que no aporten nada al análisis factorial. Esta decisión está aún abierta en la investigación. La manera común de realizar esto es haciendo la prueba de su funcionamiento, es decir, bajo el criterio que Deerwester, et al. (1990) llamaron “como funcione mejor”. Bajo este criterio se hace necesario comprobar bajo que dimensionalidad se ha trabajado más en los estudios realizados y que número de dimensiones ha resultado más fructífera. El propio Deerwester (1990) observó que la precisión en las pruebas a que se somete su LSA asciende más del doble (25% a 52%) si se pasa simplemente de 10 a 100 dimensiones. La simulación de Landauer y Dumais(1997) confirmaron que con la matriz sin dimensionar, es decir, tomando todas las dimensiones, sólo se consigue un 15% de precisión, mientras que se aumenta de 45% a 53% de precisión si se toma en torno a 300 dimensiones (véase la Figura 2.3). Si se excede este intervalo en torno a 300 dimensiones, la precisión vuelve a descender con una pendiente bastante acusada hasta el 15 % cuando se aplican todas las dimensiones. Olde, et al (2002) pusieron a prueba diferentes dimensionalidades (100, 200, 300, 400, 500) en un corpus específico de dominio que trataba sobre ciencias físicas y observaron que 300 es el número de dimensiones óptimo. De 300 a 500 no se encuentra mejoría en cuanto al rendimiento. Estos mismos resultados encontraron Kurby et al. (2003) quienes constataron que la correlación de los juicios entre humanos y modelo, tiene su pico en torno a 350 dimensiones, número este relativamente similar a los encontrados en las investigaciones antes citadas.

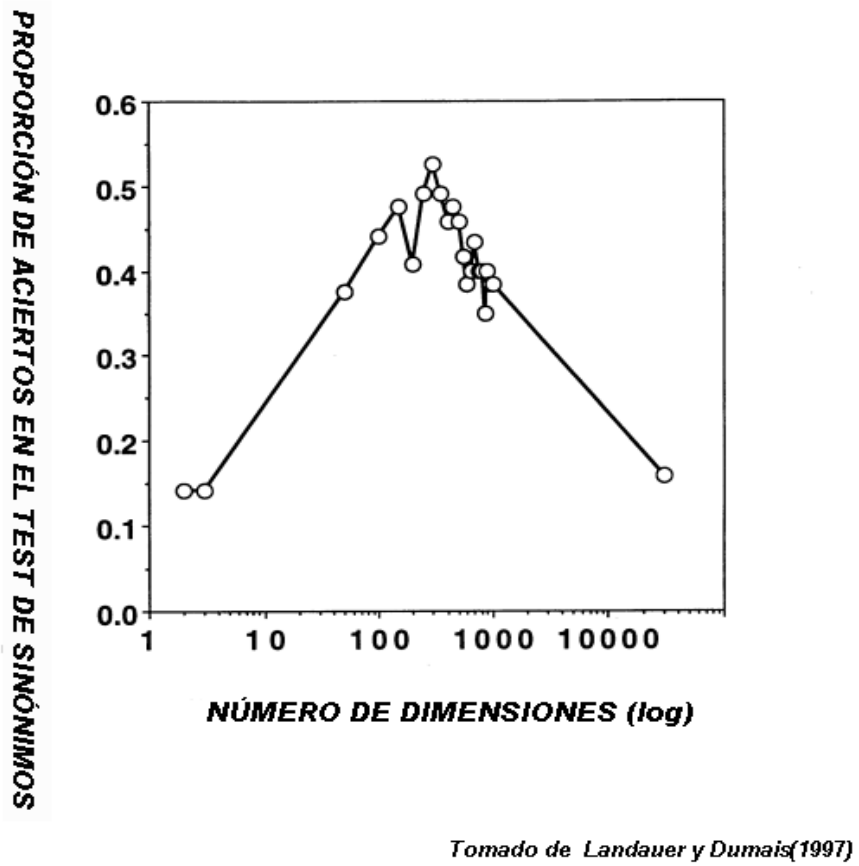


Figura 2.3.- Relación entre número de dimensiones y ajuste al criterio humano en un corpus de carácter general. Tomado de Landauer y Dumais (1997).

Este número de dimensiones (en torno a 300) suele ser eficiente para corpus que representan un uso general del lenguaje pero no está probado que funcione para otro tipo de corpus. Esta afirmación se sustenta por la extrema variabilidad de las dimensionalidades aplicadas en corpus específicos de dominio (Haley, Thomas, De Roeck y Petre, 2005) e incluso para otro tipo de aplicaciones de LSA que se salen del uso común (Pennenbaker, Mehl, y Niederhoffer , 2003). Este último autor no buscaba como de parecidos eran los términos y documentos respecto a su contenido sino como de parecidos eran las producciones escritas respecto a lo que él llama “estilo lingüístico”. Para constatar el estilo lingüístico se han de tener en cuenta palabras como pronombres, preposiciones, artículos, verbos auxiliares y su manera de combinarse en los textos. Para ello suprimió de los textos todo lo que no eran estas palabras. Al final, constato que el mejor funcionamiento lo daban 26

dimensiones. Este número queda muy por debajo de las 300 en los corpus de contenido general, pero téngase en cuenta que hay menos estilos que contenidos semánticos, en parte porque muchas menos palabras están asociadas a la especificidad de estilo en comparación con la masiva participación de todas las palabras en el espacio semántico.

2.6.3.3.- El porcentaje acumulado de valor singular

No obstante, el número de dimensiones que se tomen, dependerá de la combinación con otros parámetros, así como de los propósitos del espacio semántico (Wild, et al, 2005), lo que hace que el ajuste no sea ni mucho menos fácil tomando únicamente valores fijos. No resulta convincente que el número de factores sea igual para corpus generalistas que para corpus específicos de dominio, como tampoco lo es para textos con mucha o poca relación entre sus constituyentes (relaciones de primer, segundo y tercer *orden*)(véase a este respecto Kostostathis y Pottinger, 2006; Mill y Kontostathis, 2004). Los resultados de Wild et al. (2005) indicaron que la mejor proporción es que la matriz originaria y la factorizada compartan entre un 50%, 40% o 30% del porcentaje de valores singulares acumulado. Esta medida parece bastante eficiente para medir este efecto. Mas que insistir en el número de valores singulares que se toma, esta medida opta por el porcentaje acumulado que representan los valores singulares que se toman (recordemos que cada valor singular era representado por un número y estos tienen una sucesión descendiente). De esta manera el índice de esta medida se obtendrá del porcentaje del total que representa la suma de los valores singulares que se seleccionen.

2.7.- Consultas sobre la matriz factorizada

2.7.1.- La medida de la similitud (producto escalar y cosenos de los ángulos)

Las medidas de similitud entre términos y documentos fueron introducidas en uno de los modelos pioneros llamado modelo espacio-vectorial (*vector space model*) desarrollado por Salton (1960)**.** En él, cada documento estaba representado por un vector. Cada documento era descompuesto en patrones o características individuales que lo definían vectorialmente en un espacio multidimensional. Una vez y seguido este procedimiento, los documentos eran susceptibles de comparación mediante su producto escalar o el coseno del ángulo que dejan los dos vectores entre sí (véase la Figura 2.4). En los próximos apartados, desarrollaremos las medidas de similitud entre vectores. Esta forma de comparación se emplea en el análisis de la semántica latente una vez se cuenta con la matriz dimensionada, es decir, una vez se ha sometido la matriz a la descomposición del valor singular y se ha reducido a sus dimensiones más relevantes. LSA es una ampliación del modelo espacio-vectorial, pero comparando sólo dimensiones que marcan diferencias entre las relaciones de los términos y los documentos desdeñándose las dimensiones que no las remarcan. Además, en los modelos LSA, los patrones o características que definen las dimensiones de los vectores documentos son los términos que componen dichos documentos. A su vez, las dimensiones de los vectores que representan a los términos vienen definidos por los documentos en los que salen dichos términos.

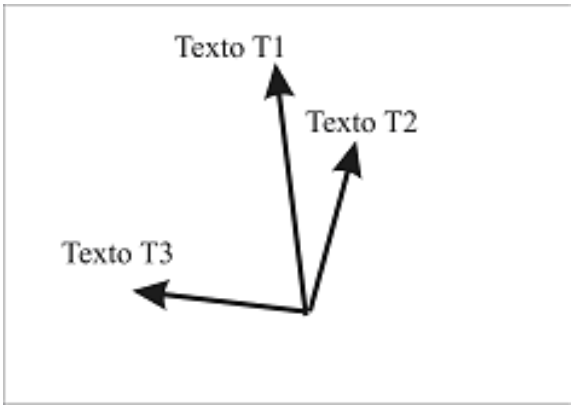


Figura 2.4*:* El resultado final del proceso es un espacio vectorial en el que están representados palabras y documentos y al que se le pueden integrar documentos nuevos. Como se puede ver en la figura, cuando se compara la similitud semántica entre tres textos dentro del espacio semántico definido por LSA, tenemos que los textos 1 y 2 son parecidos porque forman un ángulo cerrado y por lo tanto su coseno es próximo a 1. La relación semántica de los textos 1 y 2 con el tercero es casi nula. De esta manera, dos textos o dos palabras son susceptibles de comparación en base a medidas operativas lo que permite describir las relaciones de significado.

2.7.1.1.- Producto escalar

Una medida de similitud entre dos vectores, sean estos términos o documentos, puede ser el producto escalar de ambos vectores. De esta manera, se extraerán aquellos componentes (no nulos) que comparten. Conviene recordar aquí que el producto escalar entre dos vectores se haya multiplicando las componentes de los vectores dos a dos y sumando todos los productos resultantes. Es decir, el producto escalar de dos vectores es un escalar que se obtiene como la suma del producto de las componentes de los vectores.

$$V.W = (Vx\ Wx) + (Vy\ Wy) + (Vz\ Wz)... (Vn\ Wn)$$

De esta manera tendríamos la comparación de, pongamos por caso, dos documentos a lo largo de sus componentes. Retomando el ejemplo de la playa, imagínese de nuevo el lector que dispusiésemos de estos términos distribuidos en tres documentos, formando la siguiente matriz de términos-documentos en la qué no se ha realizado descomposición en vectores y valores singulares ni ningún tipo de proceso (Tabla 2.6). Simplemente contamos con datos brutos

que nos servirán para ejemplificar la manera más sencilla de comprobar la similitud entre dos términos o documentos teniendo en cuenta simplemente las apariciones en la matriz.

Tomando como medida el resultado del producto escalar de los vectores de los términos y documentos que quisiéramos comparar obtendríamos ya una medida primitiva de similitud (Tablas 2.7 y 2.8). En cuanto a los documentos obtendríamos que D1 y D2 tendrían 5 de similitud mientras D1 con D3 y D2 con D3 tendrían 1 y 0 respectivamente. Si examinásemos los documentos encontraríamos que la alta similitud entre D1 y D2 es debida a la coincidencia de las palabras que aparecen en ellos (mar, playa, gaviota y barca). En cuanto a los términos, Mar-playa tienen un 3 mientras Mar-Madre tienen 0 (son ortogonales ya que no comparten ni un solo término). Sin embargo, esta medida es algo imprecisa pues su puntuación es relativa al número de componentes comunes (Gracia, 2002). No se aporta gran información si nos dicen que hay una distancia de 5 sino poseemos la información de cuantas apariciones conjuntas se han producido.

Términos	*D1*	*D2*	*D3*
Mar	2	1	0
playa	1	1	0
madre	0	0	1
ciudad	0	0	1
cangrejo	1	0	0
cubo	0	1	0
agua	0	1	0
casa	0	0	1
coche	0	0	1
reloj	0	0	0
obra	1	0	1
pino	0	0	1
manguera	0	0	1
gaviota	1	1	0
barca	1	1	0

Tabla 2.6 . Martriz de ocurrencia brutas de 15 téminos en 3 documentos

	12	13	23
	2	0	0
	1	0	0
	0	0	0
	0	0	0
	0	0	0
	0	0	0
	0	0	0
	0	0	0
	0	0	0
	0	0	0
	0	0	0
	0	1	0
	0	0	0
	0	0	0
	1	0	0
	1	0	0
P.E.	5	1	0
COS.	0.68	0.12	0

Tabla 2.7. Comparación de los documentos de la tabla 6 empleando tanto el producto escalar como el coseno.

	Mar-Cangrejo	Mar-Madre	Mar-playa
	2	0	2
	0	0	1
	0	0	0
P.E.	2	0	3
COS.	0.66	0	0.95

Tabla 2.8. Comparación de los términos de la tabla 6 empleando tanto el producto escalar como el coseno.

2.7.1.2.- Cosenos

En el apartado anterior, se apuntaban los problemas que generaba el producto escalar como medida de la similitud. La manera de solucionar este problema es desarrollar la relación que tiene el producto escalar con los módulos de los vectores y el ángulo que dejan entre ellos estos mismos vectores. El producto escalar de dos vectores V y W relaciona el módulo de los vectores con el ángulo que forman entre ellos. Esto se expresaría de la siguiente forma:

$$V.W = |V|\ |W| \cos \alpha$$

Otra forma de calcular la similitud sería a partir del ángulo que forman los dos vectores. Si los dos vectores son perpendiculares, el ángulo será 0. Si se solapan será 1. Despejando la anterior ecuación se obtiene que:

$$\cos \alpha = V.W \ / \ |V|\ |W|$$

El coseno del ángulo entre los vectores es el producto escalar dividido entre el producto de sus módulos. El módulo viene representado por la raíz cuadrada de la suma de los cuadrados de sus componentes. Es equivalente decir que el ángulo es casi 0 a decir que su coseno es casi 1. Cuanto más se acerque el coseno a 1, mayor es la similitud entre los dos términos o los dos documentos. En nuestro ejemplo, los productos escalares entre los documentos son expresados de una manera más productiva (Tabla 7 y 8). Entre los documentos D1 y D2 el producto escalar de 5 pasa a expresarse como 0,68; la comparación D1 y D3 se expresa con el coseno como 0,12 y D2 con D3 permanece como 0. Respecto a los términos, *Mar-Cangrejo*, cuyo producto escalar es 2, se expresa como 0,66, así ocurre también con Mar-*Madre y Mar-Playa* que pasan a representarse como 0 y 0,95 respectivamente. La medida del coseno es la más utilizada en la técnica LSA (Haley et al.,2005) y es aplicada sobre la matriz ya factorizada o reducida aunque también se han probado otras como el coeficiente de correlación de Pearson, Spearman,

medidas de Minkowski, distancias euclídeas y de Manhattan (Deerwester et al.,1990; Laudauer et al., 1998; Nakov, 2000). Por ejemplo, Wild et al. (2005) obtuvieron mejores resultados por encima del coseno con la correlación de Spearman. Es importante recordar aquí que estas medidas se aplican sobre la matriz cuando ya ha sido sometida a los cálculos de entropía y se ha realizado la descomposición de los valores singulares.

2.7.1.3.- Distancia euclídea

Otra posible medida de la similitud entre dos vectores es el cálculo de la distancia entre ellos. Esta forma es tan sencilla cómo calcular el módulo del vector resta entre los vectores que representan los términos o documentos a comparar. En ocasiones, puede ser una buena alternativa al coseno sobre todo si se quieren comparar documentos que tienen mucha variabilidad en cuanto a su tamaño y contenido. En capítulos posteriores se profundizará en la conveniencia de su uso.

$$Dis(Vw1, Vw2) = \sqrt{\sum_{i=1}^{K} (Vw1_i - Vw2_i)^2}$$

2.7.1.4.- La longitud de vector como medida de la representatividad

Es importante añadir al análisis una segunda medida que puede resultar muy interesante. Esta medida es la *longitud del vector* y puede decirnos cuanta información posee el análisis LSA de una palabra representada por un vector. Los términos sobre los que el análisis aporta más información tendrán vectores con mayor longitud. Uno de los indicadores que muestra esta longitud del vector es que ese término está bien representado en los documentos y podría representar bien alguno de los conglomerados que se forman dentro del universo semántico. Como una propiedad que se deriva de la representación semántica mediante esta medida, la longitud de un vector frase será mayor que la de un vector término y, a su vez, ambas longitudes serán siempre menores

que la del vector párrafo (Kintsch,2001). Recordemos que cada una de estas estructuras textuales configuran una unidad contextual la cual queda mejor representada con el mayor número de términos que giren sobre ese contexto (la frase representa mejor un contexto que un único término). En otras palabras, el propósito de analizar las longitudes de los vectores es estimar cuan familiares son los términos o documentos dentro del espacio LSA, de tal manera que si la longitud es baja, indicará que el término será poco conocido por el LSA y, como consecuencia, podrá argumentarse que ese término aporta o transmite muy poca información sobre ciertos contenidos del corpus. Kintsch ejemplifica esta idea mediante un ejemplo con dos términos: “Pelicano” y “Pájaro”. Supongamos que estas dos palabras tienen una longitud de vector de 0,15 y 2,04, respectivamente. Estos datos indicarían que LSA posee mucha más información sobre el término “pájaro” que sobre el término “pelícano”. No pasa desapercibido para el lector que el término “pájaro” represente mejor ciertos contenidos del corpus de lo que pudiera hacer el término más restringido como es “pelícano”, un tipo de “pájaro” (al menos en un corpus de contenido general). Este efecto de la longitud de vector hace reflexionar a Kintsch (2002) en lo que se refiere a modelar la comprensión de predicaciones simples, ya que uno de los dos términos de la predicación puede verse arrastrado por el otro si este tiene mucha más longitud de vector. Fruto de esta reflexión, Kintsch (2002) propone un modelo más completo para modelar las predicaciones.

Por su parte, Rehder et al. (1998) llevaron a cabo un análisis más riguroso y demostraron que la interpretación de la longitud de vector depende del tipo de corpus de referencia, del propósito de la medición y de los procedimientos de depuración y ponderación de los términos que se hayan llevado a cabo. Estos mismos autores hicieron un interesante estudio sobre la aportación que tienen ciertas medidas LSA, entre ellas la longitud de vector, como predictor del conocimiento específico que tienen los alumnos sobre un tema concreto. El método es muy sencillo, comparar cada informe producido individualmente por cada alumno sobre un tema escogido de biología sobre “el corazón” (la extensión del informe o ensayo no puede sobrepasar las 250 palabras al que los autores denominaron -E-), con un texto estándar sacado de

un libro sobre este mismo tema (y al que los autores denominaron -C-). Con estos datos, Rehder et al. (1998) calcularon algunos índices como el coseno entre cada vector y el texto estándar (cos EC), el producto escalar (E·C) , la distancia euclidiana (Dis EC) y también la longitud o módulo del vector ||E|| y correlacionaron estas medidas con un cuestionario que aplicaron a los alumnos sobre el tema de biología y que servía de referencia para establecer el nivel de conocimiento de cada alumno. Se esperaba, por tanto, que estas correlaciones establecieran un índice de efectividad entre la medida LSA y el nivel de conocimiento de los alumnos. La correlación que estos autores encontraron más alta fue la realizada con el producto escalar, lo que sorprende por el uso extendido del coseno. Una vez que todas las variables introducidas en el análisis de la correlación son interdependientes se optó por realizar una regresión introduciendo esos mismos factores, pero con las siguientes variantes (dada la relación entre estas variables en las fórmulas).

Dada la fórmula:

$$E \cdot C = (\cos EC)\ (|E|)(|C|)$$

Y dado también que |*C*| es una constante, ya que se trata siempre del mismo texto (su aportación a la regresión es constante y por lo tanto prescindible). Se puede resolver que una fórmula derivada de este particular para los propósitos de la regresión podría ser:

$$E \cdot C' = (\cos EC)\ (|E|)$$

Esta nueva fórmula tendrá el mismo valor en la predicción del conocimiento. La conclusión a la que finalmente llegaron Rehder et al. (1998) fue que el producto escalar, E·C, puede verse afectado en función del coseno del ángulo entre los vectores (cos EC) y el módulo de E, (|*E*|*).* Además, si se predice el conocimiento con la fórmula de distancias euclidianas (Dis EC) pero con una transformación monotónica de esta, (Dis EC²), esta última, desarrollando las fórmulas, es equivalente a una predicción hecha con la combinación lineal de (cos EC) (|E|) y (|E|²).

Todo esto lleva a los autores a introducir todos estos posibles predictores en la regresión en busca de un predictor óptimo. Así pues introducen en la regresión: (cos EC) (|*E*|), (|*E*|), (|E|²), además de (cos EC) y E·C. Los resultados indicaron que las variables con más peso son (cos EC), el *coseno del ángulo* entre los dos vectores y (|*E*|), *el módulo o la longitud del vector de cada ensayo*. Estas dos variables serían los predictores con más peso. En otras palabras, aunque el producto escalar fuera la medida que más correlacionaba con el cuestionario, ello no implicaba predicciones adicionales a las proporcionadas por los componentes que lo describen: el coseno del ángulo entre los vectores y la longitud del vector del ensayo representado. Con esto, podemos tener un desglose mucho más útil para la predicción de la posesión del conocimiento. Si a esto añadimos que los resultados apuntan a que ambos factores son mutuamente independientes (no hay factor interacción), se llega a la conclusión de que (cos EC) y (|E|) resumen por ellos mismos la representación del conocimiento de las demás variables. El (cos EC) representa la dirección del vector dentro del espacio dimensional, es decir, la representación o "postura" de un objeto el dicho espacio, mientras que (|E|) crece según crece la representación de un vector en una o más dimensiones.

Además, según Rehder et al. (1998), es importante atender al protocolo de ejecución del propio análisis (el suyo con el tópico del corazón) para poder comprender en más profundidad que representa la longitud de vector y porque parece una medida eficiente de posesión de conocimiento:

1) El análisis está compuesto sólo de fragmentos que representan el tema del "corazón", por tanto, las palabras que no se usan en este tema, no pueden afectar a las medidas, incluido a la longitud del vector.
2) Las palabras que son raras en los textos (incluidas las técnicas), son ponderadas con un peso superior a las más frecuentes bajo la asunción que estas serán las que diferencien unos textos de otros. Las palabras ponderadas de una manera mayor incrementan la longitud del vector.
3) Previamente al análisis, se han obviado aquellas palabras de alta frecuencia en la lengua como las palabras de función. (lista de palabras "stop").
4) Los ensayos amplios tendrán vectores de mayor longitud.

Partiendo de estas observaciones los autores establecen un sumario: La longitud de vector es una función fuerte y positiva del número de palabras raras (técnicas) sobre el corazón, una función positiva y moderada de las palabras comunes sobre el corazón, y una función que no se relaciona con las palabras que no pertenecen al tema del corazón. La longitud del vector, por tanto, supone una medida que muestra cuán bien está representado el conocimiento en un vector o, en otras palabras, cuanto conoce un vector sobre un corpus de referencia. Pero si manejamos un corpus en el que se representa un tema específico, como es el caso de Rehder et al. (1998), la traducción inmediata de lo anterior sería cuán bien y cuanto está representado el conocimiento de ese tema en el vector.

Una aplicación práctica a la longitud del vector sería la posibilidad de establecer protocolos para la utilización de palabras familiares. Si deseamos introducir términos que transmitan mucha información y que resulten familiares para el receptor, podemos optar por desechar aquellas palabras que no rebasan un umbral arbitrario de longitud de vector. Una aplicación muy válida valdría para seleccionar títulos para los enlaces en una página web, como señalan algunos autores (Blackmon y Mandalia, 2004; Blackmon, Polson, Kitajima y Lewis, 2002). Ello conllevaría una mejora sustancial en la navegación web teniendo en cuenta las posibles dificultades del usuario. Según estos autores, la longitud del vector, en un corpus de conocimiento general, sería más o menos equivalente a la familiaridad. Tanto es así que la equiparan como procedimiento de medida a la frecuencia de uso en los corpus normativos. En suma, la interpretación de la longitud del vector, aunque haya ciertas interpretaciones inmediatas, viene definida por el tipo de depuración a que se somete a los términos, el tipo de corpus que se analiza y el propósito de la medición. Según como se haya realizado el análisis y las relaciones sobre las que se quiera indagar, así será la interpretación de la longitud del vector.

2.7.2.- Comparaciones

Una vez que la matriz resultante se haya sometido a los ajustes de entropía, a la descomposición de los valores singulares y, una vez reducidas las dimensiones, es el momento a partir del cual pueden llevarse a cabo las comparaciones entre las distintas unidades del corpus. Una primera comparación que puede llevarse a cabo es contrastar los términos o documentos entre sí. Esto nos dará una idea de lo aproximado que los términos y documentos se encuentran dentro del espacio semántico vectorial. Se trata, simplemente, de calcular el coseno del ángulo que dejan entre sí los dos términos vectores o las distancias euclídeas.

Ha habido dos formas de comparar términos y documentos que se desprenden de los trabajos empíricos, a saber: Una, sobre la matriz factorizada total X_i, y otra sobre las matrices $T_k S_k$ y $S_k D_k$ (es decir, cada una de las matrices por separado ponderando cada una de sus dimensiones por los valores expresados en la matriz diagonal)**.** Si bien la primera hace más sencillo el manejo de matrices en su implementación, pues sólo hay que manejar una sola matriz, la segunda es más económica en cuanto a recursos de memoria y velocidad y si cabe, más flexible a la hora de hacer correcciones (Kontostathis, Pottenger y Davison, 2005). En este escrito se mostrarán las dos formas, si bien se advierte que en la implementación de las aplicaciones, se prefirió esta segunda forma por las razones antes citadas.

Recordemos la anterior expresión:

$$X_¡ = Tk\ Sk\ Dk'$$

La primera forma tendrá como base de comparación la matriz factorizada X_i sobre la que se extraerán las medidas de similitud entre los vectores**.** Las comparaciones de términos y documentos se calcularán comparando filas y columnas de esta misma matriz y, cómo veremos también, los pseudodocumentos tendrán tantas dimensiones cómo términos o filas tenga

esta matriz**.** Por el contrario, la segunda forma tomará dos matrices para llevar a cabo estas comparaciones: para comparar términos, se compararán las filas de la matriz formada por la multiplicación de T_k y S_k. Esta nueva matriz tendrá vectores de una dimensionalidad menor a la matriz factorizada total (X_i). Para contrastar documentos, se compararán las columnas de la matriz formada por la multiplicación de $S_k D_k'$ (en este orden).

2.7.2.1.- Comparaciones tomando la matriz factorizada X_i

A).- *Comparaciones término con término.*

Supongamos, a modo de ejemplo, que deseamos poner a prueba la relación entre los términos “archivos” y “redundancia” extraidos ambos del ejemplo de corpus que se presentó previamente. También compararemos los términos “archivos” con “color”. Para ello, tendremos que recorrer la matriz de manera horizontal y dar con cada uno de los vectores que lo representan, esto es, los vectores de esos términos cuyas componentes se distribuyen a lo largo de los documentos.

	A1	A2	A3	A4	B1	B2	B3	B4	
archivos	0,301275326	0,248127069	0,229557343	0,21448577	-0,026364416	0,044351052	-0,009687197	-0,029748535	
planos	0,301370485	0,246768437	0,228380987	0,215097091	-0,019034531	0,04897961	-0,007019395	-0,01843675	
base	0,228025417	0,172039069	0,160047663	0,16829884	0,060526656	0,084177403	0,02196046	0,101648098	
datos	0,287215581	0,229251995	0,212504054	0,207235981	0,012121814	0,065709221	0,004324762	0,029117033	0,9979
elementos	0,258767425	0,216381999	0,200004998	0,182988624	-0,039312405	0,027612052	-0,014386928	-0,051265863	
redundancia	0,244612521	0,198865557	0,184128065	0,175127514	-0,008156059	0,044341663	-0,003042771	-0,00371208	
frutos	0,008464695	-0,03994268	-0,03432216	0,023772413	0,238831199	0,151898104	0,086923948	0,368768843	
largos	0,011268777	-0,04874006	-0,04183821	0,029970071	0,295303811	0,187977344	0,107477228	0,455994848	
recogida	0,044292864	-0,01375	-0,00990381	0,05053339	0,252624357	0,167817669	0,09193332	0,391347835	

Tabla 2.9. Comparación de los términos “archivos” y “redundancia” sobre la matriz Xi.

De esta forma, calculando el coseno del ángulo que quedaría entre estos dos vectores, se obtendría una medida de similitud entre los términos dentro del espacio semántico vectorial. En el caso de los términos “archivos” y “redundancia” nos encontramos que el coseno es 0,997, lo cual indica que los dos términos están próximos semánticamente, es decir, o bien aparecen juntos en los documentos (contextos), o bien no apareciendo juntos en ese están

asociado a otros términos que aparecen con ambos (Tabla 2.9). Por otro lado, tenemos nuestro segundo ejemplo (Tabla 2.10): “archivos” y “largos”. En este segundo ejemplo se obtiene un coseno de -0,087, lo cual está indicando que estos dos términos no tienen relación en el espacio semántico vectorial. Ambos términos representan a conglomerados diferentes. Mediante algoritmos basados en estas mediciones se pueden extraer listados de las palabras que forman un vecindario semántico.

	A1	A2	A3	A4	B1	B2	B3	B4	
archivos	0,301275326	0,248127069	0,229557343	0,21448577	-0,026364416	0,044351052	-0,009687197	-0,029748535	
planos	0,301370485	0,246768437	0,228380987	0,215097091	-0,019034531	0,04897961	-0,007019395	-0,01843675	
base	0,228025417	0,172039069	0,160047663	0,16829884	0,060526656	0,084177403	0,02196046	0,101648098	
datos	0,287215581	0,229251995	0,212504054	0,207235981	0,012121814	0,065709221	0,004324762	0,029117033	
elementos	0,258767425	0,216381999	0,200004998	0,182988624	-0,039312405	0,027612052	-0,014386928	-0,051265863	-0,0872
redundancia	0,244612521	0,198865557	0,184128065	0,175127514	-0,008156059	0,044341663	-0,003042771	-0,00371208	
frutos	0,008464695	-0,03994268	-0,03432216	0,023772413	0,238831199	0,151898104	0,086923948	0,368768843	
largos	0,011268777	-0,04874006	-0,04183821	0,029970071	0,295303811	0,187977344	0,107477228	0,455994848	
recogida	0,044292864	-0,01375	-0,00990381	0,05053339	0,252624357	0,167817669	0,09193332	0,391347835	

Tabla 2.10. Comparación de los términos “archivos” y “largos” sobre la matriz Xi.

B).- *Comparaciones documento con documento.*

Para comparar documentos se emplearía la misma estrategia que en las comparaciones de términos, pero esta vez moviéndose verticalmente hasta encontrar los vectores que representan a los dos documentos que queremos comparar (Tabla 2.11). Hemos tomado un ejemplo que puede dar cuenta de lo que estamos mostrando a lo largo de nuestra exposición. Los dos documentos que hemos introducido en la comparación no comparten ni un término en común, A2 “archivos planos elementos” con A4, “base datos redundancia”. Sin embargo, calculando el coseno del ángulo entre ambos resulta un 0,971 lo cual indica la alta relación entre los dos documentos.

	A1	A2	A3	A4	B1	B2	B3	B4
archivos	0,301275326	0,248127069	0,229557343	0,21448577	-0,026364416	0,044351052	-0,009687197	-0,029748535
planos	0,301370485	0,246768437	0,228380987	0,215097091	-0,019034531	0,04897961	-0,007019395	-0,01843675
base	0,228025417	0,172039069	0,160047663	0,16829884	0,060526656	0,084177403	0,02196046	0,101648098
datos	0,287215581	0,229251995	0,212504054	0,207235981	0,012121814	0,065709221	0,004324762	0,029117033
elementos	0,258767425	0,216381999	0,200004998	0,182988624	-0,039312405	0,027612052	-0,014386928	-0,051265863
redundancia	0,244612521	0,198865557	0,184128065	0,175127514	-0,008156059	0,044341663	-0,003042771	-0,00371208
frutos	0,008464695	-0,039942683	-0,034322162	0,023772413	0,238831199	0,151898104	0,086923948	0,368768843
largos	0,011268777	-0,048740058	-0,04183821	0,029970071	0,295303811	0,187977344	0,107477228	0,455994848
recogida	0,044292864	-0,013750005	-0,009903807	0,05053339	0,252624357	0,167817669	0,09193332	0,391347835
			0,9715					

Tabla 2.11. Comparación de los documentos A1 y A4 sobre la matriz Xi.

En el segundo ejemplo hemos querido introducir la comparación de ese mismo A2, pero esta vez con un documento sacado de lo que se supone otra temática B1 (Tabla 2.12). Ahora el coseno del ángulo entre los vectores que los representan se reduce a –0,141, lo cual induce a pensar que ambos documentos tienen poco que ver.

	A1	A2	A3	A4	B1	B2	B3	B4
archivos	0,301275326	0,248127069	0,229557343	0,21448577	-0,026364416	0,044351052	-0,009687197	-0,029748535
planos	0,301370485	0,246768437	0,228380987	0,215097091	-0,019034531	0,04897961	-0,007019395	-0,01843675
base	0,228025417	0,172039069	0,160047663	0,16829884	0,060526656	0,084177403	0,02196046	0,101648098
datos	0,287215581	0,229251995	0,212504054	0,207235981	0,012121814	0,065709221	0,004324762	0,029117033
elementos	0,258767425	0,216381999	0,200004998	0,182988624	-0,039312405	0,027612052	-0,014386928	-0,051265863
redundancia	0,244612521	0,198865557	0,184128065	0,175127514	-0,008156059	0,044341663	-0,003042771	-0,00371208
frutos	0,008464695	-0,039942683	-0,034322162	0,023772413	0,238831199	0,151898104	0,086923948	0,368768843
largos	0,011268777	-0,048740058	-0,04183821	0,029970071	0,295303811	0,187977344	0,107477228	0,455994848
recogida	0,044292864	-0,013750005	-0,009903807	0,05053339	0,252624357	0,167817669	0,09193332	0,391347835
				-0,1414				

Tabla 2.12. Comparación de los documentos A2 y B2 sobre la matriz Xi.

2.7.2.2.- Comparaciones tomando las matrices $T_k S_k$ y $S_k D_k$.

A).- *Comparaciones término con término.*

La diferencia entre el método anterior y éste radica en la matriz sobre la que comparamos los vectores. Si en el caso del primer método todas las comparaciones se hacían sobre $\mathbf{X_i}$, en este segundo caso, la matriz sobre cuyos vectores son comparados, dependerá de si lo que se desea comparar son términos o documentos. Si lo que se desea es comparar términos, entonces será $\boldsymbol{T_k S_k}$ la matriz sobre cuyos vectores descansará nuestro análisis. De la misma forma que en el anterior método, simplemente habrá que

extraer el coseno o la distancia entre dos vectores (Tabla 2.13 y 2.14). Compruébese ahora que los vectores a comparar tienen tantas dimensiones como factores a los que hayamos reducido las tres matrices, a raíz de calcular SVD. De esta forma, los cálculos pueden llegar a ser más económicos en clave de recursos de memoria.

TkSk

Término	*D1*	*D2*	
archivos	-0,499629572	-0,071033727	
planos	-0,500110908	-0,056418488	
base	-0,381701477	0,106773954	
datos	-0,477955675	0,006596246	
elementos	-0,428400386	-0,094258993	-0,0872
redundancia	-0,406245153	-0,031244259	
frutos	-0,024599908	0,475883032	
largos	-0,031750721	0,588358386	
recogida	-0,084763038	0,501203109	

Tabla 2.13. Comparación de los términos “archivos” y”largos” sobre la matriz TkSk.

TkSk

Término	*D1*	*D2*	
archivos	-0,49962957	-0,07103373	
planos	-0,50011091	-0,05641849	
base	-0,38170148	0,106773954	0,9979
datos	-0,47795567	0,006596246	
elementos	-0,42840039	-0,09425899	
redundancia	-0,40624515	-0,03124426	
frutos	-0,02459991	0,475883032	
largos	-0,03175072	0,588358386	
recogida	-0,08476304	0,501203109	

Tabla 2.14. Comparación de los términos “archivos” y “redundancia” sobre la matriz TkSk.

B).- *Comparaciones documento con documento*

De la misma forma, la comparación Documento-Documento se calculará extrayendo el coseno o la distancia de dos columnas de la matriz *SkDk (Tabla 2.15).*

SkDk

	A1	A2	A3	A4	B1	B2	B3	B4
D1	-0,66699708	-0,53389281	-0,49480239	-0,48069262	-0,02047087	-0,14776654	-0,007248274	-0,055770459
D2	-0,01227321	-0,1005129	-0,08792011	0,025459758	0,462735873	0,288503896	0,168424008	0,713450447
			0,9715					

SkDk

	A1	A2	A3	A4	B1	B2	B3	B4
D1	-0,66699708	-0,53389281	-0,49480239	-0,48069262	-0,02047087	-0,14776654	-0,007248274	-0,055770459
D2	-0,01227321	-0,1005129	-0,08792011	0,025459758	0,462735873	0,288503896	0,168424008	0,713450447
				-0,1414				

Tabla 2.15. Comparación de documentos sobre la matriz SkDk.

2.7.3.- El pseudodocumento: caso especial de documento

Otra utilidad de los sistemas basados en LSA es la posibilidad de introducir consultas o pseudodocumentos. Esta consulta es una frase, párrafo o texto que no se encuentra representado en la matriz en forma de vector-documento, al que se quiere encontrar similitudes con los documentos que ya se encuentran insertos o se quiere encontrar similitud con otra consulta introducida. Por esta razón también recibe el nombre de pseudodocumento pues es un tipo de documento al que se le ha de introducir en el espacio semántico vectorial que representa la matriz factorizada.

2.7.3.1.- ¿Qué es un pseudodocumento?

Supongamos que en nuestro ejemplo realizamos la siguiente consulta: “los elementos de la base constituyen archivos” (para una mayor ejemplificación hemos subrayado los términos que se introducen en el análisis). No hay un documento en nuestro corpus textual que sea igual a nuestra consulta y por tanto, este nuevo documento no está representado en forma de vector en la matriz. Si se diera el caso de que fuera igual que uno ya existente, podríamos realizar simplemente la consulta con una simple comparación de la representación de su vector-documento con todos los demás. Al no cumplirse esto, hay que calcular un vector cuyas componentes sean el reflejo de los términos que contiene y que sea sometido al mismo proceso que la matriz que representa el espacio semántico (preproceso, reducción de dimensiones, etc.).

Lo primero que se ha de hacer es un vector documento en el que se expresen las ocurrencias de los términos.

	A1	A2	A3	A4	B1	B2	B3	B4	PSEUDODOC
archivos	1	1	1	0	0	0	0	0	1
planos	1	1	0	0	0	0	0	0	0
base	1	0	0	1	0	1	0	0	1
datos	1	0	0	1	0	0	0	0	0
elementos	0	1	1	0	0	0	0	0	1
redundancia	0	0	1	1	0	0	0	0	0
frutos	0	0	0	0	1	0	1	1	0
largos	0	0	0	0	1	0	0	1	0
recogida	0	0	0	0	0	1	0	1	0

Tabla 2.16. El pseudodocumento es un documento más en el que se consignan las ocurrencias de los términos representados en el corpus.

En nuestro caso tendríamos que representar nuestros tres términos (“elementos”, base” y “archivos”) en un vector. Lo único que hay que hacer es consignar el número de ocurrencias de cada término en el Pseudodocumento. Nuestro nuevo pseudodocumento quedaría como muestra la Tabla 2.16.

Esta sería la expresión que tendría el pseudodocumento si sólo se atendiese a las coocurrencias de los términos. Pero la potencia de la LSA reside en la transformación de esa matriz simple de ocurrencias en un espacio semántico cuyas dimensiones hayan sido reducidas sustancialmente. El producto final sería una matriz términos y documentos en que las ocurrencias que representan las componentes de los vectores son inferidas por el propio sistema. Esto quiere decir que nuestro documento ha de ser introducido en la matriz factorizada y pueda reflejar así todas las interacciones a las que están sujetos los demás documentos.

2.7.3.2.- Ajustes y cálculos de entropía sobre el pseudodocumento

Como si de un corpus se tratara, antes de introducir el pseudodocumento en el espacio semántico que la matriz representa, hay que realizar los ajustes que se hicieron con este. Lo primero es desechar los términos que no se introdujeron en el análisis del corpus, a saber, pronombres, artículos, preposiciones, conjunciones, verbos comunes y pronombres (véase Yu, Cuadrado, Ceglowski y Payne, 2004). Una vez realizado este paso, se realizan los cálculos de entropía (Burek y Vargas-Vera, 2004). Los algoritmos de estos

cálculos están desarrollados en capítulos anteriores. Recordemos que este tipo de ajuste da cuenta de la importancia de un término para ser distintivo de la información que porta un documento. Se infiere de esta forma que los términos que salen en muchos documentos obedecen a una cuestión de pura contingencia. El peso que tienen las ocurrencias de cada término en nuestro pseudodocumento se calculará con relación al peso local y global respecto a la matriz. En nuestro ejemplo quedaría como muestra la Tabla 2.17:

	PSEUDODOC	PSEUDODOC
archivos	1	0,326943084
planos	0	0
base	1	0,326943084
datos	0	0
elementos	1	0,46209812
redundancia	0	0
frutos	0	0
largos	0	0
recogida	0	0

Tabla 2.17. Cálculo de la función de entropía sobre los términos representados en los documentos. Como peso global podrán emplearse los mismos que se utilizaron en los cálculos de la matriz principal. La columna de la derecha muestra el pseudodocumento con los cálculos de entropía.

2.7.3.3- Introduciendo el pseudodocumento en el espacio vectorial

2.7.3.3.1 .- Cálculos sobre X_i.

Una vez culminados los pasos anteriores, podemos introducir el pseudodocumento en el espacio semántico que la matriz representa. Para ello, se transforma en un vector pseudodocumento que obedece a la siguiente expresión (Alaniz-Macedo, Campos-Pimentel y Camacho-Guerrero, 2002).

$$V_{pq} = T_k T'_k V_q$$

Donde:

Vpq es el vector pseudodocumento resultante.

Tk es la matriz reducida de términos.

T'k es la transpuesta de la matriz reducida de términos.

Vq es el vector del pseudodocumento de ocurrencias (o vector query).

Una vez se han realizado estos cálculos, el vector ***Vpq*** es susceptible de compararse con los documentos existentes en la matriz (o con otro pseudodocumento) por medio de la medida de similitud deseada. En este caso, el coseno del ángulo que dejan ambos vectores entre sí. De esta manera, el nuevo pesudodocumento se podrá comparar con los documentos ya existentes en el espacio semántico (Figura 2.18).

	A1	A2	A3	A4	B1	B2	B3	B4	PSE FINAL
archivos	0,301275326	0,248127069	0,229557343	0,21448577	-0,026364416	0,044351052	-0,009687197	-0,029748535	-0,009687197
planos	0,301370485	0,246768437	0,228380987	0,215097091	-0,019034531	0,04897961	-0,007019395	-0,01843675	-0,007019395
base	0,228025417	0,172039069	0,160047663	0,16829884	0,060526656	0,084177403	0,02196046	0,101648098	0,02196046
datos	0,287215581	0,229251995	0,212504054	0,207235981	0,012121814	0,065709221	0,004324762	0,029117033	0,004324762
elementos	0,258767425	0,216381999	0,200004998	0,182988624	-0,039312405	0,027612052	-0,014386928	-0,051265863	-0,014386928
redundancia	0,244612521	0,198865557	0,184128065	0,175127514	-0,008156059	0,044341663	-0,003042771	-0,00371208	-0,003042771
frutos	0,008464695	-0,039942683	-0,034322162	0,023772413	0,238831199	0,151898104	0,086923948	0,368768843	0,086923948
largos	0,011268777	-0,048740058	-0,04183821	0,029970071	0,295303811	0,187977344	0,107477228	0,455994848	0,107477228
recogida	0,044292864	-0,013750005	-0,009903807	0,05053339	0,252624357	0,167817669	0,09193332	0,391347835	0,09193332

Tabla 2.18. Pseudodocumento final calculado sobre la matriz X_i. La columna de la derecha representa el vector ***Vpq.*** La matriz representada corresponde también a la matriz factorizada X_i *y con cálculos de entropía.*

2.7.3.3.2.- Cálculos sobre $T_k S_k$ y $S_k D_k$

Otra alternativa a la anterior (la mejor) en este caso de que se haya optado por realizar las consultas sobre las matrices $T_k S_k$ y $S_k D_k$ en vez de la matriz factorizada X_i, es la fórmula propuesta en el artículo de Berry, Dumais y O´Brien (1994). Esta forma de representar el pseudodocumento tiene la gran ventaja de hacerse contando con un vector del tamaño de los vectores que representan los documentos en la matriz $S_k D_k$ (es decir, la matriz de documentos reducida multiplicada por la diagonal reducida). Esto, como es obvio, tiene la ventaja de que por muchas filas y columnas que tengamos en la matriz principal X_i (tanto en la matriz factorizada como en la de ocurrencias brutas), nuestros pseudodocumentos serán siempre representados por un vector que posea tantas dimensiones cómo factores hayamos nosotros elegido para reducir las matrices. Por muchas que sean las filas y columnas de estas

matrices, nuestros vectores de pseudodocumentos (al igual que documentos y términos), no tendrán nunca un número superior a 300 o 400 dimensiones, cifra esta empíricamente considerada cómo dimensionalidad máxima para corpus generalistas. Al ser los vectores que representan a estos pseudodocumentos de menor dimensionalidad, utilizarán menos recursos en memoria y menos tiempo de computación. Al igual que el anterior procedimiento, se procederá de un vector que represente las ocurrencias de los términos en un vector documento del mismo tamaño que la matriz de ocurrencias brutas, este vector se transpondrá y se multiplicará por la matriz reducida de términos y la inversa de la diagonal reducida. El vector de ocurrencias del cual se parte habrá de ser ponderado de la misma forma que se ponderó la matriz de ocurrencias bruta, aplicando los pesos locales y globales a los que esta fue sometida.

La fórmula es la siguiente:

$$\hat{d} = d^T U_k \Sigma_k^{-1}.$$

Donde:

d́= pseudodocumento introducido ya en el espacio vectorial (en Dk).

***d*`=** el vector transpuesto del pseudodocumento de ocurrencias.

∑¯ı = Inversa de la matriz de valores singulares factorizada

U = Matriz de términos factorizada.

Siguiendo con el ejemplo anterior, en lugar de hacer las consultas sobre la matriz factorizada total $\mathbf{X_j}$, se aplicaría ahora sobre las matrices $\boldsymbol{T_k S_k}$ y $\boldsymbol{S_k D_k}$.

De esta manera, $\boldsymbol{S_k D_k}$ cada documento estaría representado por un vector de dos dimensiones, número que fue elegido por nosotros. Aplicando la fórmula de arriba se obtienen pseudodocumentos que tendrán este mismo número de dimensiones por lo que serían susceptibles de compararse (Tabla 2.19). En el ejemplo, se extraen índices de similitud entre el pseudodocumento "archivos base planos" y cada uno de los documentos existentes en el espacio semántico aunque la forma más útil de emplear el pseudodocumento no es compararlo con los documentos ya existentes sino con otro pseudodocumento. En este

apartado, para dejar patente que el pseudodocumento es un tipo especial de documento, hemos preferido relacionarlo con los documentos existentes en la matriz.

SkDk

	A1	A2	A3	A4	B1	B2	B3	B4	pseudo
D1	-0,666997077	-0,53389281	-0,49480239	-0,48069262	-0,02047087	-0,14776654	-0,007248274	-0,055770459	-0,4679523
D2	-0,012273213	-0,1005129	-0,08792011	0,025459758	0,462735873	0,288503896	0,168424008	0,713450447	-0,0155728
	0,9998	0,9883	0,9898	0,9962	0,0109	0,426	0,0097	0,044	

Tabla 2.19. Pseudodocumento final calculado sobre la matriz SkDk. Esta alternativa es mucho más eficiente en cuanto a empleo de memoria RAM pues no necesita tener instanciada la matriz principal sino sólo la matriz SkDk, de tamaño mucho más reducido.

2.7.3.3.3.- Cálculo del centroide simple: Una alternativa al método de introducción en el espacio vectorial (folding-in)

Otro método que se utiliza para construir un pseudodocumento, acaso por su simpleza es el llamado “Centroide”. Este método consiste en la representación de una o más palabras como la suma de todos los vectores que representan esas palabras. Como se podrá adivinar, esto no resultará un vector con las mismas dimensiones que un vector documento sino que resultará otro vector término con las mismas dimensiones que los vectores que representan a las palabras que componen el pseudodocumento.

Siguiendo con el anterior ejemplo, el pseudodocumento resultante se calcularía sumando los vectores que representan a los términos (elementos, base, archivos).

archivos	0,000	0,900	0,754	0,609	0,647	-0,112	0,263	-0,084	-0,079
elementos	0,000	0,537	0,457	0,368	0,382	-0,108	0,141	-0,080	-0,091
base	0,000	0,750	0,583	0,477	0,565	0,162	0,320	0,114	0,210
Pseudo	0,000	2,187	1,794	1,454	1,593	-0,058	0,725	-0,050	0,040

Tabla 2.20. El pseudodocumento es un término más que representa la suma vectorial de los términos insertos en él.

Como se puede comprobar, el vector resultante adopta una forma de vector término por lo que sus dimensiones serán igual al número de documentos que estén representados en la matriz (Tabla 2.20). El hecho de que un documento

se represente en forma de términos entrañará algunos problemas en algunos tipos de espacios semánticos. Aún así, este vector se podrá comparar con otros pseudodocumentos construidos de esta misma forma o bien con términos provenientes de la matriz general. Este método es recomendado para estructuras que pueden constar de dos términos cómo por ejemplo “fobia social”, “fuegos artificiales”, etc., aunque esta manera entrañe el riesgo de la diferencia de longitud de vectores entre estos dos términos y su consiguiente arrastre del contenido hacia los contenidos de sólo uno de los dos. Seguramente “perro caliente” devolverá cómo contenido ejemplares que tendrán que ver con el mundo de los canes (véase Kintsch, 2001; Kintsch y Bowles, 2002, para un análisis en profundidad de esta problemática y una tentativa de solución).

2.8.- Factores adicionales

Ofrecemos en este apartado la definición de algunos conceptos y proponemos algunas ideas para optimizar el proceso de LSA. Se revisan aquí algunos aspectos técnicos y metodológicos referentes al corpus, a las medidas de similitud y al tamaño de las matrices y la capacidad de memoria de los PCS actuales.

2.8.1.- Estructura del corpus

La naturaleza del corpus es el factor más decisivo para llevar a cabo modelos y aplicaciones basadas en LSA. El análisis que puede generar LSA dependerá estrechamente de la estructura y distribución del corpus sobre el que llevamos a cabo el análisis. Quesada (2006) propone dos errores frecuentes que los autores cometen a la hora de crear LSA. A saber: el primero de ellos el poner a prueba corpus excesivamente reducidos. Para Quesada, una regla útil aunque informal y complementaria es que un corpus es suficientemente grande cuando podemos interpretar cualquier nueva producción en función de la información ya procesada en el análisis, sin necesidad de añadir ningún texto más al análisis. El segundo error es utilizar corpus poco representativos de dominio que se quiere analizar lo que

provocaría relaciones que no se ciñen a la realidad que buscamos. No es fácil evitar estos errores, es decir, no es fácil diseñar corpus de un tamaño correcto y además que sean representativos de un dominio sobre todo cuando se están analizando corpus que no son de tamaño ilimitado. Aún así, podemos contar con algunos índices interesantes que nos pueden dar algunas pistas y algunas manipulaciones que ayuden a diseñar corpus equilibrados.

2.8.2.- Ley de Zipf

La ya clásica ley de Zipf (1932) define, mediante una función, cuales deben ser las frecuencias de los términos en una relación de orden desde los más frecuentes hasta los menos frecuentes, dado el número de términos en un corpus (Figura 2.5). Esta ley explica que si dispusiésemos todas las palabras en orden de mayor a menor frecuencia asignándose a la más frecuente el número 1 y a la menos frecuente n, se cumple la regla de que la frecuencia de una palabra es inversamente proporcional a su número de orden en el "ranking" y que cuanto menor sea el número de términos total, mayor será la frecuencia de las palabras en los primeros rangos. Esta ley se suele cumplir en todas las lenguas. En el caso que nos ocupa, conociendo la distribución teórica de las frecuencias de cada rango (f del término más frecuente, f del segundo más frecuente, f de término n más frecuente), podemos comprobar si nuestro corpus se ajusta a los textos de lenguaje natural en cuanto a la distribución de sus frecuencias (Quesada, 2006). La fórmula se expresa de la siguiente forma:

$$f(k; s, N) = \frac{1/k^s}{\sum_{n=1}^{N} 1/n^s}$$

Dónde

N: es el número de términos.

K: es el rango del término (el orden que ocupa en el ranking de frecuencia).

S: Es el exponente. En la versión clásica es 1.

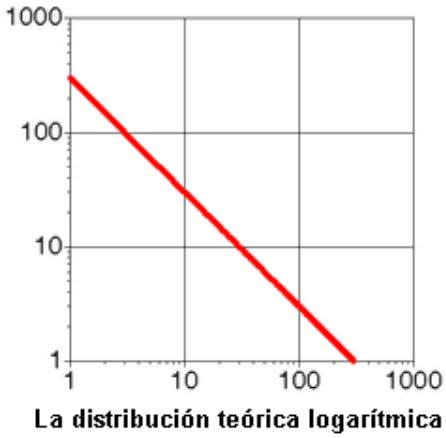


Figura 2.5. Representación gráfica de la proporción de la ley de Zipf.

2.8.3.- Marcado de documentos

En ciertas ocasiones podemos optar por imprimir artificialmente más fuerza a las relaciones entre los términos pertenecientes a documentos sobre un determinado sentido. Esto puede llevarse a cabo adjuntando a los documentos alguna información codificada. No se trata de codificación morfológica ni sintáctica sino simplemente, marcar al documento en lo que respecta a su tipo (por ejemplo, un documento que muestra una conversación sobre cómo arreglar un problema del enrutador podría un acompañamiento cómo: Rout-probl). Esta información ayudará a dar mayor estructura y consistencia a nuestro espacio semántico. El procedimiento es o introducir en los documentos algunas cadenas de caracteres (vocablos en forma de claves) que los identifiquen o bien introducir estas marcas o Tags como términos. Un ejemplo interesante es el propuesto por Serafín y Di´Eugenio(2004). En su experiencia emplean *FLSA* (Featured Latent Semantic Analysis) para llevar a cabo clasificaciones de diálogos. Para ello prueban el comportamiento de tres corpus marcados previamente. CallHome, un corpus de llamadas telefónicas en español, MapTask que contiene diálogos en torno a las instrucciones en torno a un mapa y Diag-NLP que versa sobre diálogos sobre el aprendizaje del uso de ordenadores. Todos estos corpus están marcados con Tags que aluden a varios criterios. El método *FLSA* computa dos matrices que luego concatena: la matriz términos documentos y la matriz TAG-documentos. La matriz TAG-

Documentos está formada por los TAGs de los propios corpus e identifica a cada documento. Esta matriz resultante, (w+t)*D es tratada de la misma forma que se trataría en la forma clásica de LSA. Los TAGs son tratados como términos, ocupando también filas en la matriz de datos. De esta forma se consigue forjar más cohesión entre los propios documentos y términos en coalición con los TAGs artificiales. Ambos casos LSA y FLSA se comportan de una forma muy efectiva a la hora de categorizar diálogos pero los resultados muestran que FLSA se comporta mejor. Son interesantes también las aportaciones que provienen del marcado de lo que se ha venido a llamar "actos de diálogos" (véase para una revisión Torres-Goterris, 2006) que provienen del diseño y análisis de sistemas artificiales de diálogo.

2.8.4.- Tagging

Uno de los problemas con los que se enfrentan este tipo de técnicas es que los corpus que son sometidos a entrenamiento (sobre todo si no llegan a un mínimo de alcance muestral) es que sus términos pueden no desplegar una muestra de usos lo suficientemente representativa y quedar adscritos a un ámbito sesgado. Este tipo de sesgos suele provenir, cómo se indicó en anteriores capítulos, o bien de palabras con una gran variabilidad flexiva(cómo los verbos) o bien de sinónimos no representados una forma equitativa en todos sus ámbitos o bien por nombres propios que hacen referencia a objetos o eventos y que son de uso general. Un ejemplo clásico en este tipo de problemática es el de las fechas y las ciudades. Imagínese que tenemos un ámbito semántico que esta irremisiblemente asociado con fechas genéricas y ciudades de salida y destino. Podemos pensar por ejemplo salidas de aviones o trenes. Podríamos tener un corpus en el que se reflejasen algunas frases relacionadas con fechas de salidas y llegadas y sus respectivos destinos. En estas frases, no están representadas todas los meses ni las ciudades posibles y es probable que debido a la poca ocurrencia de los que sí están, estos queden asociados espuriamente a vocablos cómo avión, tren, viaje. Existe una amplia gama de estas estructuras de este tipo que pueden sufrir este efecto. Una posible solución es eliminar este tipo de términos, pero se correrá el riesgo

de prescindir de vocablos altamente significativos del ámbito que se analiza. Otra posible solución es agrupar ese tipo de términos en una categoría superior. De esta manera, se sustituirán las ocurrencias de las ciudades concretas (Madrid, Barcelona, Vigo, Burgos, etc) por el nombre de la clase a la que pertenecen (ciudad). Este tipo de tratamiento del texto recibe el nombre de *tagging* y es en cierta manera parecida a la lematización o al *stemming* pero esta vez de una manera categorial y no gramatical. La mayoría de los paquetes que llevan a cabo algún tipo de Modelo estadístico del lenguaje (SLM: *Statistical Language Model*) para interpretar el lenguaje natural, ofrecen un tipo de tagging o modelo del lenguaje basados en clases. Este tipo de estructuras son formalizadas por medio de un lenguaje específico que será compilado e interpretado por la propia aplicación y que llevará a cabo el análisis teniendo en cuenta las clases a los que pertenecen ciertos términos. De esta forma se reduce la variabilidad de algunos términos sustituyéndolos por una clase o categoría superior que las represente o se introduce en el análisis cada una de las posibles acepciones de esta clase. Veamos un ejemplo de tagging:

```
Month
        [
                january
                february
                march
                april
                may
                june
                july
                august
                september
                october
                november
                december
        ]
```

2.8.5.- Bases de datos y diseño artificial del corpus

Una forma de diseñar artificialmente los corpus es hacer un gran acopio de párrafos especificando sobre ellos todas las propiedades que se estimen oportunas cómo, por ejemplo, la materia a la que pertenecen, libro del que fueron extraídos, tema al que pertenecen, bloque del que forman parte, volumen, tipo (expositivo, narrativo, taxonómico, etc.) o cualquier otra cualidad que los puedan representar. De esta manera, podíamos procurarnos una Base de Datos (desde ahora BBDD) en la que estuviesen representados de manera distribuida y relacional todos los párrafos, los bloques, los temas, los libros y materias de las que fueron extraídos y junto al tipo de párrafos que tenemos. La principal razón que mueve a realizar todo esto es nos permite diseñar corpus a medida dependiendo de las necesidades que se tengan y del éxito de las pruebas. Es sabido el coste de tiempo que entraña la elaboración de corpus. Si a esto añadimos que no resulta suficiente crear un solo corpus, sino varios de los que se someten aprueba pasado por diferentes composiciones, esto puede resultar infinitamente más costoso. Si automatizamos el proceso de creación de los corpus y facilitamos el protocolo de decisión de las composiciones, habremos ahorrado gran cantidad de tiempo y quizás recursos económicos.

Ponemos aquí un ejemplo de una tentativa de automatización en la creación de los corpus creado por nosotros. En este caso, valiéndonos de la arquitectura Cliente-Servidor, diseñamos e implementamos una aplicación WEB que se ocupa de almacenar los párrafos indexados en propiedades que los definen en términos de que textos proceden (bloque, tema, libro, materia), el tipo de párrafo (expositivo, narrativo, taxonómico, etc.), las fechas, los colegios en dónde se emplean, las Comunidades Autónomas, etc. (Figura 2.6). Todos estos datos son almacenados en una BBDD *sql server*. La parte que se ocupa de la gestión, es decir, de la inserción de los párrafos y su clasificación y de la selección y construcción de corpus a medida, está implementado en ASP.NET junto con VB.NET. Por medio de consultas restringidas en *Transat SQL*, se puede ir configurando los párrafos (sub-bloques) que formarán parte de nuestro corpus que será posteriormente procesado por LSA.

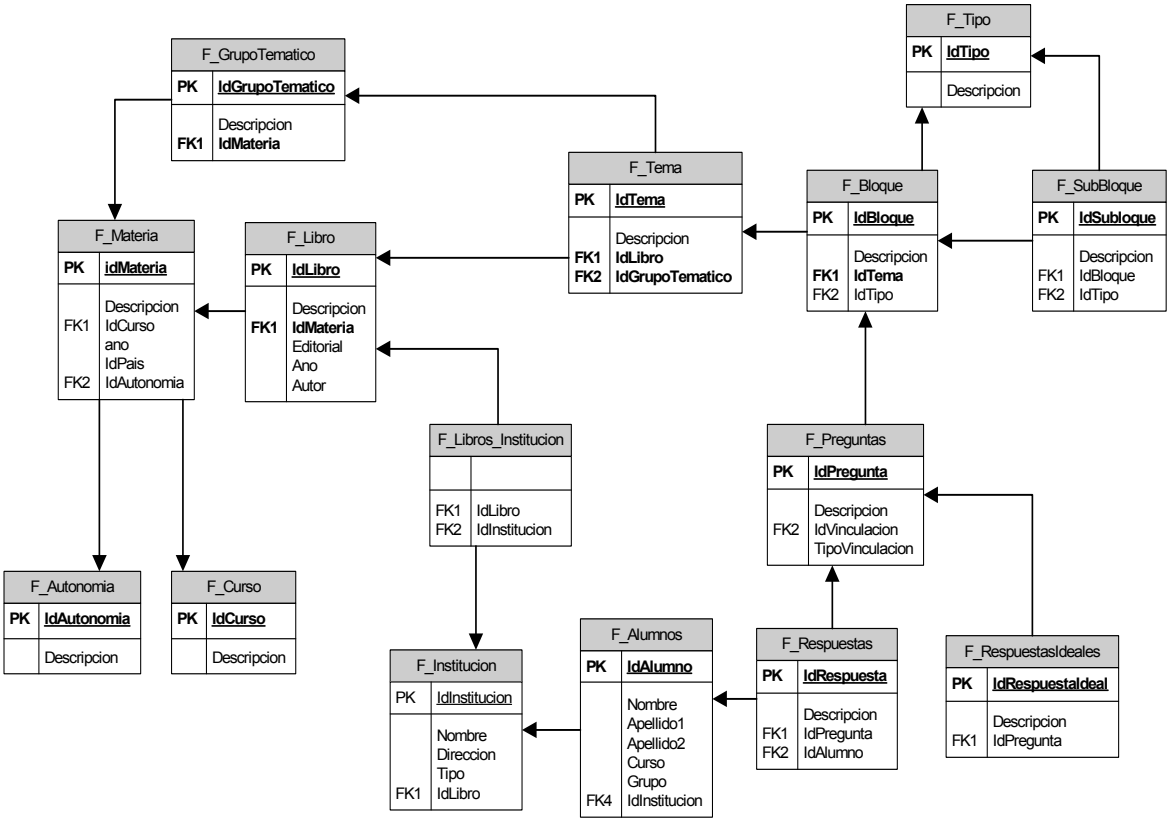


Figura 2.6. Esquema entidad-Relación de un gestor de corpus. El modelo de datos muestra como cada párrafo puede ser identificado y recuperado atendiendo a entidades superiores como materia, libro, etc.

La tabla F_SubBloque es la contiene todos los párrafos de todos los libros y textos que se guardan en la BBDD. Cómo se ve, cada SubBloque pertenece a un bloque que a su vez pertenece a un tema, etc. Debido a esta distribución de las propiedades de los datos, es posible el diseño de corpus que contengan párrafos seleccionados por sus diversas características. En un modelos así, sería posible diseñar un corpus con los párrafos contenidos únicamente en materias impartidas en algunas de las Autonomías. Júzguese la potencia de este tipo de diseños de datos a la hora de llevar a cabo comparativas de diversa índole y ajustes en sistemas aplicados a la industria lingüística. Contra la BBDD en SQL server 2000 se implementó parte de una aplicación que lleva a cabo las funciones de clasificación y diseño de corpus. El lenguaje de programación fue ASP.NET con VB.NET, el modo de acceso a datos fue implementado con una arquitectura en tres capas: capa de Interfaz de usuario (implementación de pantallas, funcionalidades e instanciación de objetos), capa de gestión de datos (definición de clases para cada tipo de entidad e

implementación de métodos de acceso a BBDD) y BBDD (base de datos en SQL Server). El prototipo de la aplicación quedó de la siguiente manera:

1) *Pantallas para la gestión y mantenimiento de los datos*.

a) En esta primera pantalla se gestionan las posibles materias que pueden ser impartidas en los cursos, instituciones y autonomías (Figura 2.7). A partir de aquí introduciremos los libros que se emplean.

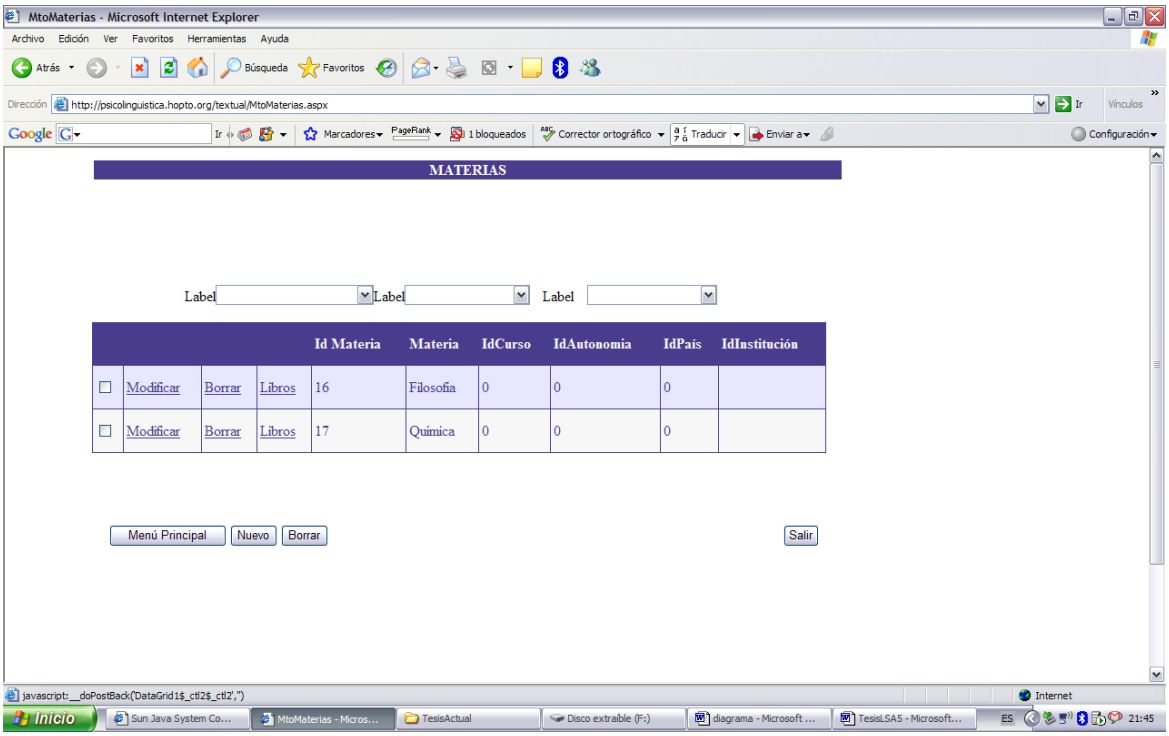


Figura 2.7. .Pantalla de gestión de materias

b) Las pantallas de los libros se insertan para cada materia que se da en un curso, institución y autonomía (Figura 2.8). De la materia filosofía tendríamos disponibles dos libros. Aún así, cabe la posibilidad de introducir más en el botón “nuevo”.

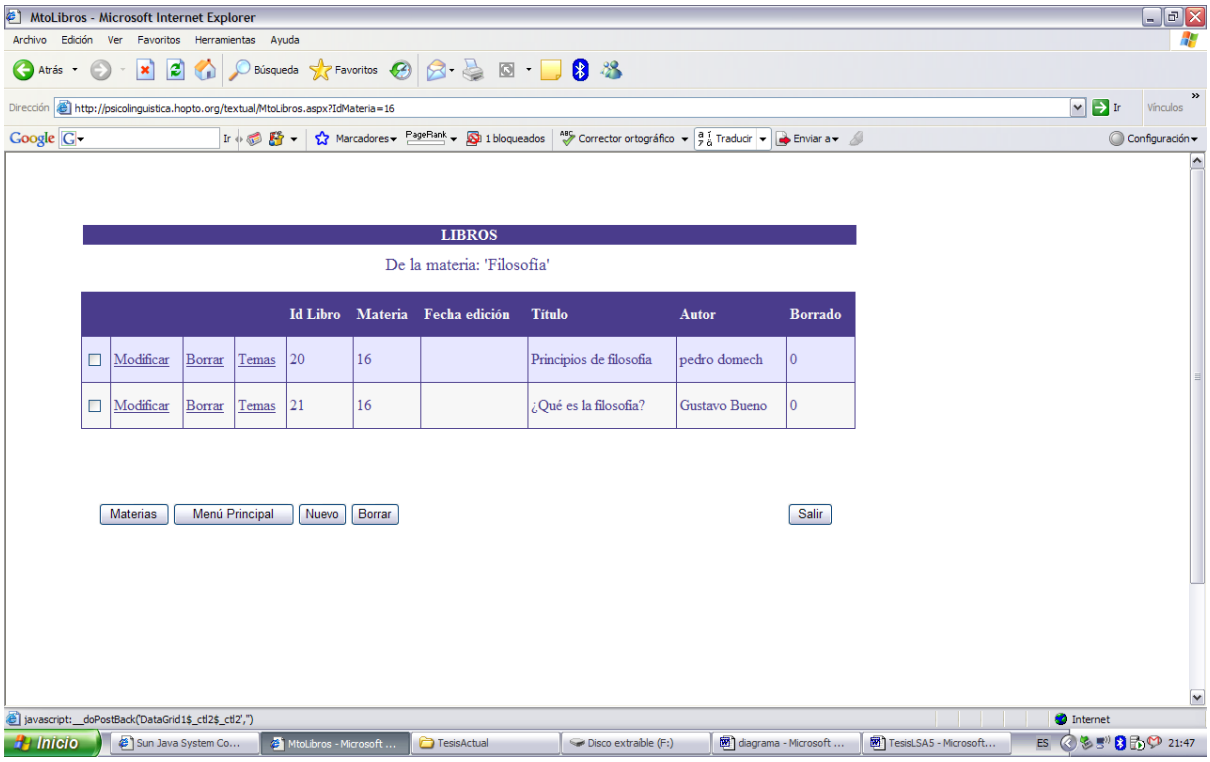


Figura 2.8. Pantalla de gestión de libros

c) Seleccionando el botón temas de la fila del libro “¿Qué es la filosofía?” obtenemos otra pantalla que contiene una tabla con los temas de ese libro (Figura 2.9). Cabe la posibilidad de insertar más temas. También existe la posibilidad de ir al libro al que pertenecen esos temas con el botón “Libros”

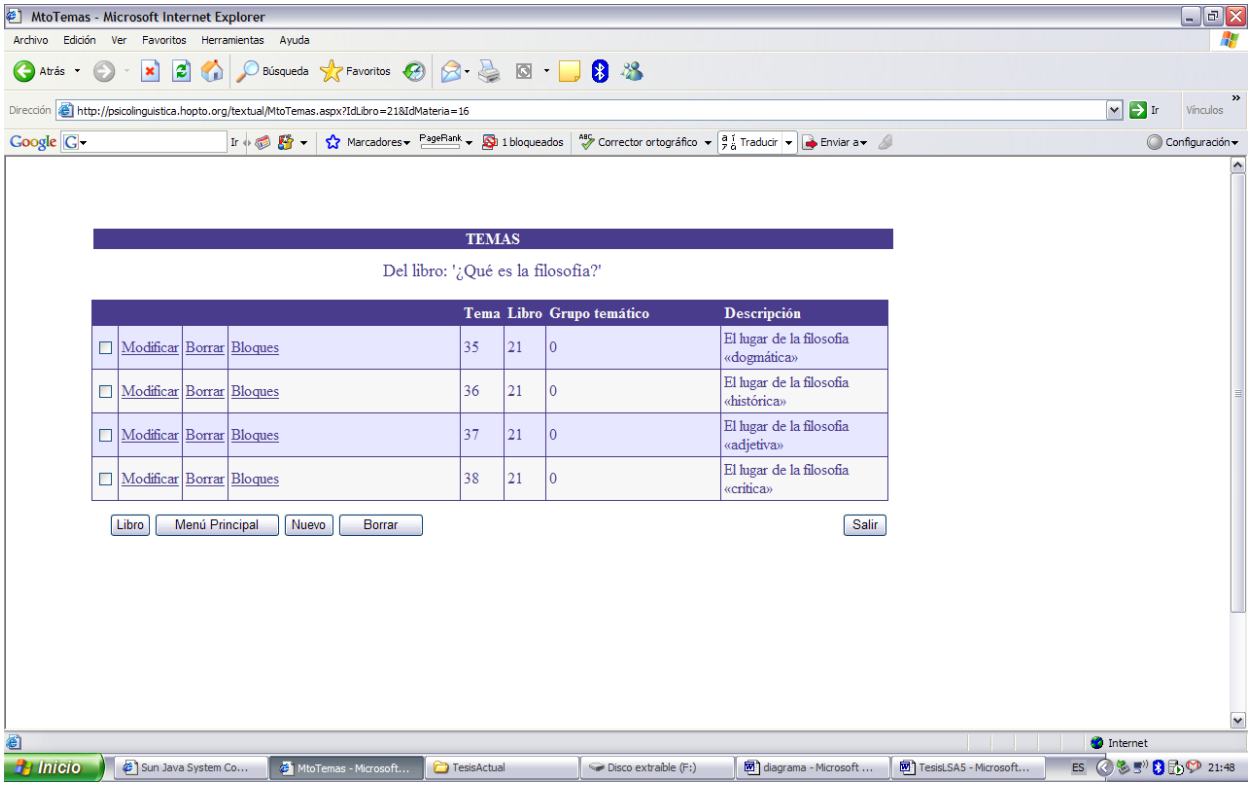


Figura 2.9. pantalla de ostión de temas

d) Seleccionando los bloques de la fila del tema “el lugar de la filosofía adjetiva” obtenemos los bloques que contiene ese tema con la posibilidad de introducir más (Figura 2.10). Existe también la posibilidad de navegar hacia el tema al que pertenecen estos bloques.

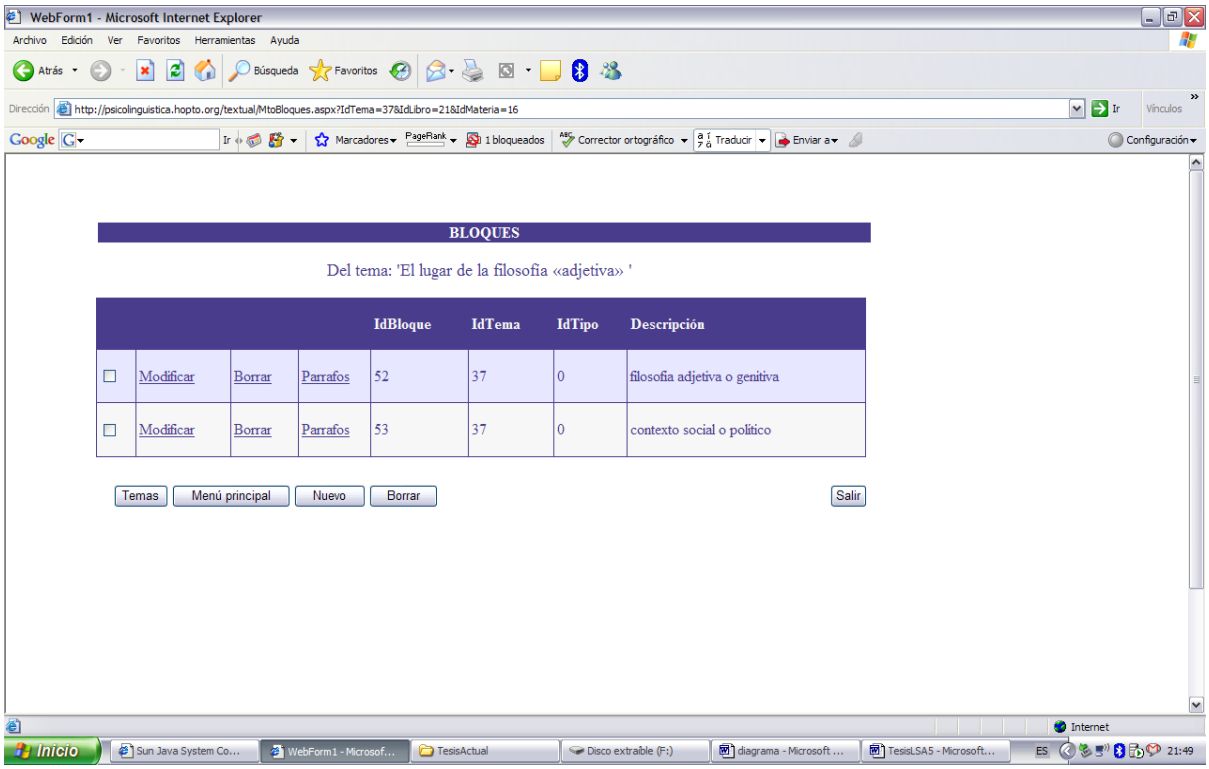


Figura 2.10. . Pantalla de gestión de bloques

e) Por último y quizás la parte más importante, seleccionando el botón párrafo de alguno de los bloques, por ejemplo “filosofía adjetiva o genuina”, se obtiene una pantalla con los párrafos (una versión pormenorizada de cada uno de ellos) que pertenecen a ese bloque (Figura 2.11). Aquí existe la posibilidad de introducir más párrafos que pertenezcan a ese bloque y de navegar hasta el nivel que se desee (bloque, tema, libro, materia)

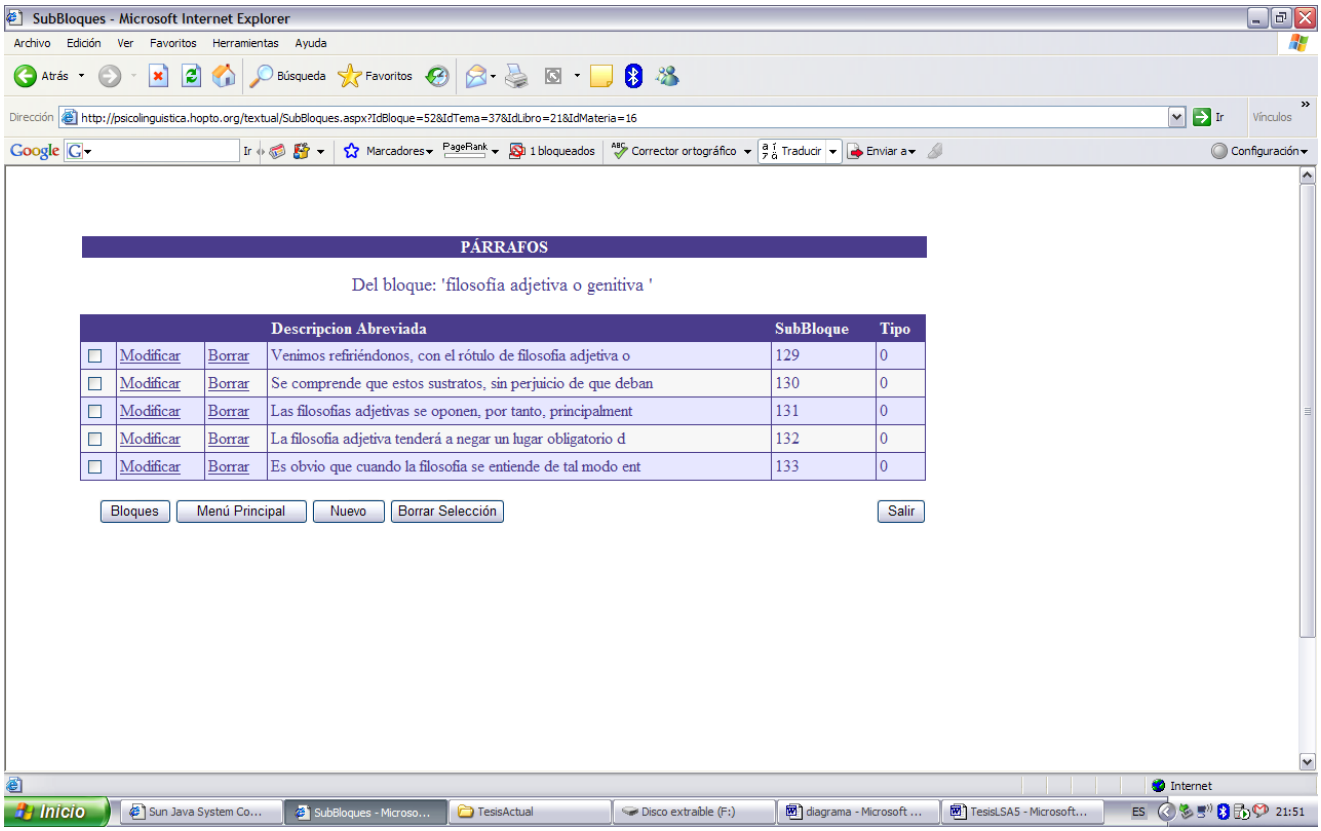


Figura 2.11. Pantalla de gestión de párrafos

2) *Pantallas de selección de las características de los párrafos*.

De esta forma, es posible introducir los párrafos de manera ordenada y estructurada. Con este tipo de representación, se podrá hacer selecciones en la base de datos que restrinjan los párrafos en base a sus características**.** A partir de estas pantallas se tendrá que contar con otras que se ocupen de ofrecer la posibilidad de seleccionar los tipos de párrafos que irán a formar parte del corpus. Un prototipo podría ser una ampliación de la siguiente pantalla en la que se visualizan los párrafos y su elección en base a las ramas de un árbol. El usuario podrá posicionarse en una rama y seleccionar los párrafos del nivel simbolizado en esa rama. A esta forma, se le puede acompañar con unos menús en los que se restrinjan esos mismos párrafos por cualquier característica que exista en la base.

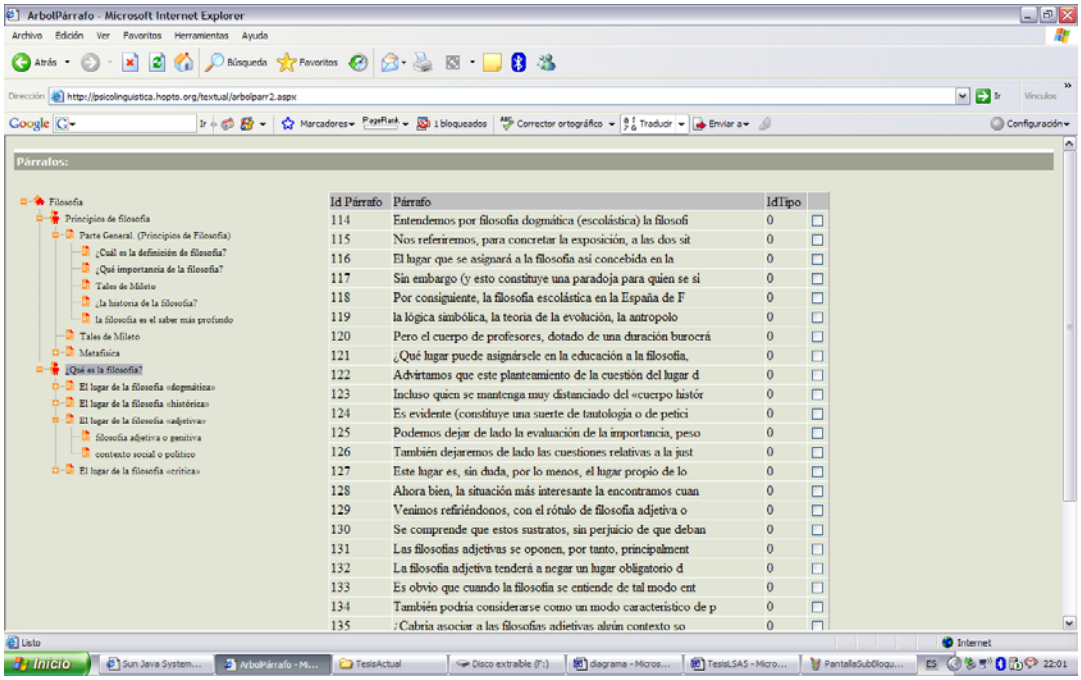


Figura 2.12. Todos los párrafos del libro "¿Qué es la filosofía?"

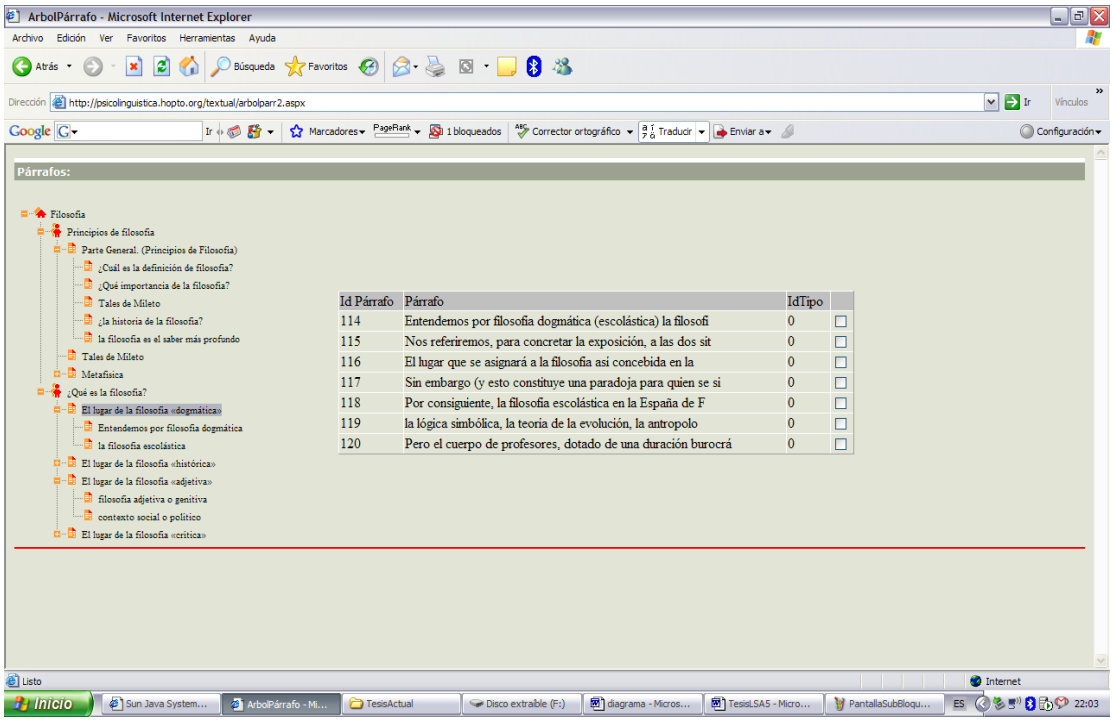


Figura 2.13. Todos los párrafos del libro "¿Qué es la filosofía?" pero del tema "El lugar de la filosofía dogmática".

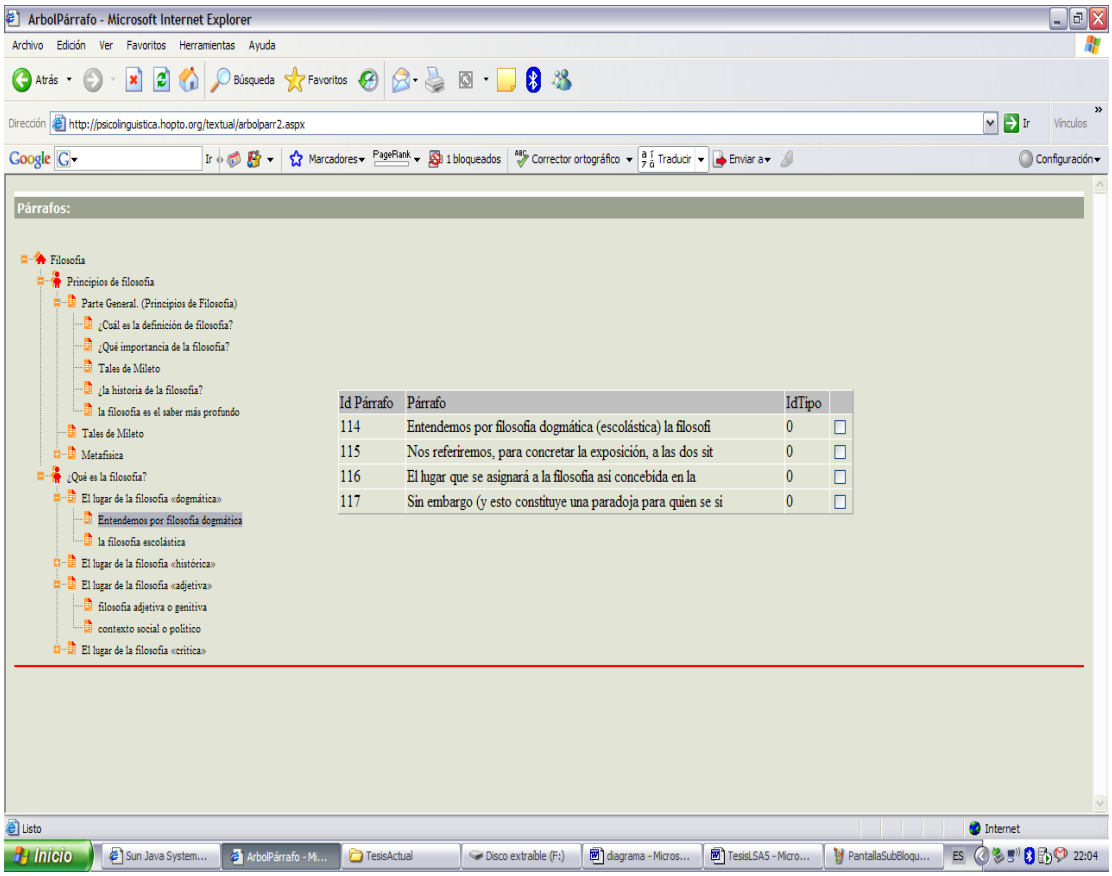


Figura 2.14. Todos los párrafos del libro "¿Qué es la filosofía?" pero del tema "El lugar de la filosofía dogmática" y del bloque "Entendemos por filosofía dogmática"

Además de estos ejemplos, es necesario añadir tantas constricciones cómo características consideremos importantes para la confección de los corpus. Esto puede ser llevado a cabo introduciendo algún objeto que permita seleccionar constricciones a las selecciones cómo por ejemplo, "Combos", "checkBox", etc. El objetivo final de este tipo de aplicaciones es generar un archivo de texto en el que se distribuyan con un formato legítimo, todos los párrafos que han sido seleccionados por algunas de sus características e incluso combinados bajo alguna proporción. Una vez obtenido este archivo de texto, este será la entrada para el análisis LSA.

2.8.6.- Vectores y medidas

2.8.6.1.- La representación del pseudodocumento mediante el centroide

Consultando la bibliografía científica sobre LSA se puede concluir que para elaborar pseudodocumentos son dos los métodos más empleados: el "*centroide*" y "*folding-in*" (o tres, si también incluimos el método de predicación de Kintsch,2001). Es importante recordar, que un pseudodocumento representado por un vector formado por el método "folding-in" tendrá tantas dimensiones cómo filas tenga la matriz **SkDk** (el pseudodocumento es un caso de documento) y que un pseudodocumento representado por un vector "centroide" tendrá tantas dimensiones cómo columnas tiene la matriz **TkSk** (el pseudodocumento es un caso de término promediado). Aparentemente y al ser reducidas ambas dimensiones a un número similar, parecería que ambas formas representan por igual al pseudodocumento, pero hay que tener en cuenta que en la matriz de concurrencias brutas, es común que el número de términos supere con creces al de documentos, por lo que la reducción de dimensiones no obedece a la misma magnitud, siempre partiendo del método "folding-in" de una representación más nutrida del vector. Esto incidirá en la manera de comportarse de ambos procedimientos. Aunque ambos son sensibles al tamaño de los pseudodocumentos que se quieren comparar, son los pseudodocumentos elaborados con la técnica del centroide los más afectados por dicho tamaño. Además, el centroide expresa la suma de los vectores que componen el pseudodocumento y puede darse el caso que unos vectores contrarresten a otros resultando un vector neutro y sin demasiada información sobre el sentido (Deerwester, Dumais y Harshman, 1990).

Aún así, el procedimiento del centroide puede aplicarse con cierta efectividad en pseudodocumentos de pocos términos, por ejemplo, en el caso de querer representar el fenómeno de que un término no sea restringido a una sola cadena de caracteres sino que puede ser varios términos que tengan una contingencia elevada como "fobia social" aunque esta forma puede generar algunas desvirtuaciones que son analizadas en otra parte de este mismo texto.

2.8.6.2.- Dependencias entre cosenos y tamaño del pseudodocumento

Cualquiera que sea la forma de introducir un texto en un espacio semántico existente, a saber: *centroide* y *folding-in*, puede resentirse de efectos que provienen del tamaño del texto. Hu, et al. (2003) advierten que el valor de similitud entre dos pseudodocumentos está directamente relacionado con el número de términos representados en dichos pseudodocumentos. Esta dependencia es un inconveniente para la interpretación de los índices de similitud resultantes*.* Varias tentativas de solución fueron sugeridas por Hu et al. (2006) quienes propusieron soluciones que van desde ponderar el coseno resultante en relación al tamaño de los pseudodocumentos, hasta tipificar dichos resultados tomando como referencia anteriores cálculos estadísticos de diversas muestras de tamaños. Aunque quizás en las matrices que se utilizan en grandes corpus generalistas, estos efectos no se produzcan de una manera tan exagerada, una de las primeras consecuencias de estos efectos de tamaño es que se podrían llegar a disipar las diferencias entre los textos a comparar, llegándose a una situación de equidistancia total*.* Dam y Kaufmann (en prensa), proponen unos ajustes sencillos a la hora de comparar documentos, que ayudan a maximizar sus diferencias. El primero y más sencillo es que se eliminen o simplemente no se tengan en cuenta, los términos que son compartidos por ambos documentos. El segundo y más costoso es eliminar manualmente de los documentos, las palabras que son juzgadas cómo irrelevantes en el ámbito de la temática de los corpus. Sus resultados muestran cómo esta manera se maximizan las diferencias de los documentos a comparar. En cualquier caso, toda eliminación artificial de términos en los documentos a comparar, puede producir infraestimaciones de la similitud.

2.8.7.- Vocablos compuestos de varios términos

Con la denominación “vocablos de varios términos” nos referimos a un cierto tipo de palabras o denominaciones que se componen de varios términos. Este tipo de vocablos son frecuentes en casi todos los textos y conviene

contemplar un forma que los tenga en cuenta. Ejemplo de esto son, “fobia social”, “casa rural”, etc. Si bien, este tipo de construcciones léxicas pueden ser consideradas casi cómo términos únicos también conviene hacer ciertas observaciones sobre la casuística de su tratamiento y de su modelado. Hay varias formas de tratar este tipo de entradas:

1) En algunos corpus cómo *LEXESP*, este tipo de vocablos vienen enlazados mediante un guión o cualquier otro carácter. De esta manera, ambos vocablos son considerados cómo un único término. Chu-Carroll y Carpenter (1999) emplearon este método para controlar los términos de los documentos extraídos de conversaciones telefónicas entre operadores y clientes en el diseño del “call routing” de un “callcenter” y para desambiguar entre textos que muestran intenciones y demandas parecidas . El análisis LSA de estos autores contiene términos formados por unigramas, bigramas y trigramas. Estos dos últimos se justifican por la alta frecuencia de aparición y por constituir servicios que se ofrecían en el “callcenter” cómo por ejemplo: car_loan, check_acount, check_acount_balance, etc. Aún así, esto podría contribuir a empobrecer el sentido toda vez que ambos componentes pueden salir sin el acompañamiento del otro. En el caso de que ambas palabras saliesen aisladas, estas palabras y la misma palabra inserta en un vocablo compuesto, no sería considerada como un mismo término y podría darse el caso de que las representaciones perdieran información de alguno de sus sentidos. En el caso de Chu-Carroll y Carpenter (1999) trataron de evitar esto hacen que palabras separadas y n-gramas aparezcan en los mismos documentos procesados como, por ejemplo, un documento puede contener: *check_acount_balance*, *check_acount*, *acount_balance*, *check*, *acount*, *balance*. Esto tiene también sus riesgos ya que, al fin de al cabo, se están introduciendo más términos al análisis. Chambers, Tetreault y Allen (2004), obtuvieron resultados por encima de lo normal si se empleaba cómo método la agrupación en N-gramas y una estimación del peso de esos mismos N-gramas dentro del cada documento. Estos autores emplearon LSA para reconocimiento del estado de ánimo de textos basándose en dos corpus del ámbito psiquiátrico.

2) Otra forma de representación es considerar este tipo de construcciones cómo si se trataran de un solo término. Lo más habitual es emplear la suma vectorial de los vectores que representan a esos términos (*centroide*), es decir, el promedio de sus sentidos y construir un nuevo término (documento formado por dos términos). Esto entraña algunos problemas debido principalmente a la longitud de vector de los términos que participan en dicha construcción. Puede darse el caso de que uno de los términos este representado por un vector con gran longitud mientras el otro apenas tenga. En ese caso, la representación de este término compuesto será prácticamente la misma que la del término con el vector mayor pasando el sentido del vector de menor longitud totalmente inadvertido (Kintsch, 2001). Otro problema que también tiene relación con este es el que proviene de la polisemia o de la homonimia de alguno de los términos de la estructura. Imaginemos que tenemos la entrada "perrito caliente" en un espacio semántico de tipo generalista. Supongamos también que "perrito" tiene una longitud de vector mucho mayor que "caliente". Puestos en la escena, imaginemos también qué "perrito" aparece representado mucho más frecuentemente en contextos de animales, caninos, etc. Dadas estas premisas, "perrito caliente" sería interpretado por LSA con un vector mucho más parejo en el espacio a los términos y documentos de estos animales que de comidas rápidas. Esto es debido a que perrito puede cobrar varios sentidos.

3) Una forma más exacta para ajustar el sentido de este tipo de construcciones es la propuesta por Kintsch (2001) y su algoritmo de predicación. Esta forma no puede ser empleada indiscriminadamente para todo tipo de vocablos compuestos sino que está restringida a estructuras que tienen forma de predicación. Esto obliga a localizar y detectar este tipo de estructuras lo que dificulta su proceso automático. Sin embargo, dados sus buenos resultados puede ser empleado por ejemplo en estructuras del tipo subcategorías cómo "fobia social" ya que se adaptan muy bien al tipo de estructuras para las que fue diseñado el algoritmo. Kintsch (2001) sugiere el ejemplo de "ese pájaro es un pelícano" (*bird is a pelican*) lo que concuerda con estructuras del tipo fobia social ("esa fobia es social"). Por tanto, puede ser

implementado este algoritmo para introducir estructuras de este tipo.

2.8.8.- Capacidad de los pc

Debido al gran número de términos y documentos que se pueden extraer de un corpus, la matriz bruta de concurrencias (incluso después de calculados los índices de importancia) tiene algunas características que la identifican. La primera de ellas es su gran tamaño, un corpus generalista puede generar matrices de unas dimensiones ingentes e imposible de procesar casi para cualquier pc. Por ejemplo, una matriz de 54119 x 45136 contiene 2, 442, 715,184 elementos. Si para cada elemento se reservan 8 bytes entonces se requerirán 19,541,721,472 bytes, es decir, 19 Gigabytes de memoria RAM, sólo para procesar dicha matriz. Además, para resolver el algoritmo SVD, se requerirá, a su vez, entre 4-5 veces el tamaño de la matriz original. El consumo de RAM rondará entonces los 100 Gb. Además, el consumo de RAM no aumenta de manera lineal según el tamaño, sino que lo hace de manera exponencial. Otra característica de estas matrices es que aproximadamente el 90% de los valores de las celdas son cero, de ahí que se les ponga el apelativo de “SPARSE MATRICES” (matrices huecas). Esta última característica será aprovechada para representar las matrices de una forma en que no se instancien las celdas en que el valor sea cero y permitiendo la realización de cálculos (cómo SVD) sobre ellas. Se trata del formato “*Sparse Matrix*”. Con esta nueva tecnología se podrán representar espacios mucho mayores en PCs comunes. Las matrices denominadas SPARSE se diferencian de las otras en que son representadas de manera distinta. En lugar de representar cada uno de los valores sean estos cero o un valor distinto a cero, en las matrices SPARSE, se representan mediante un vector, solamente los valores distintos a cero pero consignando también las coordenadas en las que están localizados, es decir, el lugar que ocupan en la matriz. A partir de este tipo de representación, los cálculos matriciales se hacen teniendo en cuenta esta nueva composición.

2.8.9.- Método de identificación de los valores críticos

Como consecuencia de las limitaciones que provienen de la escasez de memoria para operar con matrices de gran tamaño, Kontostathis, et al.(2005), propusieron un algoritmo destinado a localizar los valores de las matrices que no aportan nada a la creación del espacio semántico. El objetivo fue doble. Por un lado, convertir gran parte de los valores de la matriz a cero, cosa esta muy ventajosa pues se pueden emplear cálculos con “sparse matrices”. Por otro, eliminar los términos o documentos extremadamente irrelevantes. A partir de simulaciones y mediciones empíricas, estos autores idearon un método para rebajar hasta un 70% los valores tanto en la matriz de términos cómo de documentos sin que se resientan los resultados de manera significativa. Este método se aplica después de la descomposición del valor singular. En estudios preliminares, estos autores confirmaron algunos resultados interesantes como que si se eliminan los valores negativos de las matrices se resienten los resultados empíricos finales. A su vez, otra manipulación de las matrices mostraba que eliminando demasiados valores de la matriz ***Sk Dk*** (la matriz resultado de la matriz dimensionada de documentos por la diagonal dimensionada), los resultados se vuelven a resentir. En otras palabras, cualquier manipulación que se ejerza en las matrices con el objeto de reducirlas, tanto de términos cómo de documentos, ha de tener en cuenta estos dos efectos. Primero, que eliminando sólo valores negativos se consigue disminuir la efectividad de la representación semántica. Y segundo, cualquiera que sea la reducción, ha de tenerse en cuenta que la matriz de documentos es mucho más sensible a la manipulación y que, por tanto, ha de sufrir menos porcentaje de reducción que la de términos.

Partiendo de estas advertencias, Kontostathis et al. (2005) diseñaron un algoritmo que aún llevando a cabo la reducción mantuviese el mismo porcentaje de valores positivos que negativos y cuya eliminación de valores sea predominante en la matriz de términos $T_k S_k$. El algoritmo es el siguiente:

1) Se calcula tanto la matriz ***Tksk*** cómo ***Sk Dk***.

2) Se determina el umbral positivo por debajo del cual se elimina un porcentaje de valores positivos.
3) Se determina el umbral negativo por encima del cual se elimina un porcentaje de valores negativos.

En la aplicación generada por nuestro equipo, tanto el punto dos cómo el tres, es decir, los umbrales positivos y negativos dado un porcentaje a eliminar, fueron calculados utilizando la fórmula del centil. Respecto a los negativos, se tiene que tener en cuenta que lo que se desea es la eliminación de los valores negativos más próximos a 0, manteniendo los más extremos. Esto implicará que, a diferencia del umbral positivo, el negativo se calcule introduciendo el porcentaje inverso en la fórmula. Un buen criterio sería tomar el umbral del 70%. En este caso se introduciría en la fórmula del percentil (70 en el caso de los positivos y 30 en el caso de los negativos). Es preciso recordar que si, por ejemplo, obtuviésemos –0,21, serian eliminados todos los valores mayores de este valor y no los menores, cómo en el caso de los positivos (sería eliminado –0,01 y no –0,32 aunque se cumple qué –0,01 >–0,21 . (Con esto nos adelantamos al siguiente punto).

4) Para cada uno de los elementos de ***Tk***, se comprueba si es menor que el umbral positivo o si es mayor que el umbral negativo (es decir, si el valor cae entre ambos). Si esta comprobación es verdadera, se sustituye el valor de esa celda correspondiente en la matriz ***Tksk*** por 0.
5) Para cada uno de los valores de la matriz ***Sk Dk*** se comprueba si es menor que el umbral positivo o si es mayor que el umbral negativo (es decir,si el valor cae entre ambos). Si esta comprobación es verdadera, se sustituye el valor de esta misma matriz por 0.

El lector seguramente se habrá percatado de que en el paso 4 se comprueba en la matriz T_k y se sustituye en la matriz T_kS_k mientras que en el 5, es decir, sobre los documentos, se opera y sustituye sobre la misma matriz S_k D_k. Probablemente, esto será debido a que los autores quieren introducir el segundo particular descrito antes, a saber: que el porcentaje de

eliminación tiene que ejercerse más sobre la matriz de términos que sobre la de pseudodocumentos, por eso, aprovechándose que la multiplicación por Sk hace que $S_k D_k$ posea valores mayores que simplemente T_k, así comprueban los umbrales sobre T_k en el caso de los términos y sobre $S_k D_k$ en el caso de los documentos haciendo que el porcentaje de eliminación quede rebajado en los documentos.

En pasos sucesivos, se pueden eliminar los vectores cuyos valores sean sólo ceros pudiendo hacer desaparecer parte de los términos y documentos cuya representación es nimia. Además, la mayor parte de los valores quedan convertidos en cero. La ventaja de esto último viene dada por la posibilidad de utilizar algoritmos que empleen cálculos sobre matrices huecas (*sparse matrices*) lo que conlleva un ahorro de consumo de recursos tanto de memoria como de tiempo. Los resultados de estos autores en varios corpus de distintos tamaños muestran qué se puede reducir hasta un 70% de los valores sin incurrir en reducciones significativas de efectividad.

Capítulo 3

Modelos

3.- Modelos

3.1.- ¿Qué es un modelo?

Dentro de la simulación o de las teorías computacionales sobre el sistema cognitivo humano un modelo puede concebirse como una representación simplificada de ese sistema, construida para mejorar tanto la comprensión como nuestra habilidad para predecir y controlar el comportamiento. Un postulado filosófico básico que se aplica a los modelos de manera general dice que la realidad no "es en sí" esa realidad, sino que se formaliza de tal determinada manera que parece dar cuenta de los fenómenos que se describen en ella. De esta forma, los modelos tienen su caducidad en tanto en cuanto se afina en encontrar fenómenos o simulaciones que los contradicen. Por ello, se asume que la manera de producir conocimiento que se sustente sobre un modelo debe seguir las reglas del método hipotético-deductivo, esto es, se intuye un proceso, se propone una hipótesis (o modelo) y se realizan experimentos que puedan contrastar la hipótesis o el modelo con la "realidad". Una consecuencia frecuente de aplicar el método hipotético-deductivo es la dificultad de encontrar modelos que se ajusten con toda su plenitud a la realidad que representan. Más bien se suele encontrar lo contrario, modelos incompletos que, aunque no coinciden del todo con la realidad, resultan útiles en la medida que se convierten en un sistema "que funciona", aunque prescinden de algunas variables en la formalización. La buena praxis del modelado hace casi obligatorio situarse en niveles productivos de explicación (lenguaje descriptivo propio) sin que sea aconsejable mezclar explicaciones que se solapen en distintos niveles.

Uno de los análisis más lúcidos que se han realizado sobre los niveles de explicación a los que un modelo de simulación puede alcanzar es que realizó Marr (1985). Este autor estableció tres niveles de explicación diferentes. El primero y más básico lo denominó *nivel de explicación computacional o de cálculo,* con el que se describe las computaciones que un sistema lleva a cabo sin explicar detalladamente cómo lo hace. En este nivel de explicación consigna "el qué" se lleva a cabo, centrándose en la observación y el

experimento. El segundo nivel es el llamado *Algorítmico,* y es donde se representa la forma en que un sistema es capaz de llevar a cabo las computaciones descritas en el anterior nivel. Es en este nivel de explicación donde se formalizan los modelos y se da respuesta al "cómo". Existe también un tercer nivel llamado *Implementacional,* en el que se especifica dónde se llevan a cabo los algoritmos descritos en el anterior nivel de explicación. Este último nivel corresponde al "dónde" y "bajo qué" sustrato se explica el modelo. Sobre estos tres niveles, la ciencia cognitiva se concentra fundamentalmente en torno al nivel algorítmico y de cálculo, con el objeto de ofrecer explicaciones productivas y dejar en un segundo lugar el nivel implementacional (más dependiente de la neurociencia). Una razón de ello es que resulta siempre más fácil que se entienda el algoritmo con que se realiza una determinada función cognitiva si se comprende la naturaleza del problema que está resolviendo el sistema que si se examina el soporte físico donde se implementa (el cerebro). Existen, sin embargo algunos autores que, arguyendo la idoneidad de moverse en este segundo nivel, concentran sus críticas hacia modelos conexionistas y probabilísticos a los que LSA pertenece. Una de esas críticas ha girado en torno a que estos modelos representan micro-conexiones que emulan a sistemas incluidos en el nivel de implementación (neuronas) y, por tanto, son muy limitados para dar cierto tipo de explicaciones. Sin embargo, desde los modelos conexionistas se advierte que sus modelos dan cuenta de fenómenos emergentes que no podrían ser captados con el mero estudio de estructuras físicas y biológicas(Rumelhart y McClelland, 1992).

Precisamente, una característica de un buen modelo es sensibilidad para dar cuenta de las propiedades emergentes que surgen de las interacciones del fenómeno que se está estudiando. Un ejemplo de estas propiedades emergentes provienen del campo de la semántica cuando se asume que un texto no corresponde exactamente a la suma de sus términos, ni siquiera al conjunto de sus frases, puesto que la propia interacción de éstas hace que surjan propiedades emergentes que hacen que ese mismo texto cobre un nuevo sentido, haciendo explícito que el todo es más que la suma de sus partes. En esta dirección apunta el modelo que sustenta LSA cuando recurre al término "latente" (las micro-relaciones ofrecen fenómenos que no son

explicables sin ascender de nivel). Otro ejemplo de buena praxis en el diseño de modelos es hacer operativa toda la casuística posible, bien mediante explicaciones o bien contenida bajo otros procesos.

El modelo clásico en el que la psicología cognitiva se ha venido postulando desde hace décadas se basaba en que el procesamiento de información (donde también se incluye la información lingüística) se llevaba a cabo por medio de dos mecanismos fundamentales, a saber, las representaciones que están almacenadas en el sistema y los operadores en forma de reglas lógicas que manipulan dichas representaciones. De esta manera se puede modelar la realidad expresándola por medio de almacenes o retenes de información y operadores que transforman la información que por ellos pasan. Estos modelos son los que se conocen como *modelos de cajas y flechas* (estas últimas, indicando el flujo en el que la información a manipular fluye) y se basaban en la metáfora del ordenador, en la que la actividad mental que realiza dentro de un órgano (cerebro) se asemeja a los procesos o programas que se ejecutan en una máquina (ordenador). Así pues, es el programa (o la mente) el que manipula, trasforma e interpreta la información. Estos postulados cognitivos son equivalentes a los de la inteligencia artificial (IA) con la salvedad de que la psicología cognitiva ha de dar cuenta de procesos que acontecen en la mente a la hora de construir modelos mientras que la IA ha de confeccionar modelos que no tienen por qué estar basados en comportamientos humanos.

Uno de los problemas con que se enfrenta los modelos simbólicos de cajas y flechas es la dificultad que entraña la formalización por medio de almacenes y flujos de aspectos que quizás desborden la capacidad representativa de los modelos, a saber: la naturaleza interactiva de la representación del conocimiento y la simultaneidad de múltiples procesos. Con esto no queremos decir que no haya aproximaciones a este particular como, pongamos por caso, los modelos de doble ruta de acceso al léxico de Coltheart (1977), o los modelos de la neuropsicología cognitiva que gozan de cierta plausibilidad .Sólo referimos la dificultad añadida de poder transcribir al modelo (cajas y flechas) toda la casuística de los fenómenos lingüísticos. A esta

dificultad se refiere McClelland y Patterson(2002) cuando toma como pretexto la cuestión de la adquisición de la flexión verbal regular e irregular:

> *"Nosotros no defendemos que sería imposible construir un modelo basado en reglas de la formación de la flexión verbal que tenga todas las propiedades que alberga la evidencia […]. Si esos modelos usan reglas graduadas y salidas basadas en probabilidad. Contienen reglas que marchan gradualmente con la experiencia, incorporan constricciones semánticas y fonológicas e incorporan también información específica de la palabra, estos modelos serán empíricamente indistinguibles de los conexionistas […] (será) un nivel de análisis más alto, provisto de reglas que describen globalmente las regularidades capturadas por las conexiones de la red".*(p.471)

Lo que aquí se muestra con este problema es una simple cuestión de representación "sobre el papel" puesto que, por lo que se refiere este nivel de análisis, también podría describirse las operaciones en forma de reglas, aunque lo que subyazca, no fueran reglas en la más estricta acepción del término. Por tanto, sería legítimo movernos en el nivel algorítmico-lógico para describir fenómenos que no son lógicos (Marr, 1985) aunque esas reglas serían de alguna manera ilusorias, pues serían idealizaciones de la explotación de las regularidades. Este mismo debate sigue abierto en muchos ámbitos de la psicología, por ejemplo en torno a rol que tiene la existencia de la representación silábica vs las propiedades ortotácticas dentro de las unidades léxicas (Jared y Seidenberg, 1990).

Otro problema que conllevan los modelos simbólicos es lo que se ha venido a llamar la *modularidad de la mente* desde que Fodor (1983) propuso las características de un sistema modular. Parte de la psicología cognitiva, la neuropsicología cognitiva y algunas ramas de la lingüística, defienden la postura de que la mente se reparte en módulos funcionales y que cada uno de esos módulos poseen lo que Fodor llamó la propiedad de *encapsulamiento informativo*, es decir, que sólo pueden recibir información de su *especificidad de dominio* y no pueden nutrirse de los demás módulos para llevar a cabo su operación. También asumen que el flujo de información va de *arriba- abajo*, por lo que, procesos superiores no afectan ni varían el producto final de módulos inferiores. No todos los modelos asumen una modularidad ortodoxa como la que propuso Fodor pero sí que proponen un cierto encapsulamiento. La

formalización del encapsulamiento puede ser asumida con alguna garantía en estadios primarios de procesamiento como lo son quizás la percepción y la atención (bajo ciertas circunstancias), pero en procesos cognitivos más elaborados es más difícil sostener este postulado. Hay cierta evidencia de observaciones y experimentos clásicos muestran la filtración de información. Resaltamos los siguientes:

- **Percepción*:*** Las neuronas de V1 están determinadas por la orientación específica de los segmentos que componen la imagen y su localización (respuesta a los 80 ms.). Pero este tiempo y amplitud de respuesta está ampliamente influido por la estructura global de la imagen (véase a este respecto los experimentos de respuesta a contornos ilusorios de Lee y Nguyen, 2001).
- **Influencia del cruce de modalidades de entrada*:*** entrada auditiva con entrada visual. Una información ejerce de facilitador o anticipador (*priming*) para la otra.
- **Efecto de Stroop/facilitación** (*priming*)*:* La presentación de información de niveles superiores modula e incluso modifica, en ocasiones, las operaciones que se llevan a cabo en estratos inferiores. Presentaciones de información semántica modulan la forma de leer las palabras y provocan errores de acceso al léxico. Este efecto es conocido como *efecto de contexto* y es plenamente observable en la lectura o habla normal de la actividad cotidiana.
- **Efecto de superioridad de palabra** (Cattell, 1886; Johnston y McClelland, 1973): Resulta más fácil reconocer una letra cuando se encuentra integrada en una palabra que cuando está aislada o inserta en una serie aleatoria de letras.
- **Efecto de restauración fonémica*:*** Estímulos degradados en algún sonido pueden ser igualmente reconocidos (véase Warren (1970), en el que presentaba el ejemplo Legi#latura (# = un clic enmascarando la letra) y donde los sujetos no sólo reconocían la palabra sino que creían oír la letra).
- **Efecto "Tourple"** (Foster y Hector, 2002). En una tarea de categorización, la competición de los vecinos ortográficos no se pone en marcha a no ser que dichos vecinos pertenezcan a la categoría de la palabra. Esto muestra procesamiento arriba-abajo.

- **Asimetría en el *priming* de las palabras polisémicas** (Williams, 1992). Se consigna una asimetría en el *priming* en ambos sentidos de una lista de palabras polisémicas. Los datos contradicen la representación separada para cada sentido.

Por otro lado, la psicología cognitiva y, en general, todas las disciplinas insertas en lo que se ha venido a llamar ciencia cognitiva (psicología cognitiva, lingüística, biología, IA,…), cuentan con otra forma de modelizar la realidad de una manera un tanto distinta con respecto a la metodología, ya que contrariamente a la psicología clásica, que partía de la descripción del nivel computacional para dar una explicación en el nivel algorítmico, el conexionismo (Rumelhart y McClelland, 1992). Estos modelos, basándose en la dinámica neuronal real, al modo de las primeras teorías (McCulloch y Pitts, 1943; Hebb, 1949), han defendido frente a la modularidad de la mente un modelo distribuido y masivamente paralelo, que promueve transacciones de información en todas direcciones, de manera que unas constriñan a otras a la hora de mantener excitada o inhibida una determinada tentativa de hipótesis. Tampoco estos modelos están exentos de críticas por parte de algunos lingüistas, psicólogos y neuropsicólogos cognitivos. Una de las principales evidencias argüidas es la que emana de los casos de procesamiento anormal del lenguaje producido por un daño cerebral focalizado en algunas zona (afasias). En las afasias se pueden encontrar funciones lingüísticas preservadas y otras que no se llevan a cabo (es postulado axiomático que el nuevo comportamiento verbal no es fruto de nuevos aprendizajes post-accidente, sino de la falta de uno o varios módulos anteriormente implicados). Por ello, administrando estímulos previamente controlados en las variables que interesa comprobar, se pueden aislar módulos que están involucrados en determinadas operaciones invalidando, en cierta manera, la hipótesis de la distribución masiva e introduciendo cierto modularismo. En contra de esto y cómo evidencia a favor de los modelos conexionistas, se encuentra también la idea de que estos modelos simulan bien lo que se ha venido a llamar el efecto de “degradación elegante”, según la cual se describe como el rendimiento se deteriora gradualmente a medida que es destruida parte de la estructura neural. Es decir, pequeños daños en algunas zonas no alteran apenas la capacidad de funcionamiento. Es necesario un gran daño para que el daño funcional se

significativo. Si las representaciones no fueran distribuidas, la representación de la curva de deterioro describiría más bien funciones escalón y en realidad no suele ser así. Además y cómo indicio del paralelismo, los sistemas conexionistas son mucho menos sensibles a la información degradada introducida desde el exterior, como por ejemplo, estímulos lingüísticos incompletos.

De esta manera el modelo que subyace al LSA se puede alinear mejor en los modelos en paralelo y distribuido que en los modelos secuenciales. LSA también se puede adscribir mejor a los modelos interactivos frente a los modulares, de la misma manera que se ubica mejor en los modelos funcionales que en los localizacionistas. También se ajusta mejor a los modelos sub-simbólicos que a los simbólicos de cajas y flechas. Por otra parte, los modelos de memoria y conocimiento estático o dinámico representados por LSA son, en cierta manera, análogos a aquellos descritos por implementaciones de redes neurales. Por ello, una matriz LSA puede ser representada cómo una red neural artificial y que, a su vez, puede representarse de nuevo en forma de matriz (Rumelhart y McClelland, 1992; Landauer y Dumais, 1997; McClelland y Rogers, 2003). De esta forma, los resultados de los modelos producidos por LSA son parecidos a los fenómenos sobre los que da cuenta, a los resultados extraídos por sistemas de redes neurales que estudian la formación de redes de conocimiento.

3.2.- ¿Qué puede y qué no puede representar LSA?

Los modelos LSA representan la distribución estática del conocimiento de una persona expuesta a las entradas contenidas en un corpus concreto. En otras palabras, este corpus ejerce cómo un entrenamiento previo para configurar la red que representa el conocimiento estático, es decir, el conocimiento que tenemos de las relaciones que tienen unos términos con otros. Es de suma importancia entender que para que LSA extraiga juicios análogos a los humanos, haya tenido que estar previamente expuesto a una muestra lo suficientemente representativa del lenguaje humano (Quesada ,2006). Esta aclaración hace alusión a que LSA representa, fundamentalmente, conocimiento estático. Pero, ¿qué significa realmente la simulación con LSA? Burgess (2000), quién responde a las críticas que sobre los modelos LSA y HAL realizaron Glendberg y Robertson (2000), sentó las bases en torno a lo que significan los modelos que emanan de LSA y que reproducimos, en nuestro caso, ordenando y ampliando. Enumeramos las siguientes:

a) LSA simulará el conocimiento en tanto en cuanto el corpus al cual se somete sea representativo del conocimiento que se quiere simular.

b) LSA puede considerarse una representación estática del conocimiento. Por ello, una matriz LSA representa un conocimiento cristalizado de un ámbito de conocimiento.

c) De lo anterior se deduce que LSA es únicamente un modelo de representación y no un modelo de procesamiento del lenguaje. Resulta del todo imposible pretender que con los vectores LSA se extraigan modelos de procesamiento. LSA representa la memoria estática o a largo plazo que posee una persona.

d) Las personas tienen un sistema de procesamiento cognitivo activo y creativo que facilita el uso de los contenidos de la memoria.

e) LSA funciona como base para proponer modelos de procesamiento, llevando a cabo algoritmos que simulan procesos lingüísticos. Que el LSA funciones como base quiere decir, a grandes trazos, que el modelo de conocimiento proporcionado por LSA (memoria estática o a largo plazo), esta sesgado por un contexto, bien sea éste semántico, sintáctico o de cualquier otra naturaleza, de tal manera que simule los procesos observados en el procesamiento de sujetos reales. Tal es el caso, por ejemplo, del algoritmo de predicación (Kintsch, 2001) o de la intervención del conocimiento previo (Denhiere, et al, 2007). Estos algoritmos aplican como base la representación estática (memoria estática a largo plazo) que proporciona LSA para crear un algoritmo que dé cuenta del

procesamiento de predicaciones o del influjo del conocimiento previo en la intervención de la memoria de trabajo.

Todo ello nos lleva a pensar que LSA proporciona una representación de los términos, a imagen del contexto y que viene dado por el corpus. A partir de aquí, las medidas y algoritmos que se lleven a cabo serán más o menos complejas, dependiendo de la complejidad del proceso que se quiera emular. Éstas pueden oscilar desde medidas simples de similitud (cosenos y distancias) hasta algoritmos que introducen sesgos debidos a contextos sintácticos, semánticos, como en el caso de Kintsch (2001) o el de Juvina y Oostendorp (2005a, 2005b).

3.3.- Fenómenos simulados por LSA

3.3.1.- Pobreza de estímulo

3.3.1.1. Concepto

Resulta muy ilustrativo el título del artículo que escribieron Landauer y Dumais(1997) "La solución al problema platónico", para explicar la técnica LSA y los modelos que de ella se pueden extraer. Según estos autores, la técnica es una tentativa de solución a un problema planteado ya por Platón hace siglos, sobre como las personas tenemos más conocimiento del que puede extraerse de la información a la que hemos sido expuestos, o ¿cómo el esclavo puede llegar a hacer razonamiento sin haber recibido educación alguna sobre temas relacionados o haya sido expuesto a problemas que exijan razonamiento como forma de buscar una solución? Platón, acorde con su manera de pensar, soluciona la cuestión proponiendo que las personas vienen dotadas de conocimiento innato. La solución de Landauer y Dumais es mucho más profana. La solución sobre como las personas poseemos más conocimientos de los que se podrían extraer de los datos o de la experiencia a la que estamos expuestos se debe a que en nuestra arquitectura funcional, más que estar dotados del conocimiento innato, poseemos mecanismos que nos hacen inducir el conocimiento a partir de su entorno. Este conocimiento es extraído indirectamente, de las coocurrencias locales de los datos y, en el caso de LSA, a

partir de micro-relaciones entre términos pertenecientes a un corpus increíblemente extenso de lenguaje. Los niños suelen aprender por instrucción directa mucha menos información que la que puedan llegar a extraer por estos mecanismos de inducción indirecta (Landauer (2002). Aunque la teoría innatista Chomskiana sea actualmente discutible (Chomsky, 1991), descansa en la idea de que algún mecanismo existe en la mente de los niños que pueda emplear la información limitada y finita de la que disponen y transformarse en expertos usuarios del lenguaje. Este fenómeno es conocido también como "pobreza del estímulo". Sin embargo, no hay suficiente información en las contingencias de los estímulos del medio para entender y sacar conclusiones sobre cómo se adquiere esta información adicional, sobre como dos palabras, pongamos por caso, que no aparecen juntas en un mismo entorno, puedan, sin embargo, llegar a inducir y compartir la posesión de un valor funcional similar. En el caso de las metáforas, por ejemplo, no aparecen nunca en el mismo lugar el contenido referido y, sin embargo, en la mayoría de los casos, somos capaces de inducir su significado, si que caigamos en las redes de su significado literal. En el símil *mi abogado es un tiburón* y la expresión *mi abogado es joven*, conllevan procesos de acceso al significado diferentes. En la primera frase es necesario ser sensible al contexto en donde se produce, pues "tiburón" significaría una cosa distinta si lo analizáramos aisladamente (Kintsch y Bowles, 2002). Tiburón, en este primer caso tiene otros patrones de importancia que no coinciden con que sea pez, sino con que sea agresivo, codicioso, etc. Estos patrones de reconocimiento son inducidos por el mismo contexto de aparición. Así pues, es el sistema el que actúa sobre los dominios de conocimiento los cuales están definidos por un vasto conjunto de pequeñas interrelaciones. De estas pequeñas interrelaciones es de donde se captura la existencia de conglomerados semánticos. A juzgar por los datos empíricos, Landauer y Dumais (1997) consideraron que LSA es un buen modelo que refleja este fenómeno pues puede ayudar a modelar cómo se capturan las relaciones que mantienen los términos en distintos órdenes.

3.3.1.2. Captación de las relaciones de distintos órdenes.

En cualquier corpus lingüístico podemos encontrar que los términos mantienen relaciones de órdenes distintos. Los términos que procesamos por medio de LSA no escapan a esta afirmación.

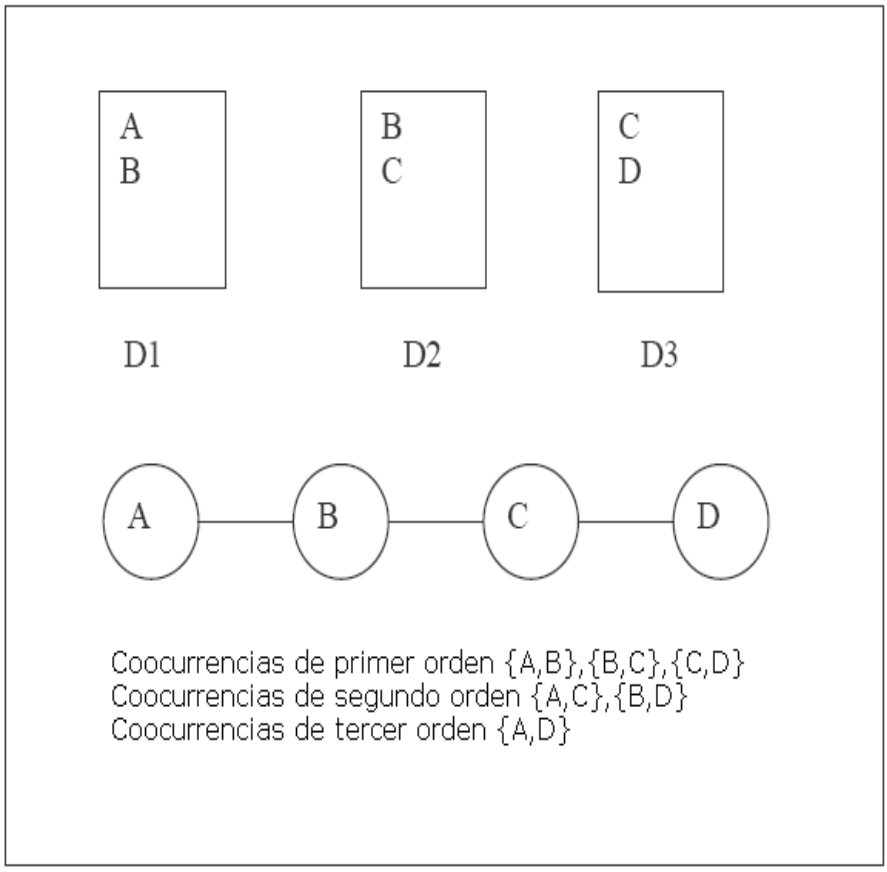


Figura 3.1.- Representación gráfica de ocurrencias de orden 1, orden 2 y orden 3.

En ellos, podemos encontrar desde relaciones de primer orden que son las que mantienen las palabras que ocurren en un mismo documento (son las que podemos extraer de la matriz), de segundo orden, que no ocurren juntas pero si lo hacen con un términos común, de tercero y hasta de orden *n*? Una de las ventajas que se le presumen a LSA como modelo es la capacidad para captar las relaciones de más de un orden que mantienen los términos entre sí pero ¿en qué medida este tipo de relaciones de más de primer orden participan

en la representación de las entidades en los modelos LSAQue el conocimiento de las palabras y párrafos no se forja únicamente por la concurrencia de los términos fue examinado por Landauer (2002). En este trabajo indagó sobre uno de los aspectos más interesantes de este tipo de técnicas que, en cierta manera, emulan la adquisición del conocimiento y es que asumiendo simplemente las palabras que ocurren en el mismo pasaje, es imposible dar cuenta de las inferencias que tiene que realizar un modelo que emule la adquisición de significado. Según los autores, el lenguaje puede ser concebido como un conjunto de sistemas sobre los cuales requiere realizar inferencias para extraer el significado. Estas inferencias se hacen tomando incluso unidades que no han concurrido nunca. Landauer hace gráfica sus explicaciones con una seria de sistemas de ecuaciones.

ecks + wye + aye = foo
ecks + wye + bie = foo

Cómo probablemente habrá advertido al lector, “aye” y “bie” nunca ocurren en el mismo pasaje, pero se puede establecer entre ellos una relación identitaria que curiosamente no se podría mantener de la misma forma con “ecks” y “wye”. Landauer sigue con la argumentación y propone añadir más complejidad para adentrarse mejor en lo expuesto:

ecks + wye + aye = foo
ecks + wye + bie = foo
ecks + wye + cee = bar
ecks + wye + dee = bar

De esta manera sabemos que al igual que “aye” y “bie”, “cee” y “dee” pueden ser considerados sinónimos. Si, además, añadiésemos un tercer sistema:

aye + cee = oof
bie + dee = rab

Al ser “aye” y “bie” sinónimos al igual que “cee” y “dee”, podemos concluir que también lo son “oof” y “rab” incluso no teniendo estas últimas palabras ningún término en común. Según Landauer (2002), el lenguaje puede parangonarse con un conjunto muy amplio de dichos sistemas, lo cual requiere de mecanismos exigentes en cuanto desarrollo de inferencias. Una vez definida la estructura potencial sobre dónde pueden extraerse dichas inferencias se cuestiona lo siguiente: ¿qué proporción de conocimiento adquirido proviene de las concurrencias indirectas de los términos y no de la simple contigüidad de estos mismos términos? O, en otras palabras, ¿qué parte de la adquisición del conocimiento proviene de la resolución de dichos sistemas?

Landauer y Dumais (1997) trataron de simular el proceso de adquisición del lenguaje con LSA. Para ello, trataron de medir de manera objetiva la parte de conocimiento sobre las palabras que aparecen en textos dentro de los cuales aparecen esas mismas palabras y que parte se adquiere en textos donde no aparecen esas mismas palabras. Dicho de otra manera, se trataba de calcular la ganancia de conocimiento sobre una determinada palabra, bien que se obtiene de los textos en los que esa palabra aparece de manera explícita y en otros textos donde esa misma palabra no aparece. Para contrastar ambas condiciones, estos autores tomaron como corpus el proporcionado por una enciclopedia de conocimiento general y como medida del conocimiento adquirido sobre una palabra, tomaron como referencia la prueba para estudiantes extranjeros TOEFL (*English as a Foreign Language*). El razonamiento es simple. Si se ha adquirido un conocimiento aceptable de una palabra en la fase del corpus, se espera entonces obtener unas buenas puntuaciones en el TOEFL (recuérdese que para confrontar el conocimiento del espacio semántico proporcionado por LSA y el TOEFL, el procedimiento será que se elegirá de las alternativas la que mayos coseno tenga con la pregunta, (aunque estos autores lo modulan para refinar la medida). Como variables independientes se controlaron: a), los textos que contienen las palabras del TOEFL y que están siendo evaluadas y textos que no las contienen; b), el tamaño del corpus (aunque se cumple que al reducir textos de un corpus, nunca se eliminarán textos en los que salen palabras del TOEFL); y c), la frecuencia de las palabras TOEFL en los corpus mediante la sustitución de

algunas de estas palabras por palabras sin sentido. De esta manera se respeta el contexto original de las palabras.

Los resultados obtenidos fueron reveladores. El grado de eficacia que se obtuvo en la prueba TOEFL estaba en función tanto de los textos que contienen las palabras referencia (TOEFL) cómo de los textos que no las contienen, existiendo una interacción significativa entre ambas condiciones. Esta interacción mostraba que el hecho de introducir textos adicionales en los que la palabra referencia no aparecía resultaba más beneficioso para palabras con mayor frecuencia en el corpus. En otras palabras, cuanta más frecuencia poseía una palabra, más beneficio obtenía en los textos en los que ella misma no aparecía. Además, Landauer y Dumais (1997) realizaron una estimación sobre la ganancia de vocabulario debida al efecto de contacto directo con la palabra, calculando que es de 0,0007 por cada palabra que aparecía en el texto. Pero el cálculo ascendía hasta 0,15 de ganancia por texto leído cuando se trataba del efecto indirecto de las demás palabras. Con estos datos en la mano, no resulta descabellado afirmar que tanto las concurrencias de orden mayor que uno cómo la ausencia de los términos en textos de distinta temática, resultan igualmente relevantes para poder explicar las inferencias realizadas para llevar a cabo un índice tan alto de aprendizaje indirecto. Esto es posible gracias a la arquitectura de dichos modelos que se basa en pequeñas conexiones locales entre palabras y que hacen posible un conocimiento por inducción. Tanto es así que, desde la técnica LSA, se han propiciado investigaciones en las que se hipotetiza una disfunción en la habilidad de captar este tipo de aprendizaje indirecto para algunos trastornos cómo el autismo, en la creencia de que este tipo de cuadros poseen el rasgo de una excesiva dependencia a las relaciones de orden 1 (ver a este respecto Skoyles, 1999).

De manera más específica, Mill y Kontostathis (2004) encontraron que la similitud de los términos entre sí no estaba en función de la simple concurrencia de orden 1 y que, entre los pares de términos que tienen concurrencia de orden 1 nula (no ocurren nunca juntas), la variabilidad de similitudes de una a otra era muy amplia. Para aquellas palabras que nunca

concurren, es precisamente la concurrencia de orden 2 la que se erige como el factor decisivo que explica sus índices de similitud. Los resultados de este tipo de palabras mostraban un crecimiento exponencial en su similitud con otra palabra conforme crece su coocurrencia de orden 2. Este crecimiento es más acusado en corpus dimensionados con menos factores debido, quizás, a que cuantas más dimensiones se empleen más se parecerá a la matriz de concurrencias brutas y, por tanto, a la matriz de coocurrencias de orden 1. También Lemaire y Denhière (2006) obtuvieron resultados similares utilizando, en este caso, el método de añadir párrafos y analizar a que es debida la ganancia en la similitud de 28 pares de términos controlados. Los resultados confirmaron que tanto el orden 2 como el 3 aportaban ganancias al cómputo de la similitud total.

3.3.1.3.- Medida de las relaciones de distintos órdenes en los textos

Una forma sencilla de medir los distintos órdenes de un determinado corpus es emplear los procedimientos de Kontostathis et al. (Kontostathis, 2004; Kontostathis y Pottenger (2006). En este procedimiento se calcula la matriz de coocurrencias de orden 1 multiplicando la matriz bruta de coocurrencias por su traspuesta. Esta operación dará como resultado una matriz términos-términos en la que se representarán las veces que dos términos ocurren en un mismo documento. Sobre esta matriz se calcularán los órdenes superiores, pero antes se pondrán los valores en binario y se sustituirá la diagonal por cero. Una vez realizado este paso, se multiplicará la matriz por sí misma, operación de la que resultará la matriz de coocurrencia de orden 2. El significado de sendas matrices es algo distinto. En cada celda de la primera matriz (Tabla 3.1) se representará si esas dos palabras tienen co-apariciones comunes. Es decir, si aparecen juntas en algún documento. Sumando las celdas de una determinada fila, se extrae el número de palabras que establecen relaciones de orden 1 con el término que representa a la fila. Véase debajo la matriz de orden 1 extraída con nuestro propio sistema experimentando con el ejemplo anterior (Landauer, 2002). Supuestos dos documentos:

D1: ecks + wye + aye

D2: ecks + wye + bie

	ecks	wye	aye	bie
ecks	0	1	1	1
wye	1	0	1	1
aye	1	1	0	0
bie	1	1	0	0

Tabla 3.1: Matriz de orden 1.

	ecks	wye	aye	bie
ecks	3	2	1	1
wye	2	3	1	1
aye	1	1	2	2
bie	1	1	2	2

Tabla 3.2. Matriz de orden 2.

Sin embargo, la matriz de orden 2, aún siendo también una matriz término-término, tiene un significado algo distinto. Cada celda representa el número de palabras que actúa de puente entre los términos que representan las filas y las columnas. Es decir, si un par de palabras no aparecen juntas en un documento cómo es el caso de “aye” y “bie” pero existen palabras que aparecerán con ambas “ecks” y “wye”, se dirá que entre “aye” y “bie” hay dos palabras que actúan de puente para que “aye” y “bie” establezcan una relación e segundo orden. Esto es lo que expresa cada valor en la celda de esta segunda matriz (Tabla 3.2). El número de términos que actúan de puente entre cada par de términos.

Cómo manera de indicar mediante un índice la cantidad de relaciones de orden 1 y orden 2 existentes y evitando que dichos índices sean dependientes del tamaño de los corpus, proponemos unas simples fórmulas para el cálculo que han sido implementadas en nuestros sistemas:

El índice de orden 1 es calculado de la siguiente manera:

$$I_1 = \frac{(\sum\sum xij\ /n) \times 100}{n}$$

Dónde:

(∑∑xij /n) es el promedio de términos que tienen relación de orden 1 con cada término. Es el sumatorio de las filas sumadas entre el número de filas.

Se aplica a la anterior fórmula una conversión a porcentaje sobre el total de términos en el corpus. El índice de orden 2 es calculado de la siguiente manera:

$$I_2 = \frac{(\sum\sum xij\ / n^2) \times 100}{n}$$

Dónde:

(∑∑xij / n^2) es el promedio de palabras que actúan cómo puentes entre cada par de palabras, es decir, la media de todas las celdas.

Se aplica a la anterior fórmula una conversión a porcentaje sobre el total de términos en el corpus. Es decir, cuanto representa el promedio de palabras puente por cada par, sobre el total de términos. Según los cálculos, el ejemplo quedaría de la siguiente forma:

$$\text{Iorden1} = (2{,}5 * 100) / 4 = 250 / 4 = 62{,}5$$

Lo que significa que cada palabra tiene relación de orden 1 con un promedio del 62,5 % de los términos del corpus.

$$\text{Iorden2} = (1{,}62 * 100) / 4 = 162 / 4 = 40{,}5$$

Lo que significa que cada par de palabras tiene un promedio de un 40,5

% de términos que actúan como puente en las relaciones de orden 2.

3.3.2.- Sinonimia y antonimia.

Los espacios generalistas LSA han sido contrastados con los juicios de sinónimos de algunos test cómo el TESL (*Educational Testing Service of English as a Second Language*) o sobre el TOEFL (Landauer and Dumais, 1997; Turney, Peter (2001). Los resultados fueron muy satisfactorios en dichas pruebas y los espacios no sólo simulaban el porcentaje de aciertos (cómo un extranjero más), sino en el caso del trabajo de Landauer and Dumais (1997) simularon también el patrón de errores. Habida cuenta de estos resultados se confirma la idea de que LSA simula de manera eficiente la captación de términos semánticamente próximos. Respecto a estudios de LSA sobre antónimos, como términos frecuentemente relacionados entre sí, el análisis LSA ofrece cosenos muy similares cuando compara antónimos cómo sinónimos. Esta peculiaridad puede considerarse una virtud en el sentido de que refleja plausibilidad psicológica de la comprensión real de la antonimia, aunque produce problemas a la hora de diferenciar el significado de frases de manera eficiente como, por ejemplo, diferenciar entre *un gato negro es mala suerte* y *un gato negro es buena suerte* (Landauer, 2002). Ambas frases obtienen un coseno de .96. Respecto a la realidad psicológica se concebiría la antonimia cómo un caso especial de sinonimia donde “buena” y “mala” pertenecerían al mismo campo conceptual. Por otra parte, si lo que se pretende es que LSA simule no sólo la representación de conocimiento sino también la comprensión de oraciones, habría que formalizar entonces algún tipo de mecanismo adicional, ya que LSA por sí solo no podría llevar a cabo dicha simulación. Recordemos que LSA no representa procesos de comprensión del lenguaje sino como están representados los conceptos que esos mismos procesos activan (Burgess, 2000).

3.3.3.- Polisemia y homonimia

Una palabra puede considerarse polisémica cuando podemos expresar con ella varios significados. Deerwester, Dumais y Harshman (1990) analizaron este fenómeno dentro del marco de LSA. Estos autores expusieron que aunque el fenómeno de la sinonimia esta fielmente representado por las simulaciones LSA, este no es el caso de la polisemia. Recordemos que un término, aunque con más de un significado, estaría representado por un único punto en el espacio. Este punto obedecería a unas coordenadas. Al ser varios sus significados, estos están representados cómo un promedio de sus significados ponderados por la frecuencia de los contextos en los que aparece. Si ninguno de sus significados es cómo ese promedio puede crear entonces un sesgo en la representación espacial, resultando una entidad que no se ajusta a ningún referente. Esto recuerda a algunas de las críticas que se vertieron en modelos basados en prototipos (Rosch y Mervis, 1975) y que se centraban en la existencia de un ejemplar prototípico suponía la suma de los rasgos típicos de los ejemplares de la categoría. La crítica aludía a que siendo el prototipo un ensamblaje de rasgos de una categoría y teniendo en cuenta la variabilidad de los elementos típicos, se daba la paradoja de que, a menudo, el prototipo resultante, con el que se establecen los juicios de similitud, resultaba ser un miembro muy atípico.

Aún así y considerando como una solución parcial, dichas simulaciones ofrecen buenos resultados en lo que respecta a la desambiguación del significado con la ayuda contextual de los demás términos que lo acompañan. Esto es, si cabe, más acertado en modelos más sofisticados de análisis que se basan en LSA, cómo el algoritmo de predicación propuesto por Kintsch (2001) a los que dedicaremos algunos apartados posteriores. Mediante este algoritmo, los vectores-término son sesgados convenientemente por el contexto para que se ajuste sólo el sentido concerniente al contexto que buscamos. Es decir, un vector término es sesgado para potenciar sólo los valores a los que se refiere el contexto. Sobre la polisemia dedicaremos también el siguiente apartado en dónde se indaga además sobre la incidencia que tiene este fenómeno en los modelos que sugieren una única representación de los términos. Es decir,

modelos que, al igual que en los modelos LSA, se guarda un solo vector que representa todos los sentidos que un términos posee. La homonimia y la polisemia, aunque con ciertas diferencias de origen, comparten la misma definición. Decimos que dos palabras son homónimas si su significante es el mismo, es decir, están compuestas por los mismos fonemas o grafemas, o su realización fonética o grafémica coincide. La polisemia se distingue de la homonimia en que esta última se trata de una relación entre los dos planos del signo lingüístico: los diferentes significados de una palabra tienen, o han tenido, un origen común. En otras palabras, la polisemia tiene relaciones entre significado y significante, mientras que la homonimia sólo lo tiene de significante. Estas diferencias no pasan desapercibidas para la lingüística atiende a rasgos etimológicos, pero, creemos, poco pueden afectar al procesamiento real del lenguaje. Por tanto, aun considerando que puede tener su utilidad taxonómica, puede que estas diferencias no sean tan procedentes para el estudio de los aspectos psicolingüísticos pues ambos fenómenos pueden operar funcionalmente de manera similar y bajo los mismos mecanismos. Esto es precisamente lo que proponemos cómo punto de partida en el siguiente apartado en el que analizamos el fenómeno de la homonimia y la polisemia en la asunción de una única representación de todos los sentidos de un término.

3.3.4.-La representación única del término

Realizada la anotación sobre la similitud funcional de homonimia y polisemia, no resulta del todo descabellado señalar aquí que el análisis y modelado de este tipo de relación ha ocupado un lugar privilegiado en lo que respecta al procesamiento del lenguaje y al modo de representarse en la mente. Desde los modelos de reconocimiento de palabra basados en el procesamiento distribuido y paralelo (Rumelhart y McCleland, 1992; Seidengberg y McCleland, 1989), se han venido proponiendo modelos basados en algoritmos, en el que el aprendizaje estaba representado por la variación de los pesos en las unidades que forman una red y, también, en la medida en que disminuye el error en cada ciclo en que sometemos al sistema a un conjunto de entradas. Esto quiere decir que la representación y recuperación de los términos (ortográficamente o fonológicamente entendidos) forman un mismo proceso y que una palabra será representada de manera única aunque posea varios significados. Será el contexto de recuperación el que constriña el significado preponderante que en ese momento es adoptado.

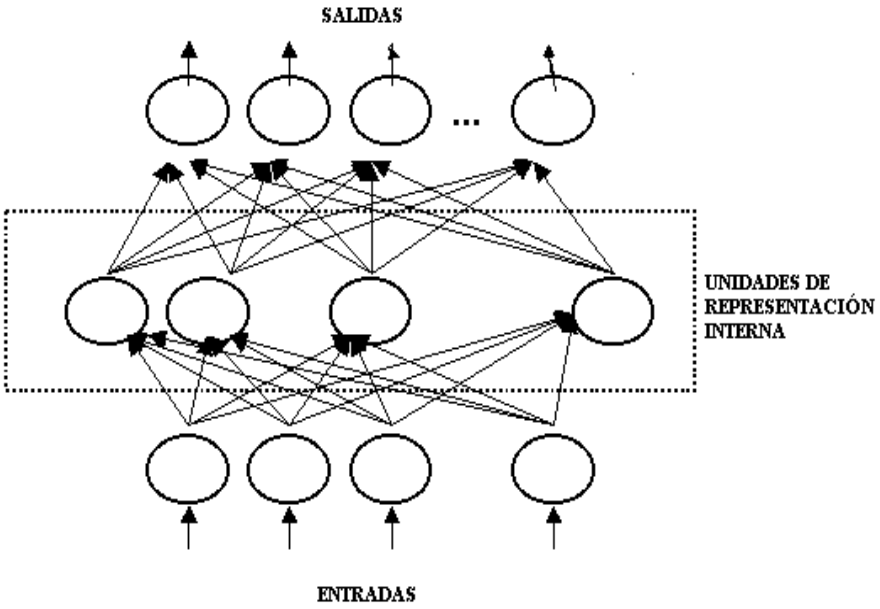


Figura 3.2.- Representación de una red multicapa en el que entradas y salidas son mediatizada por una capa oculta.

Estos sistemas suelen estar compuestos por una red que posee capas de entrada, capas ocultas y capas de salida (figura 3.2). Estas capas de entrada forman las representaciones ortográficas, fonológicas y semánticas y están mutuamente conectadas por medio de las capas ocultas (figura 3.3). De esta manera, el reconocimiento de una palabra viene dado por todas las constricciones que surjan desde las capas de entrada en un momento concreto, es decir, la parte de la red que esté en ese momento activada a causa de las diferentes entradas, a saber, entradas semánticas, ortográficas y fonológicas. Precisamente, en la observación empírica de las constricciones ortográficas, juegan un papel crucial tanto los parámetros de frecuencia léxica, como la densidad léxica y la frecuencia acumulada de vecindario. El parámetro de frecuencia léxica mide la ocurrencia estadística de las palabras en los textos a los que las personas estamos expuestos (muestras del lenguaje). Cuanto mayor sea este índice, mayor será el uso que hacen de la palabra las personas y mayor activación tendrá en el sistema cognitivo por lo que se invertirán menores tiempos en su reconocimiento. Este es uno de los efectos más robustos con los que cuenta la literatura experimental (Álvarez, Alameda y Domínguez, 1999). También se ha constatado la intervención de otros dos parámetros, la densidad de vecindario y la frecuencia acumulada de vecindario. Ambas definiciones aluden a la forma en el que el sistema cognitivo consigue seleccionar el término a reconocer de entre otros candidatos activados con los que comparte la mayor parte de los patrones ortográficos y también la manera en que estos otros candidatos facilitan o inhiben dicho reconocimiento. La densidad de vecindario (N) se refiere al número de unidades léxicas que comparten, según unos criterios, ciertos patrones ortográficos con la referente. Esto se hace generalmente adoptando el criterio de Colheart (1977) que parece ser una de las formas que más efecto tienen y con el que se han hecho la mayoría de experimentos. Este criterio consiste en que dos palabras son vecinas si comparten todas las letras menos una y mantienen el orden de sus letras (e.g., “cerro”,”perro”). Algunos autores han empleado otros criterios para considerar los vecinos como, por ejemplo, el cambio de posición o transposición de las letras (Sainz, Mousikou y Jorge-Botana, 2003, Andrews, 1996). El segundo criterio es el de frecuencia acumulada de vecindario, que se define como la frecuencia acumulada de todo su vecindario menos la palabra

de referencia. Ambos parámetros han sido muy utilizados en la literatura experimental, en tanto en cuanto se encuentran ambos efectos en los experimentos de reconocimiento de palabras (Perea y Rosa, 2000). Todos estos efectos experimentales pueden ser explicados por los modelos en paralelo pues, según éstos, lo que se produce es una competición entre los candidatos del propio vecindario léxico de la palabra a reconocer. Los nodos en el nivel de las letras mandarán una activación excitatoria a los nodos de los términos en la medida en que estos contengan alguna de las letras activadas por la palabra a reconocer. Esta activación puede ser enviada incluso cuando no se conserva el mismo orden de las letras, pues pueden producirse efectos de gradiente en los fenómenos de transposición de letras (Perea y Lupker, 2004; Sainz, Mousikou y Jorge-Botana, 2003, Andrews, 1996). Dada una cohorte de candidatos, la frecuencia de cada uno sesgará en su favor la probabilidad de ser elegido y la frecuencia de sus vecinos disminuirá esta probabilidad. Además, esto parece refrendarse, puesto que el sistema cognitivo parece ser un detector fiel de las propiedades ortográficas de las estructuras léxicas (Seidenberg, 1987) e, incluso, el contexto formado por las letras de una palabra facilita el reconocimiento y la pronunciación de estructuras subléxicas de esa palabra cómo ocurre de una manera predominante en lenguas de ortografía no-superficial (Jared y Seidenberg, 1990).

Respecto a la parte semántica y contextual, los resultados empíricos muestran, por ejemplo, que se pueden mantener la hipótesis sobre el significado de una palabra incluso antes de haber concluido el procesamiento total de su forma, es decir, de su ortografía y fonología. Foster y Hector (2002) mostraron como antes de saber si una palabra pertenece a una categoría, el sistema cognitivo ya ha activado los candidatos de esa categoría y, por lo tanto, sólo a competición con los vecinos ortográficos de la categoría buscada enlentecen el procesamiento de la palabra presentada y no los vecinos ortográficos que no pertenecen a esa categoría. La misma tarea de categorización ejerce como contexto, lo que propicia la activación de ciertos candidatos relacionados semánticamente, al tiempo que se procesan los aspectos formales de la palabra.

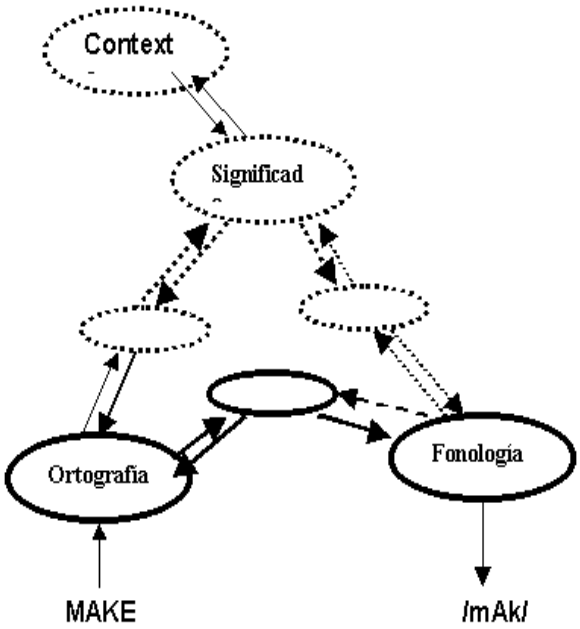


Figura 3.3.- Constricciones a tres bandas provenientes de las entradas ortográficas, fonológicas y las representaciones semánticas. Incluso se advierte la intervención del contexto de uso. En estos modelos, las palabra escrita /MAKE/ tiene una sola representación, al igual que la palabra oída /mAk/. Tomado de Seidenberg y McClelend (1989).

Estos datos empíricos provienen de experimentos con vecinos ortográficos y decisión léxica en los que los fenómenos formales de las palabras pasan a ocupar un segundo plano cuando el procedimiento de decisión léxica se desplaza al ámbito semántico mediante categorización (véase a este respecto Foster y Shen,1996; Carreiras, Perea y Grainger, 1997; Sears, Lupker y Hino, 1999). Los resultados de estos experimentos como los ya clásicos de facilitación (*priming*) semántica (e.g., McKoon y Ratcliff, 1992), corroboran de alguna manera el paralelismo masivo o en cascada que se despliega en el procesamiento de palabras. Además, otorgan verisimilitud a los modelos basados en una representación única de los términos. En el caso de palabras que posean varios sentidos como la polisemia y la homonímia (representaciones ortográficas con varios significantes), según sea el contexto de recuperación, así será seleccionado uno u otro sentido. La finura de los modelos es como representar el sesgo introducido por el contexto de recuperación. Otros modelos de reconocimiento léxico que contemplan

parcialmente aspectos de estos modelos distribuidos son el modelo DCR (Dual Cascaded Model) de Colheart, Rastle, Perry, Lagdon y Ziegler (2001), (véase la figura 3.4) elaboración actualizada del modelo de doble ruta de Colheart (1977) y el Modelo Conexionista de Múltiple Traza (MTM) de Ans, Carbonnel y Valdois (1998), aunque este último modelo contempla la creación de diferentes trazas para un mismo término ortográfico si es presentado en distinto contexto. Esto hace que este último modelo sea una excepción a los modelos que representan los términos con una sola representación ortográfica y fonológica.

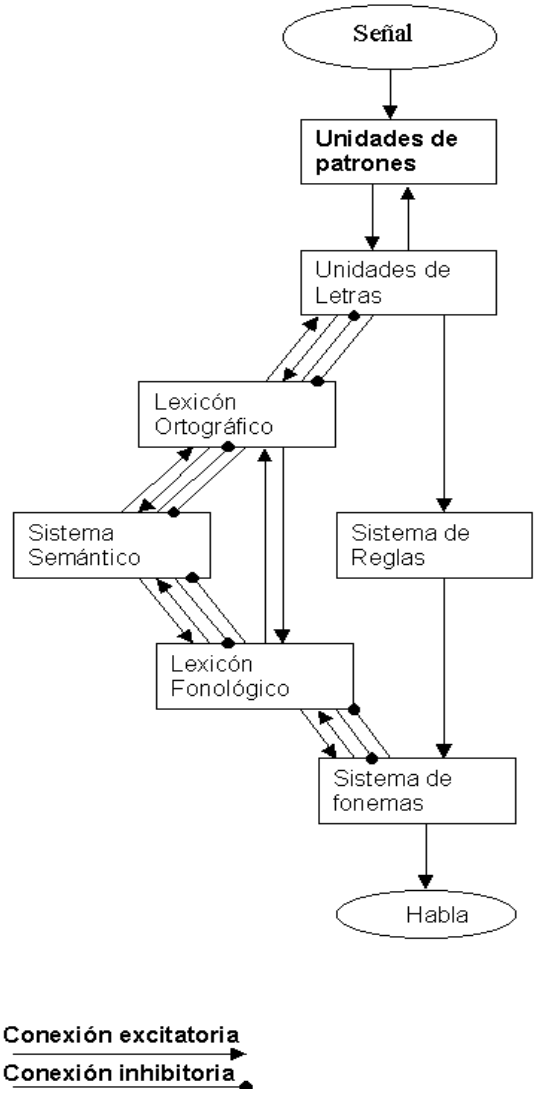


Figura 3.4.- Modelo DCR (*Dual Cascaded Model*) de Colheart, Rastle, Perry, Lagdon y Ziegler (2001), El él se integran los postulados del procesamiento distribuido y en paralelo en una estructura pseudosecuencial. Este modelo se ha mostrado bastante eficiente en la descripción de fenómenos neuropsicológicos.

Las simulaciones LSA descritas en esta tesis siguen básicamente los postulados de los modelos distribuidos y en paralelo para procesamiento léxico pero especializados en la formalización de la adquisición y distribución del conocimiento (sin representar niveles inferiores de procesamiento léxico). Incluso existen aproximaciones parecidas LSA desde arquitecturas de redes neuronales (véase Mandl 1998; McClelland y Rogers, 2003) (figuras 3.5 y 3.6) e incluso sistemas que toman los vectores salida de LSA para entrenar redes neuronales en discriminación de patrones (Mandl 1999; Mehler y Sichelschmidt, 2006).

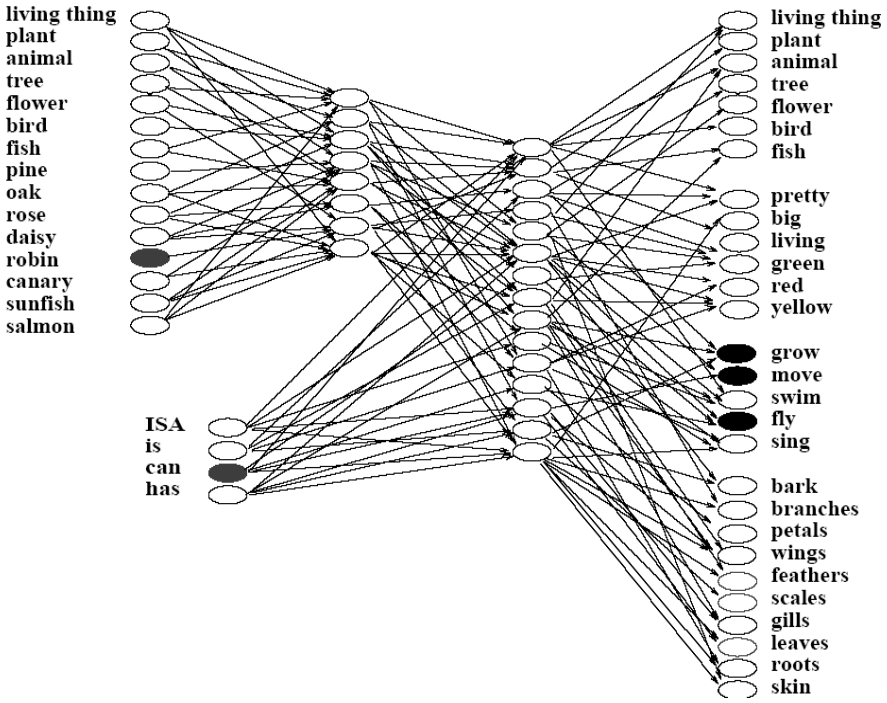


Figura 3.5.-. Ejemplo de conocimiento semántico basado en una red neural. En ella se quiere dar cuenta de las relaciones de términos insertos en proposiciones. Los términos poseen una sola representación independiente de contexto (tomado de McClelland, 2000).

En los modelos LSA, un término está representado de manera única e irrepetible por un vector, este vector será el que define su posición en forma de coordenadas. En los modelos básicos de LSA, el contexto que rodea a este vector es representado por los vectores que le acompañan en una ventana contextual dada (frase, párrafo, ensayo, etc.). Estos vectores que ejercen de contexto harán que se recalcule el vector de referencia y resulte un vector promedio del término y su contexto. De esta forma se promocionan los

contenidos del término que tengan que ver con su contexto de recuperación. El vector último será un vector sesgado por su contexto hacia uno de sus sentidos.

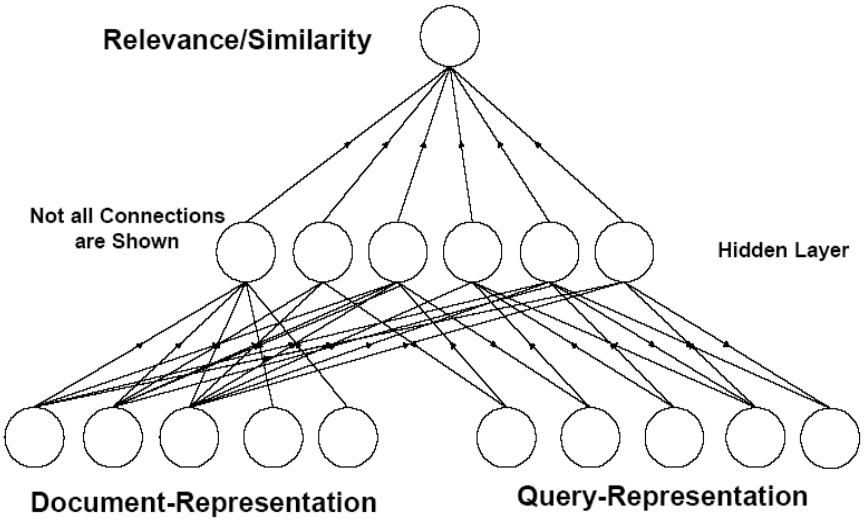


Figura 3.6.-Modelo muy parecido a LSA en cuanto a que admite comparaciones de Documentos y Pseudodocumentos (tomado de Mandl, 1998).

Aunque no sin ciertas limitaciones provenientes de la forma de incluir el contexto (Deerwester et al, 1990), el buen funcionamiento de las simulaciones con LSA refrenda en cierta manera la idea de que parece razonable pensar una única representación mental de cada término y no la representación de sus sentidos y usos de manera diferenciada. Los distintos sentidos de una palabra no están representados separadamente en la memoria sino que son generados dinámicamente como puntos evanescentes en el espacio semántico (Kintsch, 1998; Landauer y Dumais, 1997). Los conceptos son construcciones temporales de la memoria de trabajo dependiente de su contexto, no son entidades estables. El que LSA haya conseguido buenos resultados en comparaciones con juicios humanos y siendo única la representación de cada término (ortográficamente entendido), hace pensar en la posibilidad de que los modelos antes descritos estén en lo cierto. Además, existen formas y algoritmos que salvan las limitaciones de la manera clásica de sumar vectores en LSA argüidas por Deerwester et al. (1990). Con el algoritmo de predicación

Kintsch (2001) se puede modelar la comprensión de predicaciones haciendo que el valor de los argumentos de un predicado sea totalmente determinante para la formación del concepto. Entiéndase los argumentos cómo una forma de contexto para el predicado (que hace la función de polisemia). De esta forma, en vez de emplear la suma de vectores con la problemática de que la longitud de los vectores de término y contexto sea determinante, con este algoritmo se introduce un sesgo para primar el contexto de recuperación. Este algoritmo será revisado en profundidad en posteriores capítulos pero merece la pena mentar las palabras de Kintsch (2008) referentes a la polisemia:

> "*El uso de representaciones independientes de un contexto concreto, simplifica de una manera poderosa la forma de tratar la polisemia. No se ha de decidir cuántos significados y sentidos tiene un término o cuando tienen que ser recuperados unos y no otros. Lo único que se ha de tener es un único vector término independiente de contexto y un proceso que genere (algoritmo de predicación) los sentidos que emergen de ese vector*." (p.10).

Además, hay evidencia del buen funcionamiento de la representación LSA (ampliado con algunos algoritmos como el anterior) ante la comprensión de símiles y metáforas (Kintsch y Bowles, 2002; Lemaire y Bianco, 2003) que bien podrían extrapolarse estos mismos mecanismos a la comprensión del fenómeno de homonimia y polisemia. Apelando a razones funcionales y no taxonómicas, se puede concebir el modelo de comprensión de polisemia cómo funcionalmente similar al de la metáfora. La metáfora puede ser concebida como un fenómeno de polisemia en la que un término adquiere más de un significado por relación de similitud o contigüidad (metonimia en este segundo caso) cuya diferenciación es evidente en el momento de la definición, es decir, esta "viva" a día de hoy. En el caso de *"las perlas de su boca*" o *"las perlas del collar"* ambos significados están representados por un solo significante (perlas). Otro caso que se acerca más todavía al fenómeno de polisemia es la llamada *metáfora muerta*. La metáfora muerta es una variante de la metáfora que se ha incorporado implícitamente al lenguaje como por ejemplo *"la pata de la mesa"*, *"el ala izquierda de la casa"*, *"la cabeza del alfiler"*. De la misma forma, la homonimia puede ser considerada como un fenómeno de polisemia en el que

los dos significados tienen por casualidad un mismo significante o no se han extraído razones etimológicas para esta coincidencia. En este cúmulo de definiciones y relaciones entre metáfora, metonimia, homonimia y polisemia se puede intuir si cabe someramente la fina línea entre los conceptos de lingüística tradicional y puede indicar que el modelado de su procesamiento puede ser implementado con los mismos mecanismos evidenciando así iguales procesos en su comprensión. Además, encontramos más evidencia en contra de la existencia de más de una representación en estudios en los que se fuerza a LSA a contener más de una representación para una misma cadena de caracteres. Wiemer-Hastings y Zipitria (2001) propusieron un método para discriminar entre los distintos papeles morfológicos que tienen palabras con similares ortografías. Por ejemplo, la palabra inglesa *"plane"* puede tomar el valor de verbo o el valor de sustantivo. Lo que pretendían estos autores era introducirlas en el análisis con una marca que las diferencie. Para ello, marcaron cada palabra con una terminación que la identificase de una forma u de otra. En este caso se introducirá *"plane_vb"* cuando sea verbo y *"plane_nn"* cuando sea sustantivo. Cada palabra de este tipo será una palabra única. Compararon ambos tipos de corpus, el que posee palabras marcadas y el corpus estándar. En el corpus marcado, diseñaron el espacio semántico con cuatro dimensionalidades para asegurarse que los resultados no se debían a esto. Tomaron las siguientes dimensiones: 100, 200, 300 y 400. Los resultados indicaron que ninguno de los corpus marcados se comportó tan bien en lo que respecta a la similitud con los juicios humanos como lo hizo el corpus estándar. Resultados similares obtuvieron Serafín y Di Eugenio (2003) en un experimento consistente en la clasificación de diálogos telefónicos*.* Estamos, por tanto, ante una prueba consolidada de que el LSA explota el uso de las palabras en diferentes contextos. Diferenciando cada sentido y computándolo de forma diferenciada (con tantas entradas como contextos en los que aparezca), disminuye la variabilidad de usos de la representación ortográfica de un término (LSA sabe menos sobre esa palabra y los vectores que lo representan quedan mucho menos enriquecidos).

3.3.5.- Predicación

A propósito de la introducción de la sintaxis en el análisis LSA merece la pena detenerse exhaustivamente en el algoritmo de predicación desarrollado por Kintsch (2001). Este algoritmo pretende resolver algunas limitaciones que posee el LSA. Una de las principales desventajas del LSA es que no posee ningún tipo de análisis sobre las relaciones de orden de las palabras o análisis de los roles que se mantienen dentro de una determinada frase. Quizás por ello, los desarrollos LSA se desvelaban más eficientes en la comparación del párrafo que es el nivel donde, precisamente, la intervención del orden es menor o irrelevante (Wiemer-Hastings et al., 1999; Rehder et al.,1998; Landauer,2003; Kurby et al., 2003). Otra limitación es que el cálculo de la suma vectorial para representar estructuras predicativas del tipo *"El confidente sopla"* suele estar condicionado por como sea la longitud de vector de ambos términos y, por lo tanto, será difícil que el vector resultante represente su verdadero sentido. El postulado de partida que mantiene Kintsch es que el significado exacto de un predicado depende de los argumentos que le acompañan y que tanto predicado y argumentos, están constreñidos por un orden sintáctico que introduce un sesgo a cada uno de ellos. Nos detendremos seguidamente en desarrollar algo mas este punto, tomando como ejemplo el verbo "soplar":

El borracho sopla.
El alumno sopla.
El viento sopla.
El confidente sopla.

Todas estas frases comparten el verbo "soplar" cómo denominador común. Sin embargo, este mismo verbo adquiere significados distintos según nos refiramos a una u otra frase. Por ejemplo, todos sabemos que en "el alumno sopla" el verbo "soplar" no tiene el mismo sentido que en "el viento sopla". El mismo verbo adquiere unas propiedades u otras según sean los argumentos que le acompañan, es decir, las propiedades que dan significado a ese verbo son dependientes del contexto formado por sus argumentos.

Supongamos ahora la proposición PREDICADO [ARGUMENTO] expresando que el predicado adquiere unos valores u otros en función de cuales sean los argumentos. Tanto PREDICADO cómo ARGUMENTO estarían representados en sendos vectores. En la manera habitual de LSA, para calcular el vector que representase a la proposición entera, simplemente se hallaría un vector nuevo que fuese la suma o el "centroide" del vector ARGUMENTO y el vector PREDICADO. Si la representación de los vectores fuese,

V.PREDICADO= {p1,p2,p3,p4,p5,…,pn}

V.ARGUMENTO= {a1,a2,a3,a4,a5,…,an}

Entonces, la representación de la proposición sería:

V.PROPOSICIÓN = {p1+a1,p2+a2,p3+a3,p4+a4,p5+a5,…,pn+an}

Esta manera no es la mejor manera de representar las proposiciones, pues no contempla la dependencia del predicado en relación con los argumentos. En otras palabras, al computar el vector de la proposición entera no necesitamos todas las propiedades del PREDICADO, sino simplemente aquellas que conciernen al significado de los argumentos. En el ejemplo, no necesitaremos todas las posibles propiedades del verbo "soplar", sino sólo aquellas que quedan restringidas por el valor de los respectivos argumentos (*viento, alumno, borracho, confidente*). Por tanto, lo que hace el centroide o suma vectorial en la manera LSA es tomar todas las propiedades sin discriminar en función de los argumentos y sumarlas a las del argumento. Así, todas las propiedades del verbo "soplar" se tendrán igualmente en cuenta a la hora de calcular el nuevo vector. En el caso de que el argumento tuviese una longitud de vector menor que el predicado y, además, en dicho predicado estuviesen mejor representados otros argumentos que no fuese el argumento que analizamos, el vector que representa la predicación no captará tampoco el verdadero sentido. De esta forma, estarían mejor representadas en el vector las propiedades de los argumentos que más representación tuviesen en el espacio semántico y no las del argumento que se está predicando. Este método promueve, por el contrario, que sea la longitud de los vectores-términos

involucrados la que finalmente imponga qué propiedades adquirirá el vector que represente a la predicación. De esta manera, el vector más largo dominará el significado del nuevo vector suma que represente la proposición o, dicho de otra manera, si calculásemos el "centroide" o la suma de dos términos insertos en la proposición, ésta se sesgará en favor del término con la mayor longitud de vector (y las propiedades dominantes de éste).

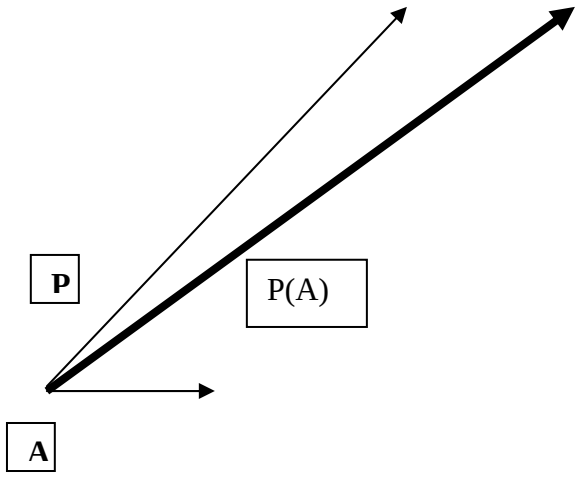


Figura 3.7.-. Esquema general de los vectores de una predicación**.**

Siendo P el vector del predicado y A el vector de los argumentos, la forma estándar de calcular el vector P(A) sería la suma de vectores (véase la figura 3.7). Como no pasa desapercibido en el gráfico, la solución queda sesgada por la longitud de los vectores. Combinando un vector de gran tamaño con otro reducido, el resultado devolverá un vector similar al de gran tamaño. Puesto que en este caso el vector del predicado es de mucha mayor longitud que el vector del argumento, el significado de la proposición final no tendrá en cuenta las restricciones que ejercen los argumentos sobre el predicado y tomará las propiedades de sus argumentos más frecuentes. De esta manera, colocar unos argumentos u otros no incidirá apenas en el significado de las proposiciones. Siguiendo con nuestro ejemplo, imaginemos el caso de "soplar", supongamos que soplar tuviese en nuestro espacio semántico una longitud de vector de 2,5 y que "confidente" tuviese una longitud de 0,5 (figura 24). Estos

datos indicarían que nuestro espacio semántico sabría más del término “soplar” que de “confidente”. Además, supongamos que “globo” tuviese una longitud de vector de 2.0 y “viento” de 2,1 siendo términos conocidos por LSA y que, además, estuviesen, cómo resulta previsible, relacionados con “soplar”. Dado este supuesto queremos calcular un vector que representase la proposición “El confidente sopla”.

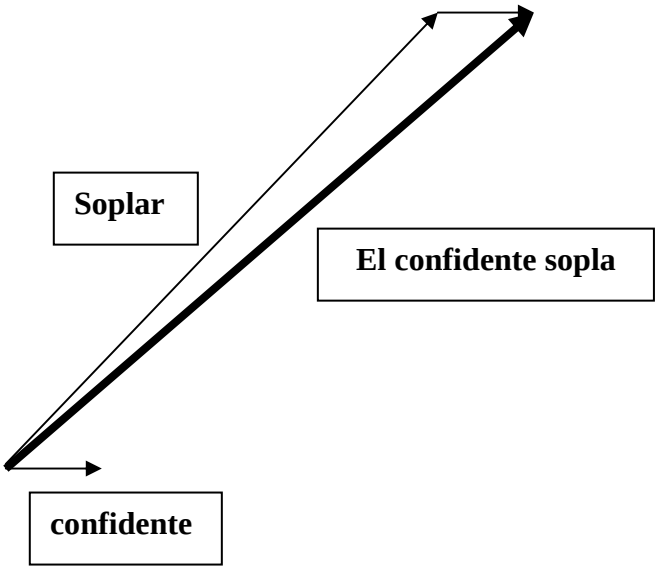


Figura 3.8.-. Superioridad de la longitud de vector del predicado en relación a su argumento.

Como puede observarse en el gráfico 3.8, al ser la longitud de “soplar” cinco veces mayor que la de “soplar”, la aportación de “confidente” al vector de la proposición final “el confidente” resulta muy pequeña. Por eso, el vector de la proposición estará compuesto por características generales del verbo “soplar” (las propiedades predominantes), sin acotar su significado al contexto que proporciona el argumento “confidente”. Probablemente, será el caso de que el vector resultante de “el confidente sopla” estará aún relacionado con frases-referencia relacionadas con el “viento” y con los “globos” como, por ejemplo, “hace un aire”, “está compuesto de gas”, y quedará menos relacionado con “confesó su delito”. Esto es a lo que se refiere Kintsch (2001) al señalar que en toda predicación, los predicados resultan dependientes de sus argumentos y esto tiene que ser asumido, de alguna manera, en los modelos que tratan de

formalizar el procesamiento de las cláusulas. El método del centroide falla en esto y promociona los contenidos generales a la hora de interpretar contextos concretos dentro de las frases. Por el contrario, lo que hace el algoritmo de predicación es sesgar la longitud del vector, añadiendo un contexto adecuado al tipo de argumento que se está predicando. Este contexto estará formado por los vecinos semánticos del predicado que también estén relacionados con el argumento. El procedimiento es ingenioso a la par que sencillo. Los pasos a seguir podían enumerarse en los siguientes:

1) Se localizan predicado (“soplar”) y argumento (“confidente”) dentro de una proposición. P(A).
2) Se extraen los n vecinos más próximos al predicado (P). Dado un espacio semántico, se calcularán los cosenos de P con cada uno de los términos que componen el espacio semántico. Hecho esto, se seleccionan los n primeros términos que más se aproximen a P (la elección de n queda abierta al tipo de modelo que se desee hacer y observaciones empíricas).
3) Se calculan los coseno entre cada uno de los n términos elegidos y el argumento(A).
4) Se implementa una red (véase la figura 3.9) en la que se establecen conexiones inhibitorias entre los n términos y conexiones excitatorias entre cada uno de los n términos con el argumento (A) y con el predicado (P). Las conexiones se harán en base a los cosenos calculados en el paso 2 y 3. En definitiva, el objetivo es localizar objetivamente aquellos vecinos semánticos del predicado (P) que a la vez sean pertinentes para el argumento (A). De ahí que se implemente una red que localice los términos que se sitúen cómo vecinos del predicado y que sean pertinentes también para el argumento(A). Esta red no necesita entrenamiento previo pues es el propio corpus que se procesó con LSA, el que lo llevo a cabo.
5) Cómo último paso, se calculará el vector P(A) con la suma o centroide del Predicado (p), más el argumento(A), más los términos que reciban más activación en la red implementada, es decir, aquellos que reciban más activación excitatoria de Predicado y argumento y menos inhibición lateral de los propios términos de su misma capa. El número de términos seleccionados, queda también a criterio del tipo de modelado que se esté llevando a cabo y observaciones empíricas a posteriori. Hecho esto, tendremos un vector P(A) en el que se tiene en cuenta que el sentido que acoja el predicado está en función

del argumento que lo acompaña pues, la suma final la compondrán también vectores de términos que siendo vecinos del predicado, son pertinentes para el argumento.

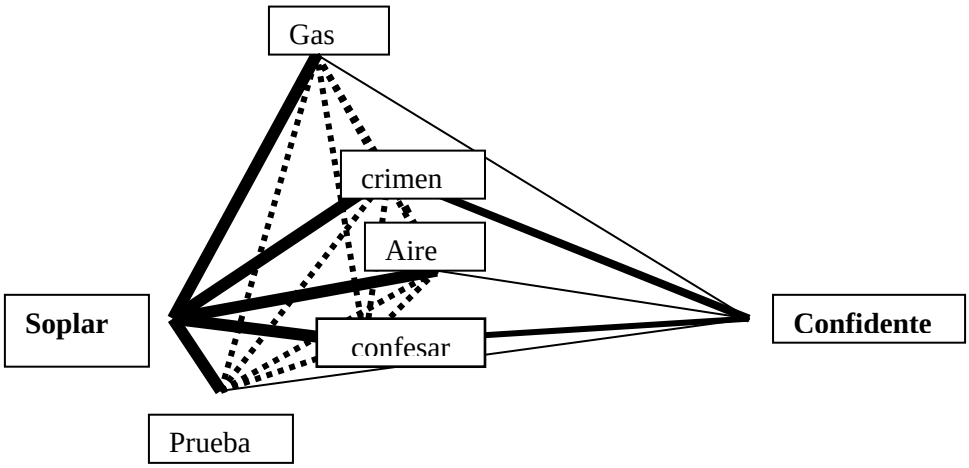


Figura 3.9.-. Red de predicación.

Los términos más activados serán los que reciban mayores conexiones excitatorias por ambos lados, es decir, dadas sus conexiones con Predicado y Argumento, las palabras que tengan altos cosenos en ambos lados serán las que serán más activadas y mandarán conexiones inhibitorias al resto. En este caso hipotético, serán los términos “confesar” y “crimen” los que resultarán más activados. Así, “crimen” y “confesar” serán las palabras que sumadas al Predicado(P) “Soplar” y al Argumento(A) “Confidente”, configurarán el vector de la proposición completa (Figura 3.9). De esta forma, se impone un sesgo a la manera estándar del centroide para que contemple el fenómeno lingüístico de que el sentido del predicado es dependiente de la información proporcionada por sus argumentos. Kintsch (2001) empleó cómo validación la comparación entre los vectores que resultaban de las proposiciones y algunas referencias conocidas. Este método se muestra eficaz para comprobar la dirección de las proposiciones calculadas mediante en algoritmo de predicación. En este artículo ofrece un ejemplo de cómo validar los resultados. Dadas estas tres frases:

"The bridge collapsed"
"The plan collapsed"
"The runner collapsed"

Se calcula mediante el algoritmo de predicación los nuevos vectores para cada proposición siendo "collapsed" el predicado y "bridge", "plan" y "runner" sendos argumentos. Pongamos como traducción más acertada al español que "collapsed" sea "venirse abajo". De manera que tenemos otra vez un verbo cuyo sentido depende de los argumentos introducidos. Para comprobar que cobran un sentido con cierta verosimilitud sobre la realidad, comparamos cada nuevo vector con una referencia que esté relacionada con cada campo semántico: "break down", "failure" y "race". Parece que ahora los resultados se ajustan a la realidad.

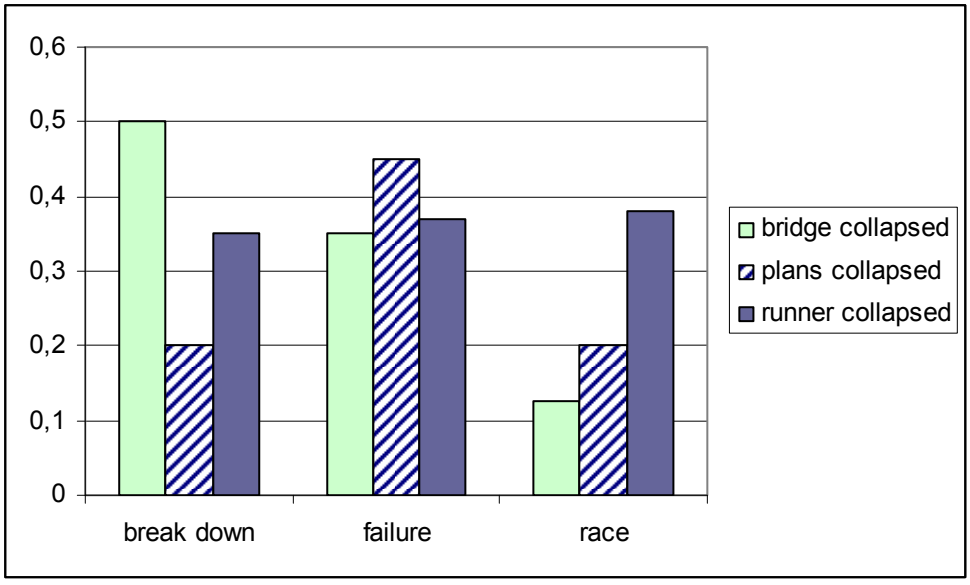


Figura 3.10.- Gráfico donde puede observarse que cada una de las proposiciones es eminentemente sesgada para asemejarse a sus referencias conocidas. Esto es porque sólo los vecinos semánticos de "collapsed" que son relevantes para cada argumento han participado en el cálculo del vector de cada proposición.

Kintsch (2001) nos sigue mostrando más ejemplos: Los términos "Pelican" y "Bird" tienen una longitud de 0,15 y 2,04, respectivamente (figura 3.10). Ello refleja que LSA conoce más sobre "*Bird*" que sobre "*pelican*". Si nosotros quisiéramos calcular el vector resultante de la proposición "*Pelican is a Bird*" mediante la suma vectorial, es decir *Pelican+Bird*, encontraríamos que

nuestro vector se parecería mucho más a "*bird*" que a "*Pelican*". Prácticamente encontraríamos que "*Pelican is a Bird"* será idéntico a "*Bird*" además de comportarse igual que "*Bird*" en el espacio semántico, ya que su coseno estaría más cerca de proposiciones referidas a los pájaros en general. Si quisiéramos calcular la proposición "*The bird is a pelican*", calculando el centroide extraeríamos semejantes resultados (figura 3.11) No habría ninguna variación ya que el procedimiento estándar no hace ningún análisis sobre el orden de los constituyentes ni sobre que rol ejerce cada uno.

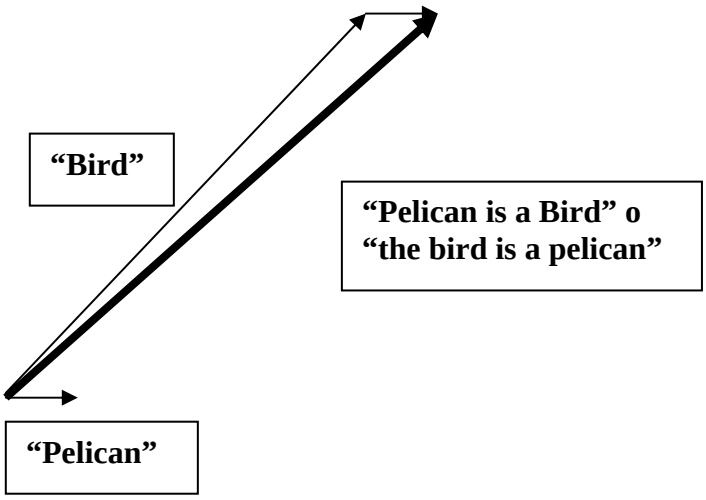


Figura 3.11.- Para la forma clásica de la suma vectorial, "Pelican is a bird" es representado por el mismo vector que ""the bird is a pelican"

Sin embargo, el algoritmo de predicación si daría distintos resultados con una proposición que con otra. Con la primera proposición, "*Pelican is a bird*" al igual que los resultados del centroide, disiparía las características de "*pelican*" convirtiendo el vector de la proposición en un vector casi similar a "*bird*". Al decir que el pelícano es un pájaro, se haría alusión a las propiedades de los pájaros y estas son las que construirían el vector de su proposición. De esta forma, la proposición "*the pelican is a bird*" estaría pareja al término "bird" y se comportaría como tal en el espacio semántico, de manera que midiendo la semejanza por medio de los cosenos, la proposición perece estar más cerca de predicaciones de los pájaros en general como "*sing beautifully*" que de los pelicanos como "*eat fish" y "sea*". Sin embargo, invirtiendo la proposición a "*the bird is a pelican*" los resultados cambiarían y se acercarían esta vez a las

propiedades de los pelicanos, ya que esta vez se está predicando que “el pájaro es un pelícano”. Es decir, en esta última frase interesa saber cuáles son las propiedades de los pelícanos para predicarlas sobre pájaro”. Sobre este punto se añaden las siguiente notas finales sobre el algoritmo:

1) Para unidades de documento mayores que este tipo de proposiciones, este algoritmo se comportaría como el centroide.
2) No está del todo claro cuál es el tamaño del vecindario que implementa la red y cuál es el número de vecinos activados que servirán para calcular el vector, quizás en futuras investigaciones se constate alguna constante exitosa.
3) El algoritmo de predicación emplea el espacio semántico de LSA para representar de manera estática el conocimiento y emplea una red de expansión de la activación encima para introducir un elemento de modificación dependiente del contexto. Este último caracterizará el proceso de comprensión.

3.3.6.- La metáfora

Uno de los fenómenos que mejor modela este algoritmo es la comprensión del lenguaje metafórico (la metáfora predicativa). Para Kintsch (2000), la comprensión de esta metáfora se rige por las mismas leyes de cualquier otra predicación en el sentido de que el predicado tomará las propiedades del argumento, en este caso, la metáfora o símil.

Tomemos un ejemplo extraído de este mismo trabajo,“*My lawyer (A) is a shark(P*)” (mi abogado es un tiburón). A nadie pasa desapercibido que el significado con el qué se quiere dotar a esta frase no corresponde al sentido literal de “tiburón”. No queremos decir que “nuestro abogado tiene aletas” ni que “nada por los mares caribeños” sino que, probablemente, nos referiremos a que “nuestro abogado es muy efectivo” o que quizás que “nuestro abogado es muy despiadado”. Por tanto, tenemos que dotarnos de un algoritmo que tenga presente esto y, además, que tenga presente que son sólo algunas

propiedades de “tiburón” las que estamos predicando y no todas. Al introducir el argumento “abogado”, estamos induciendo a tener presentes sólo algunas de estas propiedades.

Los resultados con el método del centroide o suma vectorial (Figura 3.12) muestran ciertos efectos que no tienen sentido, “mi abogado es un tiburón”, probablemente influido por la longitud del vector de “shark”, devuelve un vector relacionado con palabras como “fish”(pez) y “shark”(tiburón) más que con palabras relacionadas con “lawyer”(abogado). Sin embargo, con el algoritmo de predicación, los resultados son mucho más ajustados a la realidad. Esta vez se toman 500 vecinos semánticos para calcular el vector de la proposición ya que para dar cuenta de los efectos metafóricos, se hace necesario un mayor número de vecinos pues las palabras que están en juego se encuentran mucho menos relacionadas que las construcciones literales (véase “abogado” y “tiburón” frente a “venirse abajo” y “puente”). Una vez calculado el vector resultante, se validan con el procedimiento antes descrito: las referencias conocidas.

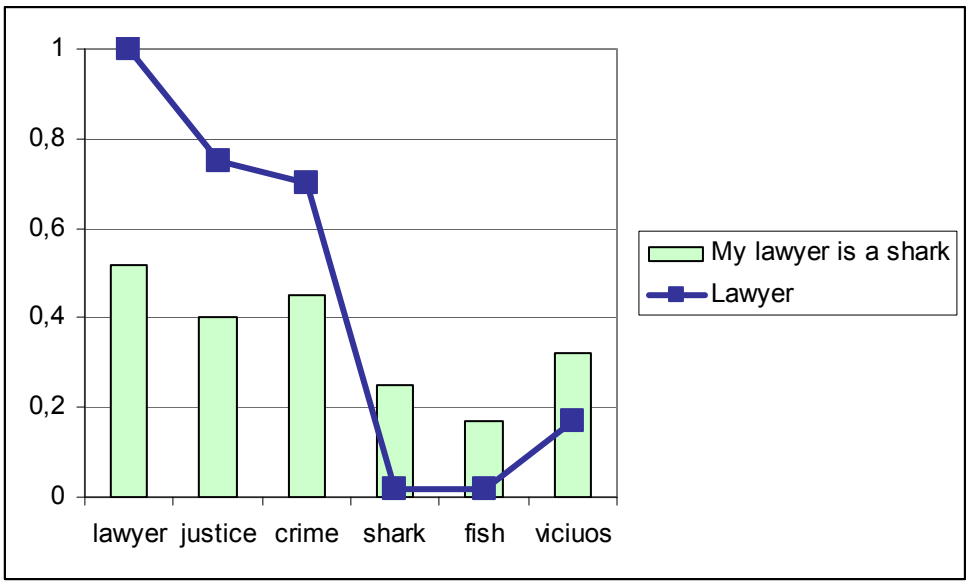


Figura 3.12.- Representación de la predicación “*My lawyer is a shark*” y del términos “*Lawyer*” (tomado de Kintsch, 2000).

Cómo se puede observar en el gráfico, predicando “tiburón” de “abogado”, se obtienen resultados diferentes de “abogado” aisladamente. “abogado” está estrechamente ligado a “justicia” y “crimen” y no mantiene

ninguna relación con “Tiburón” ni “pez”, pero está moderadamente ligado a “despiadado” o “cruel”. Predicando que “abogado” es un “tiburón” se obtiene unos resultados algo distintos. Las propiedades de “abogado” permanecen (“justicia” y “abogado” y “crimen”), pero lo más interesante es que las propiedades de “tiburón”, son ensalzadas en concordancia con la idea de que la proposición significa que mi “abogado” es “despiadado”. Además, también aparece, aunque en pequeña proporción, propiedades de “pez” y de “tiburón”. El significado de la metáfora ha copado en un cierto gradiente el sentido de la proposición. El significado de la metáfora se hace difuso y rico en matices. Siguiendo con esta metodología Kintsch y Bowles (2002), indagaron sobre las variables que hacen a unas metáforas más difíciles de entender que a otras. Diseñaron unas condiciones en las que tanto el modelo cómo sujetos experimentales se enfrentasen a dos grupos de metáforas: un grupo de metáforas fáciles y otro de metáforas complicadas. Tomando el modelo y comparándolo con las respuesta de los sujetos, encontraron que tanto estos como el modelo tienen diferente rendimiento ante los distintos grupos de metáforas (fáciles y difíciles) en la misma dirección. Las metáforas “difíciles” tienen más omisiones (7%) y mucho menos acuerdo en lo que respecta a su significado (en los sujetos experimentales). Aún así, las interpretaciones que se da a las “difíciles”, no por ser más variables son totalmente azarosas pues dista mucho de asemejarse a lo pronosticado estadísticamente si llegase a ser una distribución azarosa. La parte más importante de este trabajo es la que concierne a las variables que hacen que unas metáforas sean más difíciles que otras. Señalamos las siguientes:

- La primera tentativa podría ser la simple distancia semántica entre el argumento y el predicado de la metáfora. Eso se descarta ya que se comprueba que el promedio de los cosenos entre argumento y predicado es igual en las sentencias que representan las metáforas difíciles y fáciles. Esto invalida este primer argumento.
- Otra posibilidad era que la dificultad de la interpretación radicase en la cantidad de información de que portan tanto argumento cómo predicado. Esto, de nuevo es invalidado a comprobarse que tanto frases con metáforas fáciles como difíciles tienen un promedio de longitud de vector que no difiere significativamente.

- La siguiente posibilidad podría ser el número de vecinos que tienen ambas palabras. No hay diferencia entre los predicados de las frases que contienen metáforas fáciles y difíciles. Tampoco hay diferencias entre la longitud de vector de los argumentos de ambos tipos de frases. Parece, pues, que la dificultad de la interpretación de una metáfora no radica en propiedades de los términos aisladamente.
- La siguiente hipótesis es que la dificultad dependa de que haya ciertos términos de los vecindarios de Predicado y Argumentos que sean comunes y de cómo estos mantienen la relación tanto con predicado como con el argumento.

    - No siendo el promedio del coseno entre el predicado y ese grupo común de palabras significativamente diferente, queda la posibilidad de que sea el argumento la parte clave en la dificultad.
    - La diferencia entre el promedio de los cosenos entre grupo común de palabras (entre predicado y argumento) y el argumento es significativamente diferente en “fáciles” y “difíciles”. Esto hace suponer que lo que diferencia a ambos tipos de metáforas depende de la relación que guarda el Argumento con los vecinos del predicado que son recurrentes a ese mismo argumento. En otras palabras, una metáfora será fácil si el argumento tiene una relación semántica estrecha con los vecinos del predicado que tienen relación con el propio argumento. Si el conjunto de términos que tienen relación con ambos, predicado y argumento, tiene mucha relación con el argumento, la interpretación será más sencilla que si la relación entre argumento y ese conjunto de vecinos de ambos es poca.

3.3.7.- Morfología y sintaxis

Las teorías actuales afrontan el reto de tener que explicar cómo se crean las estructuras sintácticas, el modo en el que el lector selecciona e interpreta una estructura sintáctica y cómo esas estructuras interpretadas se integran en los modelos mentales de discurso*.* Sobre la mayor o menor importancia de las estructuras sintácticas han fluctuado la mayor parte de los modelos propuestos, poniendo unos mayor énfasis en los aspectos sintácticos y estructurales como

es la teoría de Garden-Path (Frazier, 1979; Frazier y Rayner, 1982) y otros incidiendo en la frecuencia de uso, la estructura léxica, la plausibilidad y el contexto para la resolución y adjunción de los roles en la oración (MacDonald 1994; Tanenhaus y Trueswell, 1995). También los teóricos de LSA se han interesado en dicha discusión y han tratado de integrar aspectos sintácticos en el modelo LSA para poner a prueba la intervención de la sintaxis tanto en la adquisición de conocimiento como en la comprensión de oraciones aisladas.

En cuanto a la intervención de las estructuras sintácticas en la adquisición de un modelo de conocimiento, Landauer, Laham, Rehder y Schreider (1997) acometieron dos experimentos para tratar de comprobar que la sintaxis tiene una participación mínima en la adquisición de un modelo de conocimiento. Compararon la eficiencia de un método como LSA a la hora de comprender texto con una muestra de humanos. Como sabemos, el LSA no contempla el orden ni la organización de las palabras en la implementación de sus algoritmos. Humanos y LSA se comparan ante las mismas tareas y demandas. Si el resultado fuese que el LSA puede hacerlo tan bien como los humanos, quiere decir que la entrada de LSA (palabras sin computar el orden ni la estructura) sería suficiente para explicar con parsimonia la adquisición de un modelo de conocimiento. En otras palabras, si LSA se muestra efectivo en la tarea, será porque la información sintáctica puede ser, de alguna manera, redundante (Landauer, 2002). Dicho de otra forma, el uso repetido de las palabras y la evocación de sus características es un proceso muy eficiente para inferir información referente a sus roles temáticos. Además, como señalaron Landauer et al. (1997), lo que se pone en juego en el artículo no es la comprensión de las frases aisladamente, sino la adquisición de habilidades sobre un cierto tema a partir de grandes cantidades de textos. Es importante especificar que la comprensión de las frases de manera aislada es un tema que transciende la metodología de estos experimentos. La comprobación se hace de la manera siguiente: Se entrena al LSA en un texto sobre un tema concreto. Es importante señalar que se toma la frase como unidad de documento, esto es, el entrenamiento al que es sometido LSA es a base de frases. La pregunta que se formula es la siguiente, ¿es capaz el LSA de tener un rendimiento parecido a los humanos en lo que se refiere a dominio y evaluación de un

temario con cuyas frases fuese entrenado el LSA? Tanto en el primero como en el segundo experimento los resultados mostraron que las evaluaciones hechas por LSA superaron incluso a las de los humanos y cuyas correlaciones fueron más bajas. Así, en el primer experimento, la media de las correlaciones entre LSA y un criterio externo fue de 0.81, mientras que para la muestra de estudiantes fue de 0.70. Este patrón se repitió en ambos experimentos, incluso tomando la longitud del vector cómo medida. Lo que se prueba con estos experimentos es que la adquisición del significado de los pasajes puede ser llevado a cabo sin necesidad de tener en cuenta el orden ni la organización de las palabras. Según estos autores, la sintaxis podía entonces restringirse a funciones subsidiarias como evitar la sobrecarga de la memoria de trabajo, construir el lenguaje de manera secuencial y subdividir las producciones en secuencias.

Estos resultados pueden ser cuestionados por los trabajos realizados por Padó y Lapata (2008), quienes entrenaron a los modelos semánticos vectoriales con corpus en los que los documentos estaban representados por unidades sintácticas plausibles en forma de tuplas, es decir, construyendo los espacios semánticos con ayuda del conocimiento lingüístico-sintáctico. Estos espacios se comportan de una manera superior a los espacios construidos con la manera clásica. Este experimento muestra que los datos de Landauer et al. (1997) pueden ser relevantes pero matizables, y que el análisis sobre la sintaxis puede influir en la adquisición de conocimiento. En otro experimento, Wiemer-Hastings y Zipitria (2001) introdujeron cierto control sobre la sintaxis, identificando las relaciones y acciones de los participantes en cada frase. Esto es lo que llamaron “Quién hizo qué a quién”. Para hacer esto separa sujeto, verbo y objeto. A esta segunda aproximación la denominaron SLSA (*Structured Latent Semantic Analysis*). El procedimiento que sigue fue el siguiente: 1), resolver las anáforas pronominales reemplazando los pronombres por sus antecedentes; 2), dividir las frases complejas en cláusulas simples; y 3), segmentar las frases simple en sujeto, verbo y objeto. Además las frases pasivas son normalizadas también en Sujeto, Verbo y Objeto. En el ejemplo, *“RAM stores things being worked with, and it is volatile”* pasa a ser (“stores” “RAM” “things being worked with”) (“volatile””RAM”). De cada frase se formaron

tres cadenas: Verbo, lo que acompaña al sujeto y lo que acompaña al objeto. Para calcular las puntuaciones de similitud entre dos frases con este diseño, primero se siguen todos estos pasos (preproceso). Luego se promedian las tres puntuaciones que se generan de comparar cada cadena con la correspondiente de la otra frase (verbo, objeto, sujeto) y se obtiene una puntuación general. Los resultados son que SLSA obtiene mejores resultados con respecto a su correlación con los juicios humanos que la LSA estándar. Esto, dice el autor, no es tampoco consistente con la afirmación de Landauer, et al. (1997) de que la sintaxis simplemente ayudaba a quitar carga a la memoria corto plazo y que no e importante para adquirir conocimiento.

Respecto a la comprensión de frases (no la adquisición de conocimiento, que es cosa distinta), los trabajos de Landauer, et al. (1997) también pueden cuestionarse por los modelos LSA en los que se introducen pequeñas secuencias de sintaxis para mejorar la comprensión de predicaciones y metáforas (Kintsch, 2000; Kintsch, 2001; Kintsch y Bowles, 2002). Otro ejemplo en donde se simula el procesamiento sintáctico con más complejidad son los árboles de dependencia combinados con LSA (Kintsch, 2008). Esta forma ha obtenido muy buenos resultados en la simulación de la comprensión de frases en comparación con los cosenos de LSA estándar y quizás sea una manera más refinada que la utilizada por Wiemer-Hastings (2000), de introducir aspectos sintácticos en el modelo en la memoria estática representada por los modelos LSA. Como el algoritmo de predicación fue tratado en profundidad en capítulos anteriores, sólo resaltaremos que los mismos mecanismos que lo sustentan sirven a Kintsch (2008) para armar una teoría sobre cómo se van construyendo los vectores resultantes a medida que se van teniendo en cuenta las constricciones impuestas por las estructuras sintácticas.

Respecto a la morfología, Landauer (2002) hacen manifestaciones similares. Partiendo que en los modelos LSA, los plurales y singulares, los distintos tiempos verbales, las distintas flexiones, etc., tienen índices que reflejan alta similitud. De la misma forma que la sintaxis, es posible que las relaciones inductivas entre las palabras sean suficientes para captar la similitud

entre términos con raíces comunes, sin apelar a reglas morfológicas. Wiemer-Hastings (2000) introdujo la posibilidad de un análisis sobre texto en el que estuviera representada la morfología y la sintaxis. En esta aproximación consignó un descenso de la efectividad de la propia técnica. Este primer intento de introducción de elementos estructurales en el análisis LSA sorprende al autor quien en un artículo posterior achaca este descenso a algunas peculiaridades en el diseño. En un artículo posterior, Wiemer-Hastings y Zipitria (2001) hicieron una revisión de su anterior intento y realizaron dos experimentos. Los resultados mostraron que ninguno de los corpus marcados se comportó tan bien en lo que respecta a la similitud con los juicios humanos como lo hace el corpus estándar sin marcar. Semejantes resultados son obtenidos por Serafín y DiEugenio (2003). Esto es, en definitiva, una prueba de que el LSA explota el uso de las palabras en diferentes contextos. Diferenciando cada sentido y computándolo de forma diferenciada, disminuye la variabilidad de usos de la representación ortográfica de un término (LSA sabe menos sobre esa palabra y los vectores que lo representan quedan mucho menos enriquecidos).

3.3.8.- Isomorfismo de segundo orden

Dada su naturaleza completamente abstracta y formal, LSA ha servido para mantener posiciones teóricas en contra de la absoluta necesidad de que todo símbolo tenga identificación en el mundo real. Desde las teorías corpóreas (Barsalou, 1999) se ha mantenido que todo símbolo tiene que ser vinculado con el mundo “real” antes de poder tener significado. El significado de las palabras está en estrecha conexión con el mundo, no es una abstracción independiente de él, arbitraria y amodal de la realidad. Estas teorías tendrían su basamento en el isomorfismo de primer orden (Shepard, 1987).

Sin embargo, dado el éxito de LSA sin que exista un mundo real al que pueda identificarse (ya que LSA es sólo una teoría lingüística), se ha propuesto que es un buen modelo para demostrar que el mundo simbólico es un espejo del mundo real, pero que este último no resulta necesario para generar los significados (Kintsch, 2008), sino que su participación simplemente es

enriquecedora. Para estas teorías del símbolo, el significado de una palabra no tiene un isomorfismo real, sino indirecto y en conexión únicamente con el resto de palabras. El significado únicamente tiene sentido en el LSA, considerando la posición del resto de las palabras en el espacio semántico. Para Kintsch (2008) el LSA es una teoría de la comprensión del lenguaje que, en palabras de Shepard (1987), podría llamarse isomórfica de segundo orden, donde, efectivamente, el LSA no contiene los significados directos de las palabras al estilo de un diccionario (isomórfica de primer orden), sino únicamente la relación de unas con las demás.

3.3.9.- Evaluación de resúmenes (emulación del juicio de un experto)

Fruto de su capacidad para la clasificación de documentos, LSA se ha aplicado también para la simulación del juicio del experto a la hora de corregir textos de respuesta a preguntas abiertas. Este tipo de experiencias se han llevado a cabo masivamente en el ámbito académico de muy diversas formas y empleando distintos parámetros (véase para una revisión, Trusso, 2005; también Landauer y Psotka, 2001, para una compilación de trabajos). El procedimiento consiste en entrenar un corpus específico de un dominio de conocimiento y comparar de forma vectorial los textos extraídos de las respuestas de los alumnos con textos que representen las formas ideales de respuesta a las preguntas propuestas. Landauer, Laham y Foltz (2003) muestran los pasos a seguir en la aplicación de la herramienta construida por ellos llamada *Intelligent Essay Assesor* (IEA). Destacamos aquí los siguientes:

- Aplicar LSA a un texto para representar el significado de las palabras de ese dominio. Se obtendrán mejores resultados en la medida en que el corpus sea relevante para ese dominio. Será mejor un libro entero sobre una disciplina que un capítulo.

- Hacer acopio de una muestra representativa de respuestas o “ensayos” puntuados por humanos.

- Representar cada una de las respuestas a puntuar y cada una de las respuestas previamente puntuadas de forma vectorial dentro del espacio semántico.

- Calcular la semejanza entre las respuestas a puntuar y cada una de las respuestas puntuadas previamente.

- Con esta comparación entre las respuestas a puntuar y las puntuadas previamente, se realizará un computo por el cual se infiere en qué nivel esta el alumno autor de la respuesta.

Como ejemplo de funcionamiento, sirvan los datos proporcionados por Landauer et al. (2003). Una colección de ensayos de estudiantes de universidad sobre las redes neuronales fueron corregidos por tres tipos de jueces, becarios aún sin licenciar, becarios licenciados y profesores, todos ellos sin conocer el juicio emitido por el sistema de IEA. La correlación entre los juicios del sistema y los evaluadores humanos fue de 0.69 para los jueces no licenciados, 0.78 para los ayudantes licenciados y de 0.80 para los profesores. Esto muestra que además de valorar el dominio para el que fue entrenado, LSA es capaz de discriminar las distintas respuestas clasificándolas en niveles de habilidad.

Otras experiencias en torno a la clasificación de los textos académicos proviene de Graesser et al. (2000), quienes diseñaron un herramienta llamada AUTOTUTOR capaz de, al igual que Landauer et al. (2003), clasificar las respuestas de los alumnos en diversos niveles de habilidad gracias a respuestas ideales previamente puntuadas. También la aplicación llamada **Summary Street** diseñada por Wade-Stein y E. Kintsch, (2004)*.* Esta aplicación promociona el aprendizaje de los alumnos a través de la escritura de resúmenes y la retroinformación de cómo estos han sido escritos y si cubren los tópicos e ideas principales. El objetivo es invitar al alumno a la reescritura y reformulación de los resúmenes incidiendo sobre las ideas capitales, reduciendo así la carga del profesor. Los índices de eficiencia fueron altos obteniendo una de 0.64 entre el profesor y el sistema LSA. Además, encontraron que la herramienta ejerce como factor motivacional, pues los alumnos reformulan sus resúmenes en muchas más ocasiones que los que trabajaban con un procesador de texto. Laham, Bennet y Landauer (2000) diseñaron un sistema basado en LSA que tiene como objetivo asignar al personal de la fuerza aérea de EEUU (USAF) a diferentes tareas y planes de entrenamiento, además de recomendar las sustituciones de destinos más

convenientes. También Jared, Thomson y Cohen (2000) emplearon LSA en el ámbito de las fuerzas armadas, pero esta vez con textos producidos por oficiales de la defensa antiaérea de EEUU. El dominio de conocimiento se centró en distintos escenarios posibles en el ámbito de la guerra antiaérea y sus posibles resoluciones.

En resumen, todos los trabajos en torno a la simulación del juicio del experto tienen como elemento común la comparación de las respuestas ofrecidas por los alumnos con respuestas ideales baremadas en torno a su nivel (a veces puede ser una sola respuesta ideal del nivel más alto). La manera en que se realicen las comparaciones y las inferencias que se hagan sobre las puntuaciones, pueden suscitar en ocasiones problemas de metodología. Al análisis de estos problemas son los que nos referimos a continuación.

3.4.- Limitaciones

3.4.1.-Sobreestimación de la similitud con el coseno

Cuando estimamos los índices de semejanza basándonos en el coseno, surge un problema que se ha hecho si cabe más patente en las aplicaciones destinadas a juzgar los ensayos en el ámbito académico. El problema se hace patente cuando se requiere comparar una producción de un alumno (sea esta una respuesta a una pregunta que debe desarrollar), con una respuesta marcada por expertos cómo ideal. El alumno puede optar por desarrollar su respuesta introduciendo un número muy reducido de términos que, sin embargo, son representativos y casi repetitivos de la pregunta que se hace. Siendo esta la situación, se da el caso de que la respuesta dada por el alumno está muy ajustada, vectorialmente hablando, a la respuesta ideal marcada por los expertos (Figura 3.13). Sin embargo, las longitudes de vector de ambos, son extremadamente diferentes. La longitud del vector que representa la respuesta ideal es mucho mayor que la longitud del vector que representa la respuesta del alumno. En el plano puramente aplicado, esto introduce un gran sesgo en la baremación de la respuesta del alumno, ya que aún siendo esta

respuesta ínfima, casi sin sentido, se extraería un coseno muy elevado, pudiéndose llegar a juzgar la respuesta del alumno cómo muy buena.

Una posible solución a este sesgo es emplear distancias euclídeas en lugar de los cosenos. Esta forma es tan sencilla cómo calcular la resta entre el vector que representa el texto ideal y el vector que del texto que se está juzgando. Olmos, León, Jorge-Botana y Escudero(2009) y Jorge-Botana, León, Olmos y Escudero(2010) ofrecen como resultado que las evaluaciones más parecidas que hace LSA a las que realizaron los jueces se encuentran más parejas cuando se usan las distancias euclídeas que cuando se utilizaron los cosenos. Es decir, utilizando las distancias las evaluaciones del LSA se parecen más a las de los jueces humanos. Los índices de similitud medidos con la distancia euclídea aúnan por un lado las longitudes de vector de ambos vectores además de respetar las distancias entre ambos. Esto hace que en este tipo de evaluaciones puedan ser una mejor alternativa a los índices basados en los cosenos.

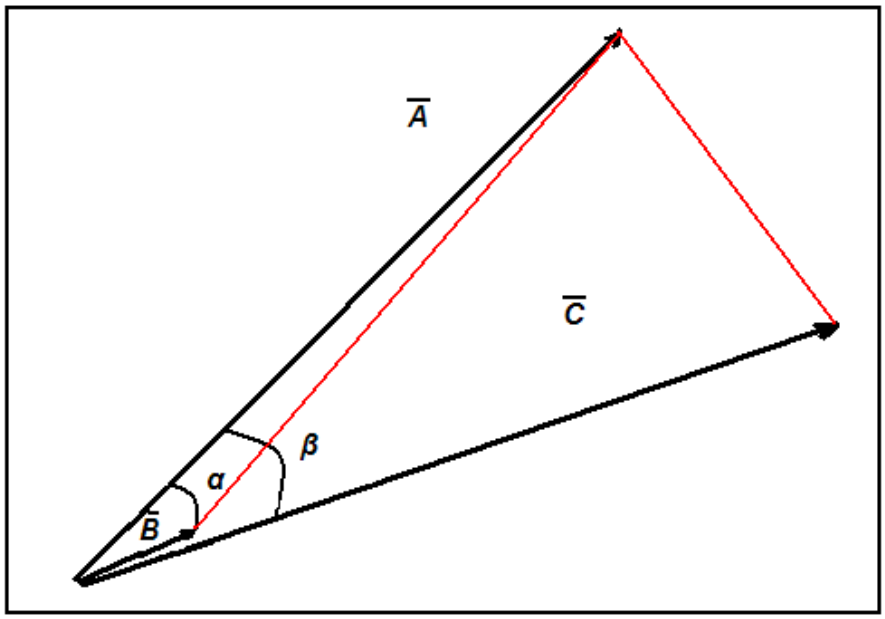


Figura 3.13.- **A** es el vector que representa la respuesta ideal y **B** y **C** representan dos posibles respuestas de alumnos. En el primer caso, el coseno del ángulo entre **A** y **B** (**α**), es muy elevado aunque el texto **B** sería juzgado como muy bueno. En cambio, el texto que representa **C** sería juzgado peor debido a que el coseno de **β** es menor. Si tenemos en cuenta que el texto de **C** es un texto mucho más equilibrado en cuanto al número de términos. ¿No sería el coseno una medida confusa?. Las distancias (en rojo) disiparían en parte este problema debido a que aúna longitud de vector y distancia entra vectores.

3.4.2.- El problema de la direccionalidad

Rehder et al. (1998) analizaron los aspectos técnicos de los autotutores implementados con LSA y hacen patente un problema que surge en este tipo de diseños. El coseno entre el ensayo de un alumno y el texto de referencia sólo dice cuan semejantes son los dos, pero nada dice sobre cuál de ellos ocupa un mayor nivel de conocimiento. En otras palabras, nada dice de si el grado de separación entre ambos textos significa que el alumno está en un nivel superior o inferior de conocimientos. Analizando los datos de la muestra de dos poblaciones (Figura 3.14 y 3.15), una por encima del texto de referencia (como son los estudiantes de medicina) y otra por debajo (como son los estudiantes de instituto), los autores descubrieron que la distribución de las distancias medidas como los cosenos entre los ensayos de cada una de estas muestras y el texto de referencia no es significativamente distinta. Sin embargo, la diferencia del rendimiento en unos cuestionarios sobre el tema es amplia y significativamente distinta posicionándose los estudiantes de medicina en un estadio bastante superior en cuanto a conocimientos. Esto muestra la naturaleza del problema de la direccionalidad: los cosenos por si solos no muestran la "dirección" en que las semejanzas se parecen, sino que simplemente nos dan una medida de semejanza cruda.

Una posible solución a este problema ha sido generar, dentro de un continuo de conocimiento, el nivel que un alumno puede poseer con ensayos nivelados por expertos. Conociendo la población a la que se destina la herramienta y teniendo textos para cada nivel marcados por expertos, el problema de la direccionalidad puede disiparse, pero aún así nunca estaremos seguros de su absoluta desaparición (Landauer, 2003; Jemni y Ben-Ali, 2004; Graesser, 2000). Aún así, estos autores proponen el uso de la técnica de escalamiento multidimensional para establecer diferencias entre el nivel de los ensayos. Proponen tres métodos en el que el último de ellos posee medidas de conocimiento externas a la propia técnica, lo que dificulta su implementación como herramienta "autotutor" aunque mejora su calidad.

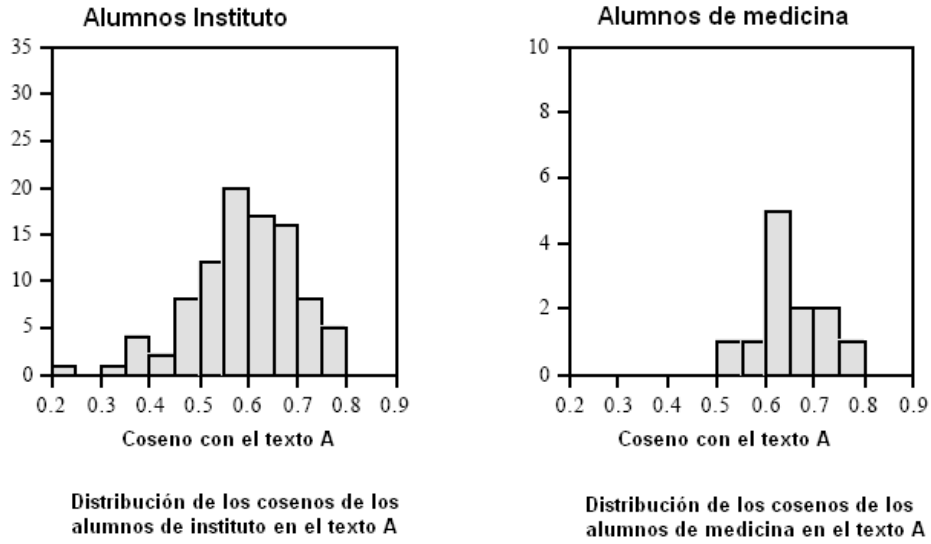


Figura 3.14.- Las distribuciones de los cosenos entre cada uno de los textos de los estudiantes de ambos grupos (instituto y medicina) y el texto A no muestra diferencias estadísticamente significativas. Tomado de Rehder et al. (1998)

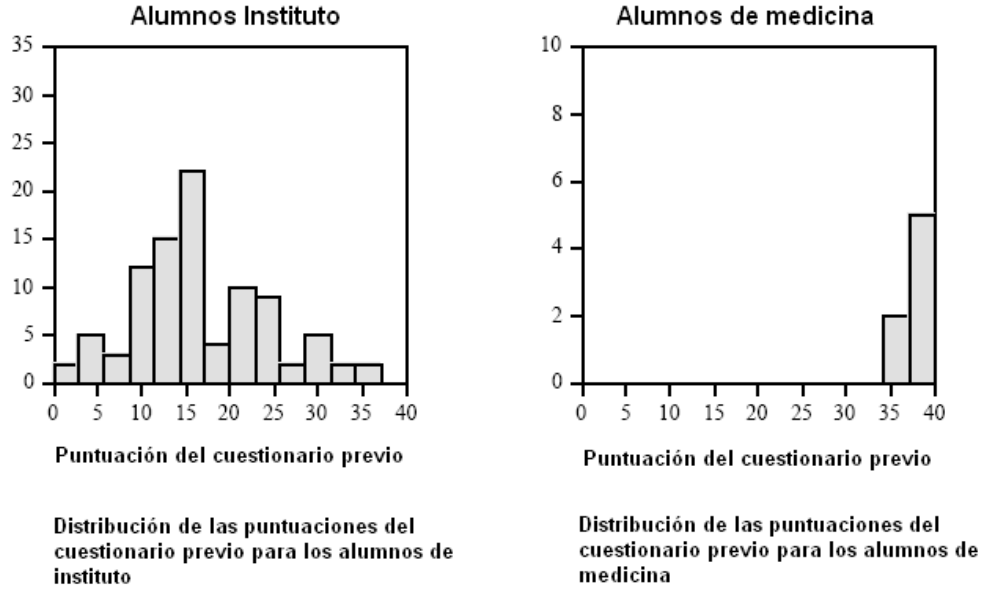


Figura 3.15. Las distribuciones de las puntuaciones en los cuestionarios previos de cada uno de los estudiantes de ambos grupos (instituto y medicina) muestran diferencias palpablemente significativas entre los grupos. Esta es la prueba que los estudiantes de medicina conocen más sobre el tema. Tomado de Rehder et al. (1998).

3.4.3.- Discriminación entre estrategias de elaboración

Cómo ya nos referimos en apartados anteriores, LSA es eficiente en discriminar las respuestas a exámenes según sea su nivel de conocimiento sobre una materia. Cosa distinta es que se ponga a prueba LSA para diferenciar, no ya si un ensayo maneja un determinado léxico, sino que tipo de estrategias de razonamiento se han llevado a cabo en su desarrollo. Existen algunas tentativas de estudio que han mostrado algunos resultados aunque aún no son del todo concluyentes.

Una de estas tentativas fue la llevada a cabo por Kurby et al. (2003), quienes investigaron la posibilidad de utilizar LSA un cómo índice de discriminación entre diferentes estrategias de elaboración sobre un tema concreto. Se trata de discriminar si dichas producciones son elaboraciones mínimas o locales (el alumno desarrolla paráfrasis e información leída en la frase anterior) o elaboraciones globales (conexión de lo leído con información que no se encuentra en las frases anteriores). Si LSA fuese sensible, las producciones basadas en estrategias locales tendrían que tener altos cosenos con textos que ejerciesen como referencia de este tipo de estrategia local y, po el contrario, bajos cosenos con textos que ejerciesen de referencia de una estrategia global. El sentido sería contrario con las producciones basadas en estrategias globales, las cuales tendrían altos cosenos con textos de referencia de estrategias globales y bajos con textos de referencia de estrategias locales. Tanto las producciones de los alumnos como los textos de referencia fueron evaluadas por expertos humanos. En suma, debe encontrarse un efecto de interacción entre las producciones de los alumnos y las referencias establecidas por expertos, dándose altos cosenos únicamente con la referencia que corresponde a cada uno de las estrategias (referencia local con estrategia local y referencia global con estrategia global). Los resultados no indicaron este efecto de interacción entre los tipos de referencias y los tipos de producciones de los alumnos. Estos autores concluyeron que LSA tiene dificultades en discriminar entre diversos tipos de elaboración de los textos (local, global). En esta misma línea, Wolfe y Goldman (2003) trataron de analizar la potencia de LSA para predecir la comprensión de un texto por parte de personas. Se

propusieron tres criterios para definir qué clase de explicaciones pueden encontrarse par inferir diferentes tipos de elaboraciones y se marcan manualmente las estrategias:

1) *Auto-explicaciones causales*: Se trata de explicaciones redundantes para con el texto como repeticiones excesivas de frases. No aporta elaboración de información sino parafraseado.
2) *Conexiones* de las frases con el mismo texto que se está leyendo: Las explicaciones se elaboran a partir del texto que se está leyendo o procesando. Aporta una elaboración media.
3) *Uniones* con información extraída de textos que han sido leídas previamente: Se trata de la elaboración máxima aportando no sólo información del mismo texto que se lee sino además, información de textos leídos previamente.

A cada grupo se le hizo leer dos textos con una perspectiva diferente sobre la caída del imperio romano (incluso con explicaciones contrarias). Los dos textos coincidían en lo que se refiere a la información general, pero no con las causas por las que los bárbaros invadieron el territorio. Una era porque era muy grande y otra porque los ciudadanos se volvieron vagos y ociosos. Bajo estos criterios, se supuso que con los dos textos, los sujetos que integrasen la información, deberían extrapolar la información de uno e integrarlo en el otro.

Un primer resultado indicó que los cosenos medios correlacionan con altos índices de razonamiento. La esencia del experimento se basaba en que las medidas de alta similitud que establece LSA entre los textos de referencia y los protocolos de respuesta de los alumnos pueden no estar indicando la verdadera integración de información de conocimiento, sino un simple parafraseado. Como consecuencia de este parafraseado, la similitud es muy alta. Lo que proponen estos autores es que los índices que muestran la verdadera integración de la información se encuentran en los lugares representados por cosenos medios, es decir, sin llegar al punto de la ausencia de relación pero tampoco con relaciones extremas. Es decir, la relación entre los índices de la elaboración de conocimiento medida con los criterios antes mencionados y las medidas de los cosenos en LSA no es lineal. Los resultados

ratificaron estos supuestos, pues la relación existente entre estos dos índices es curvilínea. Los niveles altos de razonamiento coinciden con las partes intermedias de las medidas de similitud (cosenos) en LSA. Otra conclusión a los que llegaron estos autores es que un texto del alumno puede tener un coseno alto con un texto de una interpretación y también tenerlo alto con un texto que refleja otra interpretación distinta o casi contraria. Esto puede ser índice de elaboración y por tanto de riqueza de razonamiento.

Wolfe y Goldman (2003) también calcularon el coseno entre el texto producido por el alumno y cada uno de los textos que los autores habían considerado como contrarios o como diferentes interpretaciones (provenientes de cada uno de los dos textos de las causa de la invasión bárbara). Promediaron estos dos cosenos y los compararon con el criterio de "integración del conocimiento" extraído de la evaluación de los expertos sobre los razonamientos realizados por los alumnos. Los resultados fueron concluyentes. Altos índices de elaboración de conocimiento coincidieron con un alto promedio entre ambos cosenos. Esto quiere decir que los alumnos que identificaron los textos contrarios como relacionados fueron los que obtuvieron una mejor puntuación en la "elaboración del conocimiento". Son alumnos que habrían conseguido integrar los dos textos. Por tanto, esta puede ser una medida de integración de la información de ambos textos, ya que el texto producido por el alumno contendrá información de ambos.

3.4.4.- Tipos de relación (meronimia, partonimia, hiponimia)

3.4.4.1.- LSA vs Ontologías

En los modelos LSA no se marca explícitamente la relación semántica entre sus constituyentes. LSA no se basa en diccionarios, ni en reglas semánticas, ni en gramáticas, ni en ontologías previamente diseñadas. La única representación que encontramos en los modelos LSA es la representación vectorial de cada una de sus unidades. A diferencia de otras aportaciones en dónde se hacen explícitas algunos tipos de relaciones cómo, por ejemplo, la sinonimia, la partonimia, la hiponimia o la meronimia,

LSA no hace explícita ninguna relación explícita, sino que interpreta las similitudes en base a operaciones con los vectores sin extraer más datos que su proximidad semántica. Estas características otorgan a estos modelos algunas ventajas, pero también algunos inconvenientes. Recogemos, a continuación, un análisis minucioso de las ventajas del modelo LSA frente a las ontologías basadas en algunos trabajos previos (Rung-Ching Chen, Ya-Ching Lee y Ren-Hao Pan, 2006; Kaur y Hornof, 2005; Dumais, 2003). Destacamos las siguientes:

- En los modelos LSA, la métrica está claramente especificada. Los vectores que representan las unidades dentro de un espacio semántico (términos, párrafos, documentos, etc.) son susceptibles de mediciones y comparaciones mediante álgebra lineal: productos escalares, cosenos, distancias euclidianas, longitud de vector, etc. Los índices extraídos de las operaciones entre los vectores son objetivables y fácilmente entendibles. Las ontologías no tienen medidas objetivables[4].

- Las ontologías no son fácilmente escalables para nuevos términos y nuevos dominios. Esto es un inconveniente a la vista de la velocidad de los cambios a los que están sometidos los medios en la red.

- Mientras las ontologías provienen de marcado lógico de sus unidades, los modelos LSA están basados en ocurrencias reales del lenguaje y, por tanto, gozan de plausibilidad psicológica (*un catálogo no es una teoría del significado*, Kintsch, 2001). Una muestra real de ocurrencias de lenguaje es en definitiva una muestra de la organización que ha tenido la información a la hora de introducirse en el sistema cognitivo.

Aún así, una gran ventaja de las ontologías frente a los modelos LSA es la que proviene, precisamente, de la explicitación del tipo de relación que se establece entre las unidades. Dados dos términos, A y B, mediante las técnicas basadas en ontologías y recorriendo la estructura de los datos, podemos saber, por ejemplo, que un brazo no es un tipo de cuerpo (relación de hiponimia), sino una parte del cuerpo (relación meronimia), o que una ballena es un ejemplar de animal mamífero y cetáceo (relación de hiponimia). Es decir, mientras que con los métodos LSA se extraería un índice alto en cuanto a la similitud de ambos términos, con las técnicas basadas en ontologías careceríamos, quizás, de un

[4] Esto no es del todo cierto toda vez que se han hecho algunas aproximaciones. Por ejemplo, Pedersen, Patwardhan y Michelizzi,(2004) han desarrollado una herramienta llamada "*WordNet:Similarity*" en donde emplean una métrica objetivable cómo basada en índices a partir de las propias definiciones del diccionario y de las distancias entre los nodos.

índice fiable de similitud, pero nos proveería de un dato que podría ser capital para la implementación de algunas utilidades, a saber, el tipo de relación que se mantienen los términos.

3.4.4.2.- ¿Por qué las ontologías no parecen modelos mentales?

Independientemente de la carencia de objetivación de las medidas de semejanza, uno de los problemas que poseen las ontologías es que son una representación discreta e idealizada de la realidad. Una ballena es un mamífero, lo que la haría funcionalmente igual a una vaca. Nadie iría desencaminado al decir que vaca y ballena tienen semejanzas, pero los datos empíricos muestran como hay ejemplares de las categorías que son más característicos que otros. Seguramente, en este caso la similitud percibida entre vaca y ballena sea menor que entre vaca y canguro siendo, en todos los casos, mamíferos. Es obvio que las ontologías tienen descripciones mucho más elaboradas que éstas, pero también es cierto que, aún así, siempre tienen la desventaja de que no parten del análisis de descripciones de ocurrencias o contingencias reales (como el lenguaje natural) y, por tanto, el ajuste entre los juicios que parten de los sistemas que las usan y los modelos humanos no es el óptimo. La causa de ello se encuentra en que las ocurrencias del lenguaje natural son una muestra de cómo la información es presentada al sistema cognitivo humano, es decir, de la materia prima real con la que cuenta el sistema cognitivo para realizar sus elaboraciones, cosa que no se cumple en las definiciones taxonómicas. Por muy elaboradas que estén estas taxonomías en cuanto a su ajuste con el medio real, el modelo real humano introduce sesgos a la categorización del conocimiento que son consecuencia directa de la forma en que la información es presentada al sistema cognitivo y la forma en que éste aprende, lo que propicia la formación de los llamados *modelos mentales* (Doyle, 1998). También recuerdan a las críticas que se vertían sobre la teoría clásica de la formación de conceptos, que son afirmaciones que pueden decirse también de las ontologías (véase para una revisión Sainz, 1991). Esta teoría clásica se apoyaba en los postulados (estos postulados son generalmente compartidos con las ontologías a las cuales se parecen en diseño) siguientes:

A) Los conceptos son mentalmente representados como combinaciones de propiedades necesarias y suficientes que definen la categoría que representan.

B) Las propiedades de la categoría se aplican igualmente y en la misma medida, a todos los miembros de la categoría.

C) Las categorías se organizan de forma jerárquica, de tal forma que los miembros de una determinada categoría están representados como una combinación de propiedades heredadas de los miembros de las jerarquías superiores más las propiedades definitorias que caracterizan al nivel al que pertenecen. Todos los conceptos son representados de esta manera.

D) Todas las personas representan los conceptos de la misma manera.

Como corolario a estas cuatro se puede extraer que:

> *"La hipótesis crítica del modelo clásico se basa en que existe alguna descripción sumaria del concepto natural susceptible de expresarse en términos de una regla formal"* (Sainz, 1991).

Al modelo clásico se le adjudicaron algunas críticas provenientes del campo empírico en el que se encontraban ciertos fenómenos que estos modelos no podían explicar. Estos efectos se recogen en Sainz (1991) de los que destacamos los siguientes:

A) *Indefinición de rasgos nucleares*: No existe una descripción sumaria de los conceptos. Se constata imposibilidad práctica de la determinación de los rasgos definitorios que comparten todos los miembros de una misma categoría (Medin, 1989).

B) *Indefinición de límites*: Imposibilidad de determinar a priori la extensión de un concepto dada su intensión. Es decir, no están claros los límites de las categorías y una persona puede asignar un objeto a dos categorías distintas dependiendo del contexto de categorización.

C) *Indefinición jerárquica*: De acuerdo con la lógica formal, los conceptos se organizan en taxonomías y partonomías. De acuerdo también con esta lógica los niveles inferiores comparten con los de niveles superiores todos sus predicados. se considera que existe una relación interna jerárquica entre los miembros de la categoría, debido a que en estos modelos la relación entre los miembros de la categoría es transitiva, es decir, que si A es B y B es C, entonces A es C, heredando los miembros inferiores en la jerarquía las definiciones de los miembros de las superiores. Resulta sin embargo que esta relación de transitividad no se cumple en los juicios de categorización humanos. "Aunque el Big Ben sea un reloj, y un reloj sea un mueble, de ahí no se sigue que el Big Ben sea considerado como un mueble"

D) *Indefinición de membrecía*: Los miembros de un concepto no son igualmente representativos del concepto al que pertenecen. En este sentido, se produce el llamado efecto de tipicidad o prototipo en los juicios humanos y es que algunos miembros, quizás por una cuestión probabilística, resultan mejores representantes de la categoría que otros. Y no sólo eso sino que este grado de representatividad fluctúa en base al contexto de categorización.

Si se analizan meticulosamente cada uno de los fenómenos empíricos, llegamos a la conclusión que un modelo mental basado en ontologías sería un mal modelo. Las ontologías son inflexibles en cuanto a este tipo de experiencias y serían malas consejeras a la hora de medir la similitud percibida por un modelo en cuanto a su ajuste con el humano real. Por otra parte, los modelos LSA tienen la ventaja de extraer su análisis estadístico de muestras de lenguaje natural qué es un reflejo de la manera en que la información se organiza y se nos presenta. Por esta causa, los modelos LSA son más aptos para reflejar los sesgos que pueden introducirse en la organización del conocimiento. De la misma manera, la representación vectorial de las unidades en LSA y su naturaleza continua y no discreta, puede ser una ventaja para emular los efectos que se producen en los juicios humanos, a saber: Indefinición de rasgos nucleares, indefinición de límites, indefinición jerárquica o la indefinición de membrecía. La representación vectorial es mucho más flexible a la hora de emular efectos propiciados por el contexto que es en definitiva la parte central de la simulación del procesamiento del lenguaje.

3.4.4.3.- ¿Es LSA sensible a todos los fenómenos empíricos de categorización?

El que la arquitectura funcional de LSA sea más apta para dar cuenta de efectos que se producen en el ámbito de la categorización no quiere decir que sea un modelo totalmente correcto. No son muchos hasta ahora los estudios que se han ocupado de ello. Una de las investigaciones de las que se puede extraer datos sobre LSA y categorización es la realizada por Schunn (1999). Este autor reflexiona sobre la capacidad de LSA para dar cuenta de algunos fenómenos que se producen en la categorización humana cómo son los efectos de *tipicidad* y de *centralidad*. La tipicidad hace referencia a que el razonamiento

humano es sensible a que un objeto puede tener características más frecuentes que otras como, por ejemplo, la pregunta, ¿con que frecuencia tienen los pájaros alas? En la definición y categorización de "pájaro", la característica de tener alas juega un papel u otro según sea su distribución entre los ejemplares. El segundo efecto, el de centralidad, es de mayor complejidad, ya que apela a la cuestión no ya de cuántas veces se identifica la propiedad en los ejemplares, sino cómo de importante es esa propiedad para definir para incluirlos dentro de la categoría. En el ejemplo anterior, si nos presentan un ejemplar "sin alas", ¿cuán "pájaro sería?, ¿qué probabilidades tendría de incluirse en la categoría de pájaro? Si la propiedad "tener alas" fuera de suma importancia, el ejemplar a clasificar no llegaría a clasificarse cómo pájaro si no tuviese alas. Dentro del concepto de centralidad se pueden hacer varias observaciones (véase Shunn y Vera, 1995). Primero, que la tipicidad es un predictor de la centralidad. Las características centrales suelen ser también típicas. Segundo, que además de la tipicidad, hay algunos predictores más de centralidad. El más importante es el de causalidad. Las características que tienen que ver con una función importante del objeto en cuestión, predicen más la centralidad. En nuestro ejemplo, "tener alas" para un pájaro puede que promueva que esa propiedad sea central.

Precisamente, el estudio de Schunn (1999) consistió en contrastar los juicios humanos referentes a propiedades de objetos y compararlos con los juicios que hace LSA de la proximidad de estos mismas características a los objetos. Referente a los juicios humanos, diseñó tipos de preguntas con las que indagar cinco propiedades: tipicidad (¿cuántos objetos de esta categoría tienen esta propiedad?), centralidad (¿cómo cambiaría la categorización de un objeto si no tuviese esta propiedad?), causalidad (¿es importante esta propiedad para el éxito de este objeto?), definición (cómo es esa propiedad de importante para la definición científica del objeto) y reconocimiento (¿cuán importante es la propiedad para reconocer el objeto?). Los resultados mostraron que LSA predijo bastante bien la tipicidad en las características en los objetos. Aún así, sólo predice la centralidad en sólo algunos objetos, extrayéndose un promedio sea bastante humilde. El autor indagó sobre la posibilidad de que las propiedades que LSA predijesen bien la centralidad, ya fuesen éstas fruto del

efecto de tipicidad, es decir, que la buena predicción sobre centralidad se debiese a que esa propiedad sea también bastante típica. Con este objetivo analizó las correlaciones obtenidas de puntuaciones de tipicidad, causalidad, reconocimiento, definición y las comparó con la centralidad. En otras palabras, trató de analizar la correlación entre los juicios humanos y de LSA de esas cuatro variables respecto al juicio de centralidad.

Los resultados fueron concluyentes. El éxito de LSA en predecir la centralidad de una determinada propiedad se debe tan sólo a que esa propiedad sea típica y no a sus relaciones con otras características. LSA no puede predecir centralidad más allá de la tipicidad. Cuando la categoría tiene una estructura en la que la tipicidad, la causalidad o la definición no correlacionan entre ellas, LSA no predice centralidad. Esto muestra que, al menos, la versión clásica de LSA, tiene problemas para dar cuenta de todos y cada uno de los fenómenos, aunque tenga éxito en predecir la tipicidad.

3.4.4.4.- Orientaciones híbridas LSA/Ontologías

Fruto de las ventajas y desventajas de cada orientación, se han implementado sistemas que hacen uso de las dos técnicas, unas veces LSA ejerce de complemento a un sistema basado en ontologías y, otras, son las representaciones de esas ontologías las que ayudan a crear espacios semánticos en LSA. A la primera aproximación pertenecen los trabajos de Cederberg y Widdows (2003). Estos autores aumentaron la eficiencia de un sistema de extracción de hipónimos basado en reglas. Sobre los resultados de dicho sistema se aplicaron índices extraídos de LSA para imponer así un doble filtro basado en criterios más cercanos a los de los humanos. Los resultados revelaron que, combinando las dos técnicas, aumentan los índices de precisión y "*recall*". También Ozcan y Aslandogan (2005) llevaron a cabo un sistema que contaba con la aportación de ambas técnicas. Por un lado, se empleaba WordNet 2.0, de donde se extrajeron relaciones de sinonimia y meronimia y, por otro, LSA con la identificación de conceptos. Los resultados confirmaron de nuevo un aumento de la eficiencia.

Otra mejora de un sistema basado en ontologías por parte de LSA es el concerniente a que las ontologías no son fácilmente escalables para nuevos términos y nuevos dominios. Rung-Ching Chen, Ya-Ching Lee y Ren-Hao Pan 2006) mejoraronn ontologías previamente diseñadas con la introducción de algunos términos nuevos. Mediante índices LSA se buscaron términos que tuviesen una alta similitud con los términos de la ontología. Según los autores se trata de un método de bajo costo y muy efectivo. Por otro lado, existen también aproximaciones que se han servido de ontologías para crear espacios semánticos en LSA. Burek, et al (2004) aprovecharon ontologías creadas previamente para construir documentos que se introdujeron en la matriz LSA. El procedimiento fue como sigue: a), los documentos se crean a partir de cada una de las clases que forman la ontología de un dominio específico; b), cada uno de los documentos contiene la información de cada clase (nombre, propiedades, relaciones). El documento o la columna de la matriz está representado por cada una de las clases de la ontología y cada fila de la matriz, como en otras ocasiones, representa los términos únicos que aparecen en las clases. Y c), Cuando se le aplica a la matriz de ocurrencias los cálculos de entropía, los términos que dan nombre a las clases son multiplicados por 2.

Esta forma de crear espacios semánticos puede resultar de suma utilidad para localizar segmentos relevantes que ejerzan de documentos en el análisis LSA. También puede emplearse en la obtención de documentos de referencia en las aplicaciones de evaluación automática de repuestas en el ámbito académico. Un documento que ha sido creado a partir de una clase puede emplearse cómo documento ideal. Este enfoque puede resultar un puente entre las ontologías y el LSA ya que se hace continua una descripción que es discreta cómo las ontologías, pero resta validez ecológica al LSA pues las ontologías adolecen de verosimilitud con el criterio de categorización humana al no ser técnicas basadas en análisis de corpus sino en criterios de expertos. Esta manera de diseñar matrices puede ser de suma utilidad cuando contemos con restricciones de cálculo en las propias máquinas.

Capítulo 4
Aplicaciones

4.- Aplicaciones

4.1.- Medidas de cohesión y coherencia textual

Tanto desde la lingüística, psicolingüística como de la psicología del discurso se asume que, tradicionalmente, existen al menos dos formas de evaluar la fluidez y semántica de un texto. Un primer nivel más superficial se ha denominado *cohesión textual* y un segundo nivel, más profundo, se identifica con la *coherencia* y con la continuidad del sentido. La razón principal por la que se considere a la cohesión como un nivel más superficial es porque ésta no trasciende del nivel lingüístico de análisis, lo que permite ser más cuantificable. Por otra parte, la coherencia se diferencia de la cohesión en que aquella no está especialmente centrada en el lenguaje, sino más bien en procesos más globales y de más alto nivel del discurso, como es la intencionalidad de los emisores (Kintsch, 1990)**.** Sin embargo, a pesar de ser conceptos diferentes ambos están íntimamente relacionados. Tal es así que para dotar de mayor coherencia a un texto se debe garantizar su cohesión.

Dentro de la coherencia suele distinguirse, a su vez, entre *coherencia global*, *coherencia lineal* y *coherencia local*. Mientras que la coherencia global tiene que ver con la unidad temática del texto la coherencia lineal, por su parte, se suscribe con la estructura del texto y con la organización lógica de las ideas, es decir, se preocupa de que las distintas partes del texto mantengan relaciones de significado y también de asegurar una adecuada progresión temática. Finalmente, la coherencia local se encarga de aspectos menos moleculares y no menos importantes como es dotar del sentido más lógico a cada enunciado. Por todo lo expuesto hasta ahora, la coherencia textual no puede interpretarse en términos absolutos, coherencia máxima o mínima, sino en términos relativos, puesto que existe toda una escala de coherencia que se distribuye entre los dos polos siendo esta de naturaleza parcial.

Debido a la laxitud operativa de la definición de coherencia y, proponiendo que gran parte de la coherencia queda garantizada con el mantenimiento de la cohesión, uno de los métodos para medir la coherencia es extraer índices objetivos de cohesión. Tanto es así, que para algunos autores consideran la cohesión cómo la expresión lingüística de la coherencia, pudiendo ser esta cohesión de carácter formal (aspectos cómo la conjugación) como semántica (sucesión lógica de contenidos entre las partes). Esta cohesión semántica y dependiendo de los mecanismos semánticos involucrados puede subdividirse en cohesión gramatical (afecta a la referencia, a la sustitución o a la elipsis) o cohesión léxica (afecta al léxico). Es, precisamente, en esta cohesión léxica donde opera LSA.

Tradicionalmente, se define cohesión léxica cómo la ocurrencia de repetición de palabras, de sinónimos, de antónimos, etc., que marcan las relaciones semánticas entre palabras y sirven para enlazar y relacionar los distintos enunciados y secuencias de enunciados que conforman el texto, evitando así su fragmentación. Uno de los intentos en los que se ha tratado de medir, entre otras cosas, la cohesión (mediante la correferencia entre unos textos y otros) ha sido la aplicación llamada *Coh-Metrix,* (Graesser, McNamara, Louwerse y Cai, 2004). Para comprobar la efectividad de LSA en este tipo de medidas, estos autores calcularon previamente y de manera automática la forma en que dentro de unas frases se hacía mención a términos y raíces morfológicas de otras. De esta manera, podría conocerse que frases hacen correferencia a otras. Supusieron que una matriz contuviese el valor de correferencia entre las frases, de manera que en ambos ejes estuviesen ordenadas las mismas frases con una diagonal con correferencia 1. Así, podría calcularse diversos índices sobre los que se pudiese precisar cuan cohesionados están los textos. Además, se calculaba una ponderación que imponía un orden de importancia a dicha correferencia, teniendo en cuenta la distancia entre dichas frases, dando una mayor cohesión a frases contiguas. Partiendo de estas medidas se propusieron dos índices: A) *Cohesión por correferencia local*, que indica cómo las frases adyacentes tienen una cohesión por correferencia; y B), *Cohesión por*

correferencia global, que incluye la correferencia de todas las frases de un texto.

Extraídos estos índices, la cohesión calculada por LSA es similar a la expresada por estas fórmulas, pero empleando en vez del valor de la correferencia, el valor del coseno entre las frases. En otras palabras, la matriz estará compuesta por valores cosenos que representan la relación semántica de unas frases con otras y no la correferencia de algunos términos o raíces. De esta forma, se mide la manera en que unas hacen mención a otras semánticamente, lo cual se muestra más efectivo ya que además de permitir la medición de la correferencia semántica, incluye también las anteriores medidas como se muestra en la figura 4.1.

Además de la cohesión entre las unidades y solapando su definición, se puede reconocer en los textos una cierta estructura jerárquica en cuanto a sus contenidos. El lector de un texto pone en marcha una serie de estrategias organizativas para capturar esta estructura y organizar el conocimiento que se puede extraer del texto en unidades moleculares más amplias de significado llamadas macroproposiciones. En realidad, cualquier puede componerse de varias macroproposiciones. Autores como van Dijk(1972) ya propusieron que de todo un conjunto de microproposiciones (unidades mínimas de significado) se puede extraer las macroproposiciones que las representen. Esta organización de macro y microproposiciones estará distribuida de manera jerárquica. Para llevar a cabo estas operaciones, van Dijk & Kintsch (1983) propusieron que el lector contaba con unas reglas u operadores que hacían posible estas transformaciones. Estas reglas se denominan reglas de formación de macroproposiciones o macrorreglas y son tres, a saber: la macrorregla de *eliminación* (prescindir de microproposiciones consideradas irrelevantes); la macrorregla de *generalización* (sustituir un concepto general en lugar de una sucesión de especificidades); y la macrorregla de *construcción* (sustituir un evento general en el lugar de una sucesión de eventos). En el plano textual, una macroproposición puede definir un párrafo, un apartado, un bloque, e incluso al texto completo.

Description	Measure	Text 1	Full description
Title	Title	pueba	Title
Genre	Genre	Science	Genre
Source	Source	In cohmatrix home	Source
JobCode	JobCode	chavo	JobCode
LSASpace	LSASpace	Encyclopedia	LSASpace
Date	Date	38.922	Date
Adjacent anaphor reference	CREFP1u	0.3	Anaphor reference, adjacent, unweighted
Anaphor reference	CREFPau	0.116	Anaphor reference, all distances, unweighted
Adjacent argument overlap	CREFA1u	0.65	Argument Overlap, adjacent, unweighted
Argument overlap	CREFAau	0.377	Argument Overlap, all distances, unweighted
Adjacent stem overlap	CREFS1u	0.575	Stem Overlap, adjacent, unweighted
Stem overlap	CREFSau	0.352	Stem Overlap, all distances, unweighted
Content word overlap	CREFC1u	0.129	Proportion of content words that overlap between adjacent sentences
LSA sentence adjacent	LSAassa	0.316	LSA, Sentence to Sentence, adjacent, mean
LSA sentence all	LSApssa	0.251	LSA, sentences, all combinations, mean
LSA paragraph	LSAppa	0.396	LSA, Paragraph to Paragraph, mean
Personal pronouns	DENPRPi	39.29	Personal pronoun incidence score
Pronoun ratio	DENSPR2	0.14	Ratio of pronouns to noun phrases
Type-token ratio	TYPTOKc	0.683	Type-token ratio for all content words
Causal content	CAUSVP	15.209	Incidence of causal verbs, links, and particles
Causal cohesion	CAUSC	0.182	Ratio of causal particles to causal verbs (cp divided by cv+1)
Intentional content	INTEi	12.674	Incidence of intentional actions, events, and particles.
Intentional cohesion	INTEC	0.182	Ratio of intentional particles to intentional content
Syntactic structure similarity adjacent	STRUTa	0.094	Sentence syntax similarity, adjacent
Syntactic structure similarity all-1	STRUTt	0.097	Sentence syntax similarity, all, across paragraphs
Syntactic structure similarity all 2	STRUTp	0.107	Sentence syntax similarity, sentence all, within paragraphs
Temporal cohesion	TEMPta	0.925	Mean of tense and aspect repetition scores
Spatial cohesion	SPATC	0.443	Mean of location and motion ratio scores.
All connectives	CONi	2.535	Incidence of all connectives
Conditional operators	DENCONDi	0	Number of conditional expressions, incidence score
Pos. additive connectives	CONADpi	1.267	Incidence of positive additive connectives
Pos. temporal connectives	CONTPpi	3.802	Incidence of positive temporal connectives
Pos. causal connectives	CONCSpi	1.267	Incidence of positive causal connectives
Pos. logical connectives	CONLGpi	1.267	Incidence of positive logical connectives
Neg. additive connectives	CONADni	6.337	Incidence of negative additive connectives
Neg. temporal connectives	CONTPni	2.535	Incidence of negative temporal connectives
Neg. causal connectives	CONCSni	1.267	Incidence of negative causal connectives
Neg.logical connectives	CONLGni	6.337	Incidence of negative logical connectives
Logic operators	DENLOGi	46.895	Logical operator incidence score (and + if + or + cond + neg)
Raw freq. content words	FRQCRacw	1.661.935	Celex, raw, mean for content words (0-1,000,000)
Log freq. content words	FRQCLacw	2.094	Celex, logarithm, mean for content words (0-6)
Min. raw freq. content words	FRQCRmcs	20.5	Celex, raw, minimum in sentence for content words (0-1,000,000)
Log min. freq. content words	FRQCLmcs	0.998	Celex, logarithm, minimum in sentence for content words (0-6)
Concreteness content words	WORDCacw	390.426	Concreteness, mean for content words
Min. concreteness content words	WORDCmcs	262	Concreteness, minimum in sentence for content words
Noun hypernym	HYNOUNaw	5.034	Mean hypernym values of nouns
Verb hypernym	HYVERBaw	1.527	Mean hypernym values of verbs
Negations	DENNEGi	7.605	Number of negations, incidence score
NP incidence	DENSNP	280.101	Noun Phrase Incidence Score (per thousand words)
Modifiers per NP	SYNNP	0.887	Mean number of modifiers per noun-phrase
Higher level constituents	SYNHw	0.722	Mean number of higher level constituents per word
Words before main verb	SYNLE	4.488	Mean number of words before the main verb of main clause in sentences
No. of words	READNW	789	Number of Words
No. of sentences	READNS	41	Number of Sentences
No. of paragraphs	READNP	8	Number of Paragraphs
Syllables per word	READASW	1.621	Average Syllables per Word
Words per sentence	READASL	19.244	Average Words per Sentence
Sentences per paragraph	READAPL	5.125	Average Sentences per Paragraph
Flesch Reading Ease	READFRE	50.166	Flesch Reading Ease Score (0-100)
Flesch-Kincaid	READFKGL	11.043	Flesch-Kincaid Grade Level (0-12)

Figura 4.1.- Ejemplo de puntuaciones extraídas en una prueba llevada a cabo por nosotros, préstese atención a las medidas basadas en LSA marcadas en amarillo. La aplicación completa se encuentra en la dirección: http://cohmetrix.memphis.edu/cohmetrixpr/metgit.html

Posteriormente, Kintsch (2002) propuso que si LSA simula bien el conocimiento humano y con dicho modelo podríamos representar vectorialmente palabras, frases, párrafos y textos, también podríamos comprobar que es lo que sabe el lector de ese texto y cuan conveniente para él

es la estructura jerárquica propuesta. De esta forma, se abría la posibilidad de estructurar un texto de manera que los títulos de los apartados representasen a dichos apartados y que ambas estructuras, a su vez, quedasen representadas bajo un único tema o tópico. En otras palabras, podemos agrupar conjuntos de microproposiciones en unidades mayores formando macroproposiciones o, simplemente, comprobar la bondad de ajuste entre las existentes. Aún así, el mismo autor advierte de la parcialidad de LSA para analizar las reglas de formación de macroproposiciones, tanto en cuanto se necesita un análisis sobre las frases y de su estructura sintáctica. Kintsch (2002) propone los índices de utilidad:

> 1) El coseno entre el párrafo y el subtítulo propuesto es una medida de cómo el subtítulo representa al párrafo cómo un todo.
>
> 2) De la manera en que esté relaciona una sección del texto con las restantes puede ayudar a reorganizarlas dentro del texto general.
>
> 3) La importancia que se otorgue a una determinada parte del texto puede medirse mediante el coseno del texto completo y de cada una de las secciones que lo constituyen, aunque se nos advierte de la posibilidad de que el tamaño de las secciones influya sobre tal índice. Una alternativa a ésta es medir la fuerza de la relación de la sección seleccionada con todas las demás. Esto se hace calculando el promedio de los cosenos de cada sección y todas las demás. De esta manera podemos localizar la sección más prototípica del texto.
>
> 4) Las mismas medidas se pueden extraer de las frases de dichas secciones. Por ejemplo, podemos obtener la frase más prototípica de una sección (esta será la que tenga un coseno promedio mayor de las relaciones entre cada frase y todas las demás).

4.2.- Marcadores lingüísticos de cambio psicológico y salud general

Este análisis del discurso ha tenido y tiene importantes aplicaciones. Una de ellas se ha dirigido hacia el ámbito de la psicología clínica, para evaluar el estado anímico de las personas y el riesgo de sufrir determinados trastornos psicológicos. Precisamente, una de las formas de detectar estos patrones poco adaptativos ha sido analizar el tipo de discurso que las personas realizan para describir un evento. Así, por ejemplo, las llamadas técnicas proyectivas cómo el

Test de Apercepción Temática TAT (Murray, 1943) han utilizado esta fórmula, en la que el especialista clínico insta al evaluado a describir un determinado evento y analiza el tipo de discurso que el paciente despliega.

El análisis del discurso puede llevarse de dos maneras, bien teniendo en cuenta los fenómenos léxicos y de contenido, o bien, teniendo presente los fenómenos morfológicos. En ambos casos se ha intentado aplicar LSA. Respecto a la primera aproximación que se concentraba en los fenómenos léxicos y de contenido, Chambers, Tetreault y Allen (2004), llevaron a cabo la técnica LSA sobre dos corpus que provenían del ámbito psiquiátrico. Ambos corpus estaban marcados en cuanto a su contenido con diferentes reseñas que indicaban las actitudes y el compromiso que las personas comentaron en cada uno de los párrafos. Al igual que otros estudios (Chu-Carroll y Carpenter, 1999), obtuvieron mejores resultados cuando el texto era procesado en forma de N-gramas, es decir, cuando los términos se agruparon formando unidades mayores. Cómo particularidad de este diseño, merece la pena reseñar que se realizaron algunas modificaciones respecto a la forma estándar de purgar de palabras en LSA. En este caso, los autores conservaron algunas palabras indicadoras de indicios de actitudes y compromisos cómo interjecciones tales como “hmmm”, “uh”.

Respecto a la segunda aproximación, la que se concentraba en el estudio de fenómenos morfológicos, Campbell y Pennebaker (2003) pusieron a prueba la técnica de LSA para medir posibles predictores de cambio psicológico y su relación con índices de salud física. Anteriores resultados habían concluido la importancia de las llamadas palabras de función en relación con los estados psicológicos y salud en general cómo, por ejemplo, pronombres y preposiciones (Pennebaker, Francis y Booth, 2001). Campbell y Pennebaker (2003) llevaron a cabo tres estudios con corpus previamente procesados. En el primer estudio contaba con tres grupos experimentales. Al primero de ellos se les pidió que escribiesen sobre emociones experimentadas en la universidad, mientras que al segundo grupo experimental se les pidió que relatasen un acontecimiento traumático y, finalmente, al tercero, pacientes internos de un psiquiátrico, narraban acontecimientos traumáticos. Cada uno

de los tres estudios incluía también sendos grupos control en el que se les pidió que escribiesen sobre temas que carecían de emocionalidad. Todos los estudios se correlacionaron con índices de salud, tales como la frecuencia de consulta en los centros de salud. Una vez cumplimentada esta primera parte, se llevó a cabo la aplicación con LSA. Para ellos se extrajeron tres tipos de espacios semánticos: de contenido, el cual indexa palabras de contenido cómo sustantivos y verbos, de estilo, el cual contiene las unidades que no son de contenido y de partículas, el cual tiene sólo pronombres y preposiciones. Cabe señalar que para el espacio semántico de contenido se emplearon 276 dimensiones y que para los de estilo y partículas se emplean 26 dimensiones, haciendo gráfica las diferencias entre las dos formas de hacer espacios, ya que el de contenido soporta, por su naturaleza semántica, muchas más dimensiones que el de estilo. La razón fundamental de estas diferencias en las dimensiones se debe a que no hay tantos aspectos diferenciables en estilos cómo en contenidos. Resulta paradójico señalar también que en una técnica creada con el fin de aislar conglomerados semánticos se diseñen corpus dónde sus unidades carezcan de semántica, pero los autores argumentaron que unidades cómo preposiciones y pronombres señalan una actitud del narrador ante los acontecimientos, lo que entraña algún significado. Los resultados fueron concluyentes. En el espacio semántico de contenido no se detectó significación alguna en la correlación entre escribir de diferentes contenidos y los índices de salud o, lo que es lo mismo, no hay suficientes datos en el contenido de un texto para distinguir su riesgo de enfermedad. Respecto a las unidades sin contenido, se produjo una correlación positiva entre los estilos (emplear de una forma u otra las unidades que no son de contenido) y los índice de salud. Igualmente se detectó una correlación positiva en el tercer espacio compuesto sólo con pronombres y preposiciones, con lo que los autores concluyeron que los pronombres y preposiciones fueron los que mejor predijeron la variación de los índices de salud. Además, el estudio también revelo que aquellos participantes que en su narración cambiaban su estilo (línea temporal), eran también quienes más variaban sus índices de afluencia a los centros de salud. Estos resultados confirmaron otros estudios anteriores en los que trataron los índices de ocurrencias literales de preposiciones y pronombres procesados con LIWC2001 (Pennebaker, Francis y Booth, 2001), o

aquellos que contabilizaban las ocurrencias de ciertas unidades en un texto, correlacionaban, en ambos casos, con índices de salud general de los participantes (Pennebaker, Mehl y Niederhoffer, 2003; Chung y Pennebaker, en prensa).

4.3.- Simulación de navegación en páginas WEB

Cada vez es mayor la preocupación de las grandes entidades por la usabilidad de sus portales. Actualmente estamos asistiendo a un ascenso vertiginoso de las operaciones de negocio por medio de la WEB. Podemos realizar infinidad de operaciones por medio de un ordenador y una conexión a la red: inversiones bursátiles, operaciones bancarias como transferencias, asegurar vehículos, altas en contratos telefónicos y de acceso a Internet, etc. Como no podía ser de otra forma, la captación de clientes en este ámbito viene pareja a la seguridad de las comunicaciones y a la facilidad de uso de los servicios ofrecidos. Al ser amplia la competencia, cualquier esfuerzo en la captación y fidelización de los clientes repercute en las cuentas generales. Una forma de captación y fidelización es hacer que los portales corporativos sean cómodos, ágiles y con un uso restringido a cortos periodos de tiempo por operación. En definitiva, conseguir que el usuario del servicio no abandone por el simple hecho de no comprender el funcionamiento o considerar que invierte un periodo excesivo de su tiempo. El estudio de la usabilidad trata de estudiar todo lo que concierne a las variables de factor humano en el uso de las aplicaciones. En un artículo de *New York Times* (17-12-2000) encontramos un ejemplo de la evolución sufrida por la compañía *Fidelity Investments* y cómo la usabilidad ha contribuido a mejorar sus resultados. Esta empresa optó por invertir en investigación referente al factor humano en sus aplicaciones WEB y cuenta en su departamento de "*Human Interface Design*" con un equipo de 14 personas además de disponer de dos laboratorios para realizar test de usabilidad. Los resultados se podrían resumir en que los clientes del *broker de Fidelity* aumentaron en un 37% sus operaciones por Internet. El 57% de sus altas en planes de pensiones fueron realizadas on-line. Según remarca la revista *Business 2.0* (Octubre 2002) en un seguimiento de este hecho, después de rediseñar el sitio de nuevo haciendo más clara la información que los usuarios debían

tener más a mano, el número de operaciones creció en un mes el 4%. Esto puede darnos una idea de la importancia que tiene la forma de guiar al usuario hacia la información.

Tomando otro ejemplo, un estudio de la empresa española *Usolab* (2002), advierte de la existencia de términos frecuentemente utilizados en las web de banca que no son entendidos por un alto porcentaje de usuarios. Son términos que por ser muy usados en el lenguaje financiero, acaban por formar parte del lenguaje común del personal de banca, de tal forma que no se repara en la posibilidad de que los usuarios "no bancarios" no los conozcan. En el estudio se realizaron test con 20 usuarios en 10 sitios de banca de particulares y detectaron varios términos frecuentemente utilizados en este tipo de sitios, que muchos usuarios no entienden. El estudio daba algunos ejemplos de términos conflictivos como "*Posición"*: Este término se utiliza para nombrar la opción donde se expone el resumen de los saldos de todos los productos contratados por un cliente con la entidad. Suele ir acompañado de calificativos como "Global", "Completa" o "Integral"*.* Durante los test realizados los usuarios no utilizaron apenas esta opción en tareas donde era la opción de camino óptimo para culminar la tarea. Además, en varios ocasiones, una vez finalizada cada sesión individual de test, se preguntó a los usuarios la causa por la que no la habían utilizado y comentaron que desconocían su significado. Otros bancos electrónicos utilizan vocablos más entendibles como **"**Mis Cuentas**","**Mi Cartera**", "**Situación de Cuentas y Contratos**"**. Para buscar soluciones al problema, al finalizar los test, se pasó un cuestionario a los 20 usuarios, en el que se les solicitó que puntuaran con valores del 1 al 10 la adecuación de nueve de estos términos para nombrar esta utilidad. Los términos: "Todos tus Saldos", "Resumen de Productos" y "Situación de Cuentas y Contratos" fueron, por este orden, los mejor valorados. Además de "Posición", los términos *"Extracto Integral"* y *"Mis Cuentas"* recibieron una valoración baja; no fueron claros para el usuario. Otros términos conflictivos fueron "Tablón de anuncios" que se refiere a un listado con información referida a algunos tipos de interés, comisiones y otro tipo de condiciones que el banco de España exige se ponga en lugar visible. Bajo la tarea "Buscar el tipo de interés de descubierto aplicado en las cuentas corrientes", dato que normalmente se encuentra en el "Tablón de Anuncios" de las webs, fue

encontrado sólo en un 14% de las ocasiones**.** Otro conflicto encontrado fue la diferencia entre "Traspaso" y "Transferencia". Los dos términos son confusos a la hora de seleccionar uno u otro enlace. Su significado extensional se solapa y hace dudar a los usuarios de manera que pueden desistir y criticar la ausencia del servicio. Como solución algunas entidades Incorporan una sola opción ("transferencias" ó "transferencias/traspasos") que incluye tanto la operativa de traspasos como la de transferencias; y una vez dentro de ella realizan preguntas o utilizan términos que resultan claros al usuario. Esto hace descender el impacto de los términos confundibles. Este estudio más o menos estructurado, evidencia la importancia del diseño de los interfaces y en especial, el diseño de los enlaces y guías con contenido semántico.

Los usuarios de la WEB y del software en general, están continuamente aplicando su propio conocimiento semántico cuando navegan. Para realizar una búsqueda, el usuario encontrará en su camino enlaces o “links” que le indicarán la ruta a seguir para conseguir su objetivo. La cantidad de información y la forma en que estos enlaces solapen su significado extensivo, es decir, parezcan referirse a los mismos objetivos, harán que los usuarios dilaten el tiempo invertido en sus búsquedas a riesgo de que abandonen finalmente nuestro portal y recalen en otro. La proliferación de los servicios ofrecidos vía WEB hace probable que el tiempo y el numero de “clicks” que un usuario considere aceptable para dar con lo que busca sea relativamente bajo. Por eso, los diseñadores de los portales WEB (en especial los de grandes compañías), son cada día más sensibles a introducir protocolos de usabilidad en los portales que construyen. En otras palabras, hacer un diseño centrado en el usuario.

4.3.1.- Modelo semántico de usuario

Además de su diseño visual, copioso es el caudal de razones que inclina a considerar protocolos que contengan como sustrato el modelo mental del usuario, es decir, el modelo semántico que cada usuario posee del mundo que le rodea. Este modelo semántico de los usuarios, expresaría cómo las palabras se relacionan unas con otras y de qué manera las agrupaciones de estas

palabras (enlaces de más de un término) se parecen entre sí. Algunos de los fenómenos empíricos que pueden encontrarse son los siguientes:

1. Los usuarios aplican de continuo su conocimiento semántico mientras buscan en la web. (Kaur y Hornof, 2005).
2. Por regla general, los usuarios no evalúan todos los ítems disponibles *(enlaces y cabeceras)* antes de seleccionar el que consideran acertado (Brumby y Howes, 2003). En registros oculográficos se consigna que los sujetos se saltan candidatos.
3. Incluso los factores visuales más primitivos están influidos por las propiedades semánticas de los apoyos a la búsqueda (Pierce, 1992). Procesos Arriba-abajo, (*top-down*). Esto significa que la percepción del diseño gráfico de las aplicaciones o portales estará influido por cómo los usuarios consideren las indicaciones de los enlaces, cabeceras y demás indicaciones lingüísticas e icónicas.
4. Según los datos empíricos, los usuarios tienden a elegir los enlaces del menú que porten más información (más pistas hacia sus propósitos) (Brumby y Howes, 2004). Es decir, realizan una bondad de ajuste entre la representación propia de la meta y las indicaciones proporcionadas por el diseño.
5. La elección del ítem (enlace) viene afectada por la información de los ítems que le rodean *(*Brumby y Howes, 2004).La selección entre los ítems examinados está afectada no sólo por la información del ítem que nos lleva a la consecución de la meta propuesta sino por la información de los ítems distractores.
6. La similitud subjetiva percibida por el usuario entre la información deseada y los enlaces es dependiente del valor informativo de los otros enlaces *(*Brumby y Howes, 2004). La manera en que los usuarios verbalicen que el enlace indica la ruta hacía una información buscada es dependiente de los demás enlaces.
7. Además de saltarse candidatos, los usuarios acotan los candidatos a ser elegidos en subconjuntos cada vez más pequeños (Brumby y Howes, 2004). Fijan su foco atencional en subconjuntos más pequeños de ítems (enlaces), es decir, siguen heurísticos que propician la simplificación de

la información proporcionada por los propios enlaces en menos candidatos y prescinden de algunos por no considerarlos semánticamente pertinentes en una primera evaluación. Esto puede considerarse similar al modelo de análisis racional de solución de problemas de Anderson (1990). Este modelo se basa en la evaluación de la utilidad de una reevaluación medida en costes y beneficios. Los ítems son evaluados en la medida en que se supone que evaluando otro se consigue una ganancia adicional que supere el coste de esta última evaluación. Por tanto, los sujetos son conservadores a la hora de buscar nuevos ítems que tengan mejores pistas hacia la meta y acotan los candidatos en subconjuntos para realizar reevaluaciones.

8. Cuando el ítem que indica la información buscada (la meta), lo hace de manera diferenciable (no hay otros enlaces que parezcan indicar esa misma información*),* los usuarios son mucho más rápidos en seleccionar ese ítem y lo hacen con una mayor seguridad pues fijan su atención en menos ítems después de la fijación inicial (Brumby y Howes, 2004).

9. Controlando los ítems (enlaces) confundibles entre ellos, su familiaridad medida en la frecuencia de uso y su representatividad en relación a la información deseada se mejora de forma palpable el porcentaje de aciertos, el número de "clicks" empleados y el tiempo invertido (Blackmon, Kitajima y Polson, 2003; Blackmon y Mandalia, 2004).

Todos estos datos y algunos otros pueden agruparse en un corolario común: existe una interdependencia de los enlaces, cabeceras y demás guías del menú en cuanto a su evaluación y consiguiente elección. En líneas generales el hecho de que un usuario fije su atención en uno de los enlaces depende de la recurrencia de la información que porta este enlace, además de la recurrencia de la información que portan los demás enlaces que han sido evaluados. Además, los usuarios son sensibles a la familiaridad de los términos de los enlaces y a como de representativos son esos mismos términos en relación con el tópico que es motivo de la búsqueda.

4.3.2.-Paseo cognitivo

Uno de los modelos que se han aplicado para detectar los problemas de representación semántica que pueden estar interfiriendo en el uso o navegación de una aplicación ha sido el llamado "*Cognitive Walkthough for the Web*" (*CWW*) o "Paseo Cognitivo para la Web" (Blackmon et al., 2003; Blackmon y Mandalia, 2004). Este modelo orientado a su aplicación trata de detectar los problemas derivados de los términos utilizados como guía (enlaces, cabeceras, "links") para la consecución de un objetivo. El modelo CWW es heredero de otro modelo llamado simplemente *"Cognitive Walkthough" (CW)* o "Paseo Cognitivo" propuesto por Polson et al., 1996).

El Paseo Cognitivo trata de pronosticar los problemas que tendrán los usuarios en el uso de una aplicación. Para ello, simula la navegación de los usuarios asumiendo que todas las acciones de este irán destinadas a la consecución de una meta. Este modelo identifica los problemas de usabilidad simulando paso a paso en una interfaz la conducta del usuario en una tarea dada. El equipo de expertos que lleva a cabo la simulación ha de contestar a un protocolo de preguntas claves*:*

1. ¿Está clara para el usuario la acción correcta para la consecución de las metas?
2. ¿Identificará el usuario la descripción de la acción correcta con lo que él trata de hacer?
3. ¿Interpretará el usuario correctamente la respuesta del sistema para una acción?

El modelo *"Cognitive Walkthough for the Web" (CWW)* asume como herencia los postulados de este anterior modelo pero introduce mejoras en cuanto a costes y efectividad. Como el anterior, concibe que el usuario siempre tiene un propósito. CWW propone que para generar una acción en busca de este propósito (apretar un botón, un enlace, un "link", etc.), el usuario involucra dos procesos*:*

1. Divide la página en regiones y atiende a las regiones correctas (aquellas cuyas descripciones están de acuerdo con sus propósitos)
2. Elige un enlace y actúa sobre él.

Fruto de estos dos procesos, el modelo CWW toma el anterior protocolo de preguntas pero introduciendo dos preguntas fundamentales que emanan de la pregunta número 2.

> 2a) ¿Identificará el usuario el título de la subregión correcta con la representación propia del propósito que desea llevar a cabo?
>
> 2b) Usando los enlaces, "links" y otra clase de informaciones en dichas subregiones ¿identificará el usuario las descripciones de estos controles como "pistas" que indican una posible consecución del propósito de lo que quiere llevar a cabo?

Para evaluar estas preguntas, CWW introduce el uso de un modelo de conocimiento de un usuario prototipo. No sólo simula este conocimiento con expertos sino que lo hace modelando el conocimiento que el usuario trae consigo a la hora de enfrentarse con nuestra aplicación.

Para evaluar si las descripciones de títulos y enlaces son identificados por el usuario como la propia senda hasta la consecución de su propósito, lo que hace este modelo es analizar si los términos que se utilizan pueden estar incurriendo en alguno de los problemas que emanan de los hechos empíricos anteriormente descritos, a saber: los sesgos en la selección provocados por la interdependencia de las descripciones de títulos y enlaces, la familiaridad de las descripciones y la representatividad que estas tienen del propósito o meta del usuario. Si los términos de las descripciones incurren en algún tipo de problema, habrá que retirarlos y sustituirlos por otros.

Para evaluar estos términos y su conveniencia, hace uso de un modelo de representación mental de los usuarios. Este modelo mental es una representación semántica de los términos en la mente de los usuarios, es decir, cómo los usuarios tienen representados los términos en su modelo mental y cómo, a consecuencia de esa representación, se relacionan unos con otros.

Si poseemos un modelo de la representación mental de los usuarios podemos analizar los enlaces que encontramos en una aplicación y optimizarlos para que haya una navegación ágil y sin los problemas que emanan de los hechos antes descritos. Una forma de disponer de dicho modelo es emplear una técnica como LSA. Si contamos con un modelo mental de las relaciones semánticas que manejan los usuarios a la hora de enfrentarse con nuestra interfaz, podríamos predecir los ítems que resultaran deficientes y

sustituirlos por otros. Es decir, si los ítem son interdependientes unos de otros y la información semántica influye hasta en el mismo diseño, debemos aislar los grupos de ítems que son confundibles unos con otros en cuanto a la representación de la meta, los ítems que no son representativos de ninguna meta pues apenas evocan información relacionada con ella, los ítems poco familiares bajo el modelo mental del usuario, etc. Ahí es donde se introduce LSA como simulación de modelo mental de distintos tipos de usuario (grupos sociales). En definitiva, el modelo mental proporcionado por el análisis de la semántica latente (LSA) hace algo que ni siquiera los expertos entrenados en usabilidad y psicología cognitiva pueden hacer: predecir las acciones que llevará a cabo un grupo de usuarios cuyo modelo mental o sustrato de conocimiento es muy diferente al de estos mismos expertos (Blackmon)**.** Una vez que poseemos un modelo mental de la representación semántica de un determinado perfil de usuario que utilizará nuestra aplicación, podemos realizar el análisis de la interfaz.

4.3.3.- Análisis de la interfaz

El primer paso es formular cuales son los propósitos o metas que desea conseguir el usuario cuando entra en una página web como la que se está analizando. Lo ideal es entrevistar a potenciales usuarios y hacer una lista de sus propósitos de tal manera que nos describan que información es la que desean encontrar. Hecho esto, el modelo introduce ciertos *parámetros a medir*:

I) SIMILITUD: Al permitir LSA medidas de similitud entre los vectores representados en la matriz*,* sean estos términos, documentos o documentos introducidos a (Figura 33) *,* podemos medir de qué manera se relacionan los títulos y enlaces entre sí (en qué medida se confunden entre sí) y cómo se relaciona el título o enlace que se considera correcto con el texto o meta final (en qué medida porta información sobre la meta).

1) Se considera que una enlace considerado correcto proporciona pocas "pistas" de la meta a encontrar si la similitud entre los dos medida con el coseno del ángulo que dejan los dos vectores entre si no llega a 0.10 (coseno<0.10).

2) Se considera que dos títulos o enlaces son confusos entre sí cuando el índice de similitud entre ellos medido de la misma forma es mayor que 0.60. (coseno>0.60).

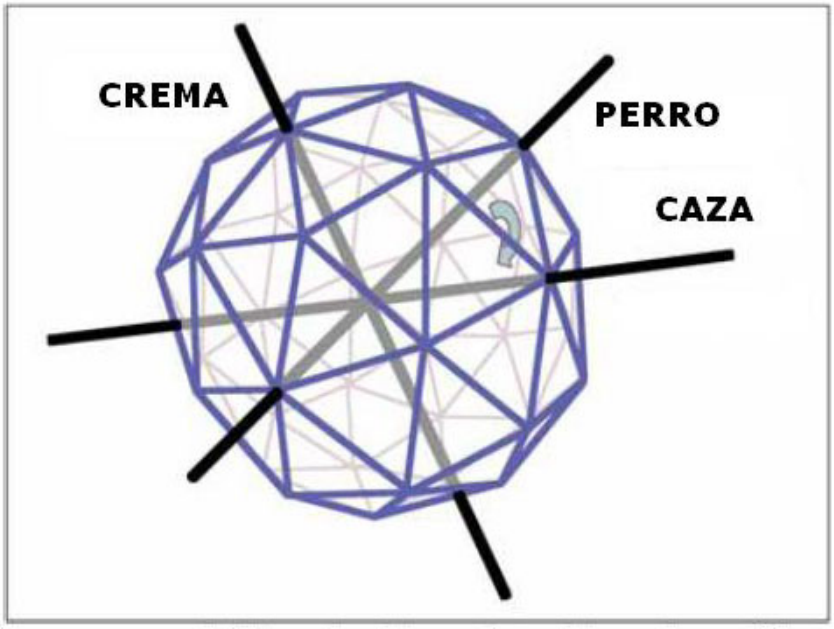


Figura 4.2. Vectores en un espacio tridimensional. Los vectores del espacio semántico-vectorial que representan términos y documentos, son susceptibles de comparación calculando el coseno del ángulo que queda entre dos vectores.

II) FAMILIARIDAD: Se consideran dos formas de medir la familiaridad de los términos. La primera es medir la frecuencia de aparición de un término en un corpus representativo del conocimiento de nuestro usuario prototipo. La segunda es acudiendo otra vez al modelo semántico-vectorial generado por el análisis de la semántica latente (LSA) y consignando la longitud del vector que representa ese término. Esta medida dará cuenta de la cantidad de información que porta un término. Puede decirnos "cuanta información posee el análisis LSA de ese vector.

Los términos sobre los que el análisis porta más información tendrán vectores con mayor longitud porque aparecen muy frecuentemente en el

corpus en diversos contextos. Lo que muestra esta longitud es que ese término está bien representado en los documentos y podría representar bien alguno de los conglomerados que se forman dentro del universo semántico (Kintsch, 2001). Si un término posee un índice alto de similitud con un tópico, interesa que sea familiar pues se deduce que los usuarios conocen bien el término.

1. Para la primera forma es necesaria una frecuencia de 15 en adelante.
2. Para la segunda forma se hace necesaria una longitud de vector superior a 0.55 para una palabra y mayor de <0.80 para las estructuras complejas de varias palabras (recordemos qué la longitud de un vector frase es mayor que la de un vector término y a su vez, ambas longitudes son menores que la del vector párrafo). Toda longitud menor de estos umbrales serán considerados insuficientes.

4.3.4 Un caso de estudio

Un caso analizado por Blackmon y Mandalia (2004) fue el análisis sobre los enlaces y cabeceras del interfaz de la enciclopedia ENCARTA, utilizando para ello la aplicación computerizada del CWW que ellos mismos diseñaron. Para obtener una mayor parsimonia experimental diseñaron una interfaz simplificada en la que se homogenizaba el diseño y la ubicación de enlaces y cabeceras. En una caja de texto en la parte superior se ofrecía a los sujetos experimentales un párrafo sobre un tema y la tarea consistía en pinchar en donde buscarían. Los temas podían ser desde razas, tipos de música, cibernética, etc., pero ningún texto contenía palabras no-familiares.

Tanto los modelos de espacio semántico surgidos del análisis LSA como usuarios reales de distintos niveles escolares son probados en la tarea. Primero se pronostican los problemas con los modelos de usuario LSA y luego pasan los usuarios reales. De los 154 enlaces que conectan los temas, tomando diferentes espacios semánticos, el modelo pronostica que 79 son no-familiares para los niños de tercer grado (8 años), reduciéndose esta cifra a 27 en el espacio semántico de sexto grado. En definitiva, el pronóstico sería, por ejemplo, que un niño de tercer grado tendría bastantes problemas de

usabilidad a la hora de buscar un artículo en la enciclopedia ENCARTA. Palabras como Paleontología (0.06), lo oculto (0.08), arqueología (0.10) ofrecerían dificultades a un niño de tercer grado dado su baja longitud de vector. Serían palabras no-familiares. Para solucionar esta falta de familiaridad se puede optar por añadir a la existente otra más familiar o sustituirla por otra más familiar. Por ejemplo, sustituir "Paleontología" por "paleontología y fósiles".

El siguiente análisis se hace en relación a la similitud y representatividad que tiene cada enlace de sus respectivos temas a buscar y cada cabecera de sus enlaces además de la similitud de cada enlace meta con los demás enlaces. Para ello se relaciona, mediante el análisis de similitud de los cosenos de cada texto, el tema a buscar con un pequeño texto compuesto de las palabras que más tienen relación con cada enlace (link) y cada cabecera con sus enlaces. Esto dará cuenta de si hay enlaces y cabeceras compitiendo entre sí y si los enlaces representan o no el texto a buscar. Tomemos, por caso, el ejemplo, "Buscar un artículo sobre Hmong", en el que compiten las cabeceras "ciencias sociales" con el enlace "países". En la cabecera "Historia", hay tres enlaces compitiendo por ese misma meta como son "Historia de Asia y Australasia", "Historia de Estados Unidos" y "Gente de los Estados Unidos" , y en "Geografía" los enlaces "países" y "regiones del mundo" se solapan entre sí. Una posible solución sería hacer enlaces múltiples desde cada una de los enlaces o agrupar pues si no estaríamos ante muchos enlaces que resultan confundibles entre ellos.

Las correlaciones de los pronósticos con usuarios reales medidas estas antes y después de corregir los problemas pronosticados por el sistema corroboraron que los pronósticos eran acertados. Los modelos proporcionados por los espacios semánticos surgidos del análisis LSA (de cada uno de los grupos) parecen ser eficientes a la hora de usarse como usuarios prototipo de cada grupo y detectar sus posibles problemas.

4.3.5 Adecuación de ruta

4.3.5.1 Definición

Un nuevo índice extraíble de los parámetros que proporciona LSA y que tiene relevancia a la hora de medir la navegación WEB es el llamado "adecuación de ruta" (*path adecuacy,* véase a este respecto Juvina y van Oostendorp, 2005 b). Este índice añade complejidad a los índices de "similitud" y "familiaridad". En particular:

- La "adecuación de ruta" consiste en cuan similar sea semánticamente la ruta a la meta del usuario, entendiendo "ruta" como el conjunto de enlaces y links que guían hacia esa misma meta.
- Aunque se pueden introducir variaciones debidas a los hechos empíricos referentes a la memoria de trabajo, a la cual se quiere emular (Juvina y van Oostendorp, 2006 a), la ruta es concebida como un todo, es decir, se representa con un solo vector en el espacio semántico resultante del análisis LSA (Es conveniente recordar que en el análisis LSA, cualquier término o conjunto de términos pueden representarse por medio de un vector que formará parte de un espacio semántico).
- Es importante señalar también que se entiende por ruta a cada uno de los enlaces que guían hacia una meta desde el primero hasta en el que se encuentra el usuario en un momento preciso. En otras palabras, desde el enlace desde el que se parte hasta el enlace último por donde se pasó.
- Se asume que la selección de los ítems o enlaces posteriores no sólo depende de la relevancia de la meta final con respecto a dichos ítems por separado sino de la consistencia que tiene toda la ruta considerada enlace a enlace. De ahí que se calcule la relación semántica de la ruta en conjunto con la meta del usuario y no la relación de cada enlace por separado.

4.3.5.2.- Sesgo de la memoria a corto plazo

De una manera general se asume que una ruta está compuesta por varios enlaces. De los datos empíricos descritos en la literatura experimental (e.g., Brumby y Howes, 2004) se desprende que la manera en que los componentes de esta ruta sean semánticamente consistentes unos con otros y con la meta, influirá en que el usuario vuelva atrás o desista directamente. Este postulado está basado en que la memoria de trabajo mantiene una representación de los enlaces anteriormente empleados "recordando" elecciones pasadas de manera que representaciones "residuales" de enlaces anteriores sesgan la representación de los presentes e inducen selecciones futuras. Juvina y van Oostendorp (2006) introducen este particular en una simulación que ellos llaman *COLIDES+,* pues es heredero de *COLIDES* elaborado por Kitajima, Blacmon y Polson (2000). La diferencia entre uno y otro estriba en que el COLIDES anterior no introducía en el análisis la posibilidad de que el proceso de "vuelta atrás" del usuario fuera debido a las propiedades de la ruta como un todo. En *COLIDES+* se intentan formalizar los procesos de formación de los conceptos o, lo que es lo mismo, un compendio entre el conocimiento estático, que puede ser representado con una matriz LSA, y el sesgo introducido por la memoria a corto plazo, que mantiene representaciones activadas e induce a distintos tipos de formación de conceptos, en la línea de que los conceptos son temporales y relativamente evanescentes (Kintsch, 1998).

4.3.5.3.- Procedimiento para pronosticar anomalías: “el próximo mejor” (“next best”)

El procedimiento para pronosticar anomalías sigue los dos pasos siguientes:

1. Se calcula "adecuación de ruta" en cada instante de la navegación, calculando el vector que representa la ruta, tomando los enlaces desde el primero de la ruta hasta el que se encuentra en ese momento el usuario y se compara semánticamente con la meta. Los mismos autores señalan que hay razones para pensar que no se debería tomar toda la ruta por igual, sino ponderar la participación de los enlaces por su proximidad al actual pero lo dejan para sucesivos estudios.

2. Si al pinchar en otro enlace (introducir otro enlace en el vector que representa la ruta), no aumenta o disminuye la "adecuación de ruta", se pronostica abandono o "vuelta atrás". Para pronosticar que el usuario cambiará o volverá para atrás, los autores emplean la siguiente estrategia (Figura 34 y 35): el usuario cambiará o volverá para atrás si un link no aumenta el índice de "adecuación de ruta". En otras palabras, si pinchando en otro link y después de introducido este nuevo link en el cálculo del vector nuevo de "ruta", la "adecuación de ruta" no asciende, las posibilidades de que el usuario cambie o vuelva para atrás aumentan.

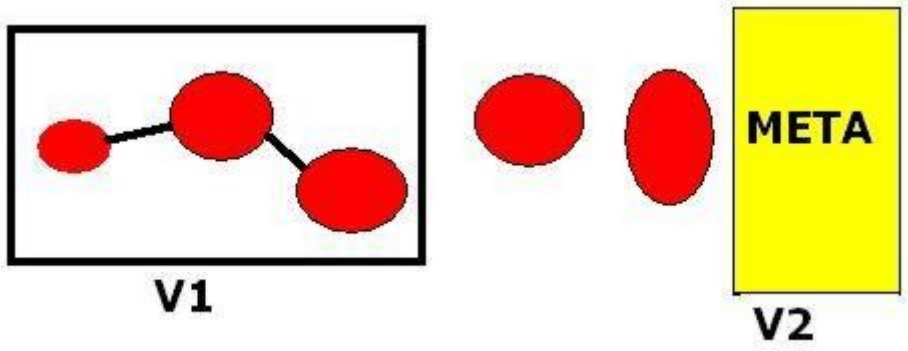


Figura 4.3.- En el momento 1, se calcula un vector tomando todos los enlaces que componen la ruta hasta ese momento, este vector resultante se comparará con el vector que representa la meta V2. Esto dará cuenta de la "adecuación de ruta" en ese instante.

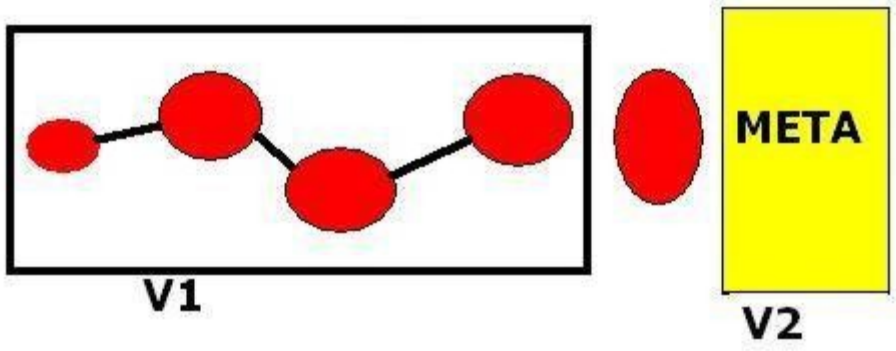


Figura 4.4.-. En momento 2 de la navegación, se volverá a calcular el vector de ruta V1 pero introduciendo el nuevo enlace. De la misma manera, se compara con la meta obteniéndose al "adecuación de ruta". Si la adecuación de ruta es igual o menor que la anterior, se pronosticará abandono o "vuelta atrás" del usuario.

4.3.5.4.- Datos empíricos

Esta predicción parece encontrarse en los datos empíricos de estos autores en la comparación entre usuarios y datos simulados. Sin embargo, el sistema de simulación que lleva como fundamento un espacio semántico producido por LSA está hipersensibilizado y provoca un comportamiento más vacilante que los usuarios. Esto es debido a que por la propia naturaleza del espacio semántico de contenido general, es frecuente encontrarse con casos como el que refieren los autores: "hotel"-"sleep" tienen un coseno de 0,24 mientras "hotel"-"wait" tienen un coseno de 0,34. Esto hace que al ser un algoritmo de todo o nada, la conducta de la simulación vuelva un número superior de veces hacia atrás a reevaluar otra vez las alternativas. En este artículo es sugerido que sería bueno probar con espacios semánticos de contenido más específico. No obstante, en estos primeros resultados, la tendencia es que el índice de "adecuación de ruta" puede predecir el cambio de ruta o la vuelta atrás.

Este tipo de modelos se fundamentan en que la construcción de un significado está basado en el conocimiento previo más el contenido activado en la memoria de trabajo (Kintsch, 1998). Este contenido activado proviene de la misma lectura de los enlaces presentados en la pantalla sucesivamente. A su vez, el conocimiento previo es representado por el espacio semántico proporcionado por la matriz LSA. De esta forma, los contenidos semánticos de

los links anteriores de la ruta participan de alguna manera, siquiera en forma de gradiente, en la interpretación de los nuevos y en la representación de la cercanía a la meta.

Es esta unión de memoria activada y conocimiento la que propicia la comprensión de textos incluso tan elaborados como la metáfora o las inferencias - véanse los modelos mucho más elaborados de Kintsch (2001) y Kintsch y Bowles (2002) para un modelo de comprensión de la metáfora y las inferencias a partir de un conocimiento estático proporcionado por el análisis de la semántica latente y la activación de contenidos que propician los componentes de una predicación -.

4.4.- Recuperación de la información

Las técnicas de recuperación de información nacen de la necesidad de extraer automáticamente información desde repositorios de datos de gran tamaño y en dónde se encuentran referencias potencialmente valiosas. Para llevar a cabo esta propósito, han de elegir, de entre numerosos candidatos, sólo la información que se considere pertinente a la búsqueda, es decir, analizar bajo unos criterios lo que es pertinente y lo que no lo es. La manera común de hacer búsquedas en grandes repositorios de texto es empleando lo que se ha llamado búsquedas literales (*lexical matching*), es decir, el sistema constata o no la aparición literal del término o los términos de la búsqueda en los textos en los cuales realiza esa búsqueda. Este tipo de técnica en su forma básica es llamada también búsqueda booleana debido a que indaga si es cierto o falso la ocurrencia del término buscado (Kiefer, 2006). El resultado será la recuperación indexada de los documentos en los que esos términos salen ordenados según alguna regla lógica (termino1 y termino2 > termino1 o término2, etc.). Estas búsquedas literales tienen el problema de que no recuperarían documentos en los que no ocurriesen las palabras de la misma búsqueda desdeñando aquellos documentos en los que ocurriesen términos muy relacionados semánticamente con aquellos de las búsquedas. Este es el

caso de sinónimos, polisemia, partonomías, meronimias, etc. Dumais (2003) hace una buena reflexión sobre lo que supone un sistema con búsqueda literal y cómo se ha intentado corregir sus deficiencias. Según la autora, debido a sus deficiencias, los sistemas tradicionales fallan en dos cuestiones fundamentales, recuperan demasiada información irrelevante y fallan sin embargo en recuperar la relevante. Por ejemplo, con una búsqueda literal como "tumor" no se recuperarían documentos que contuviesen "neoplasma". Todas las mejoras destinadas a mejorar las deficiencias de los sistemas tradicionales van en la dirección de reducir la variabilidad de términos y formas para referirse a referentes similares. Para ello se han hecho algunos intento fructíferos que han dado algunos resultados. El primero de ellos es el empleo de las técnicas de lematización (*stemming*). Esta técnica se basa en eliminar las desinencias y conservar las raíces de los términos de manera que palabras de la misma familia sean representadas por un mismo término. Esta técnica tiene la ventaja de poderse implementar mediante reglas y puede ser empleada mediante procedimientos automáticos. "cognitive", "cognition","cognate" y "cognitively" pueden ser reducidos a "cognit". En cualquier caso, estos métodos poseen también un margen de error al lematizar bajo el mismo lema términos que pueden no ser de la misma familia semántica sino coincidir ortográficamente. También hay lenguas que no aceptan de una manera directa la utilización de estas técnicas. Tampoco esta técnica serán sensible a la sinonimia como por ejemplo: "physician" y "doctor" cuya similitud no percibiría.

Una segunda forma para salvar algunas deficiencias de la primera es emplear vocabularios controlados. Los vocabularios controlados no es más que agrupar previamente los términos en categorías funcionales o temáticas. Se harán búsquedas con la ventaja de que la palabra de la búsqueda enlazarán con otras que no tienen por qué coincidir ortográficamente. Aún así, esto tiene la desventaja de que las creaciones de este tipo de vocabularios son costosas y que no siempre se extraen los resultados deseados. Otras técnicas son las llamadas técnicas estadísticas y se basan en el análisis y la distribución de grandes muestras del lenguaje. Estas técnicas también son habitualmente llamadas "basadas en aprendizaje" o "basadas en corpus"(*learning-based*) e incluyen el modelo espacio-vectorial de Salton (1968), LSA y las redes neurales

artificiales. Aunque más tarde fue cobrando interés para otras utilidades, entre ellas el modelado de conocimiento, LSA fue introducida por primera vez con este propósito, es decir, con el propósito de una técnica de recuperación de la información Deerwester (1990) y se ha seguido empleando para estos menesteres. Todos los sistemas de recuperación de información se pueden referenciar respecto a su efectividad. Una forma operativa de consignar esta es calculando las medidas estándar que se utilizan habitualmente, a saber: Precisión y Exhaustividad (*Recall*). Deniston (2003) hace una buena definición de ambas medidas.

> **Precisión:** se trata de la habilidad del sistema de recuperar sólo documentos relevantes. Responde a la pregunta de ¿es lo que he encontrado relevante en relación a lo que estoy buscando? Por ejemplo: Si el sistema de búsqueda obtiene 80 documentos pertinentes a la búsqueda pero sólo 20 lo son de verdad, la precisión será de un 25%.
>
> **Exhaustividad** (*Recall*): La virtud con la que un sistema encuentra todos los documentos relevantes a una consulta. Responde a la pregunta ¿cuánto se ha dejado fuera?. Por ejemplo, podría haber 100 documentos recurrentes pero sólo haber extraído 80. En ese caso el índice *recall* será de 80%.

Habitualmente se registra la eficiencia del sistema poniendo la precisión en base a la exhaustividad (Dumais, 2003). De esta manera, se representan unas curvas parecidas a las curvas ROC de la teoría de detección de señales y se muestra la capacidad de los sistemas de mantener un cierto equilibrio entre estos dos índices ya que podría darse el caso de sistemas que recuperasen un gran porcentaje de documentos relevantes de la totalidad (una alta exhaustividad) pero que trajeran consigo un gran porcentaje de falsas alarmas (baja precisión). Tanto Deerwester (1990) como Dumais (2003) obtienen buenos resultados en la utilización de LSA como técnica de recuperación de información. Esta última autora encuentra que teniendo en cuenta las curvas que relacionan Exhaustividad con Precisión, LSA puntúa un 30% más de media en precisión que las búsquedas literales en todas las escalas de Exhaustividad y por ejemplo, con una exhaustividad del 0,5, el 68%

de los documentos extraídos por LSA son relevantes frente al 40% en las búsquedas literales.

Actualmente aún es posible encontrar discusiones en torno a si LSA/LSI esta implementado o no en las nuevas aplicaciones de búsqueda de Google en los foros SEO (*Search Engine Optimization*) y SEM (*Search Engine Marketing*) y encontrar acaloradas discusiones. Los perfiles SEO y SEM son comúnmente conocidos por participar personas o empresas dedicadas a posicionar una página WEB de una compañía en las primeras posiciones de los buscadores. En otras palabras, se ocupan de garantizar que la WEB del cliente se posicione en los primeros puestos dadas unas búsquedas potenciales. Por este motivo, estas empresas se ven avocadas a practicar la *ingeniería inversa*, es decir, inferir los algoritmos de los sistemas de búsqueda en base a las pruebas que ellos realizan y, así, nutrir a las páginas de todos los textos y las palabras clave que propicien que las búsquedas recalen en la página deseada, consiguiendo así un buen rango de posicionamiento de la página en los principales buscadores (*Page Rank,* en el caso de Google). Además, tienen que contar con posibles penalizaciones que vienen dadas por una serie de malas prácticas definidas por cada buscador, si es el caso (e.g., que se descubran "artificialidades"). Como últimamente viene siendo habitual que los buscadores valoran los parámetros semánticos, resulta entonces muy importante para estas compañías "descubrir" que algoritmos utilizan Google y otros buscadores para establecer dichos parámetros semánticos. En Google, por ejemplo, esta manera de búsqueda semántica está en fase de pruebas y una búsqueda de estas características es posible añadiendo el símbolo "~" antes del términos buscado (e.g., ~zoo ~trips). Adjuntamos algunas muestras de estas discusiones para dar cuenta del estado de opinión en torno a LSA como motor de búsqueda y técnica de posicionamiento en buscadores.

- Buscadores de Microsoft:

http://www.roncastle.com/seo-newsletters/microsoft-live-latent-semantic-indexing.htm

- Página de LSI aplicada a SEO/SEM:

http://www.latentsemanticindexing.co.uk/category/tools/

- Bajo el título *Major Google Changes: Latent Semantic Analysis?* y la cabecera *This thread is for discussion of recent changes at Google that may be due LSI/LSA factors*. http://forums.searchenginewatch.com/showthread.php?t=4009

- Bajon el título *Google Feb. 2005 Update: Observations About Changes*. http://forums.searchenginewatch.com/showthread.php?t=4082

- Con el título: *Latent Semantic Analysis (LSA) - Crawl into the Google Algorithm? http://www.seroundtable.com/archives/001478.html*

- Una compañía que explica LSA como solución: Se trata de Orange County SEO y explica a sus potenciales clientes lo que significa LSA en el mundo del marketing dentro de internet. http://www.jasontchandler.com/internet-marketing-talk.htm

- Una página de opinión sobre SEM en el que se propone LSA (en italiano): http://www.soupmarketing.com/category/sem/

- Otra página de posicionamiento: http://www.brianshoff.com/tech/latent-semantic-indexing-lsi-simplified.htm

- Una compañía de posicionamiento en buscadores cuyos servicios estan basados en parte sobre LSA. http://www.sem.mosaic-service.com/

- Una página en la que se infiere que Google confiere mucha importancia a los índices basados en LSI. Con el título: http://www.searchenginepromotionhelp.com/m/articles/search-engine-optimization/google-semantic-analysis.php

4.5.- Enrutamiento automático de llamadas en IVR

4.5.1.- Reconocimiento, enrutamiento y diálogos

Tanto los servicios telefónicos como las aplicaciones **IVR/VRU** (*Interactive Voice Response*), en general, han introducido a las nuevas tecnologías ciertos parámetros que le son propios, en lo que respecta a la gestión y explotación automática de los diálogos que se producen en su ámbito. Según Dybkjær y Bernsen (2001), son tres los pilares que sustentan

esta tecnología. Son los siguientes:

(1) El primero y más maduro es el que se refiere al propio reconocimiento de voz (*IVR*) y se basa principalmente en técnicas de reconocimiento de patrones de la señal de entrada. Estas técnicas han sido implementadas en herramientas que hoy en día son fáciles de encontrar en el mercado y que son las que están siendo empleadas masivamente en los servicios ofrecidos mediante telefonía móvil y fija. En estas aplicaciones, habitualmente se definen unas posibles entradas por medio de las combinaciones definidas en unas gramáticas[5]. El proceso empareja la entrada con uno de estos ejemplares definidos en las gramáticas bajo unos umbrales de probabilidad.

(2) El segundo es el que se refiere a la Gestión del Diálogo (*Dialog Management*) y trata de determinar qué hacer o por dónde encauzar al llamante en el supuesto de que haya sido reconocida una respuesta u otra. Este segundo proceso presupone un cierto conocimiento de las intenciones del usuario. Habitualmente se lleva a cabo por medio de preguntas dicotómicas y menús. Reconocida una entrada, se lanza una salida concreta a la aplicación principal y esta, mediante una estructura condicional, dirige la aplicación a uno u otro sitio. Las técnicas basadas en LSA también se pueden considerar dentro de este segundo pilar, es decir, dentro de la Gestión del Diálogo ya que LSA, aunque de otra manera, contribuye a decidir, a la vista de una entrada, cuál es la intención del usuario y dónde dirigirlo.

(3) En el tercero, Generación de las Salidas (*Output Generation*), se diseñan las respuestas que serán ofrecidas al usuario en el supuesto de una cierta demanda. Es importante introducir en su diseño modelos y directrices de usabilidad que provienen de la observación de las interacciones entre usuarios y agentes reales que fueron recopilados en las mismas situaciones que en las que se van a encontrar los agentes virtuales. Un sistema formal de evaluación de usabilidad en los diálogos lo proporciona *CODIAL* (Dybkjær, Bernsen y Dybkjær,

[5] Una gramática es una formalización de la combinatoria de las posibles palabras o frases que puede responder un usuario y los valores que se devolverán en caso de detectar unas respuestas u otras. Además, se podrán asignar pesos a cada estructura según se estime su frecuencia de aparición. Póngase por caso que el sistema nos pregunta "Nuestro servicio le ofrece una amplia gama de equipos informáticos como ordenadores portátiles, de sobremesa o enrutadores, ¿en que tipo de equipo está interesado?. En este caso, se han de crear gramáticas para que el sistema IVR pueda reconocer por ejemplo estructuras del tipo "estoy interesado en portátiles", "estoy interesado en ordenadores de sobremesa", "en portátiles", "en ordenadores portátiles", "en routers", "quiero un portátil" y así hasta que cubra un segmento suficiente de las posibles entradas. Existen varios estándares entre los que se encuentran GSL, ABNF y grXML.

1998), el cual, por un medio de códigos y descripciones, nos permite etiquetar los diseños en cuanto a sus posibles errores de usabilidad. Este sistema se puede obtener de la página del proyecto DISC en el que se encuentran pautas y ejemplos de su uso (http://www.disc2.dk/tools/codial/index.html). Es muy ventajoso emplear este tipo de estándares en la aplicación y pueden ser incluso integrados en los diseños VISIO y de otras aplicaciones en forma de códigos numéricos.

Como no pasa desapercibido, en estos dos últimos pilares (Gestión del Diálogo y Generación de las Salidas) entra en juego el conocimiento de las intenciones y deseos del usuario y la gestión de todo lo concerniente a sus percepciones, creencias, niveles de ansiedad ante una operación, sobreestimación de las recompensas, etc. En estos casos se ha de conocer, en primer lugar, cuáles son los deseos del usuario para, en segundo lugar, diseñar diálogos, bien para informar de los pasos para conseguirlos y los compromisos que contraerá con ello, bien para informar de la imposibilidad de llevarlos a cabo y cuál es la situación que se mantiene. Las técnicas basadas en LSA tiene el propósito de identificar y dar significado a los deseos de los usuarios de manera que puedan ser dirigidos a donde puedan llevarse a cabo. Dada una demanda, los sistemas basados en LSA la clasificarán en una categoría y dirigirán el flujo donde se informe o se actúe en torno a los tópicos de esa categoría.

4.5.2.-LSA y enrutamiento

La forma clásica de gestionar los diálogos y acciones en las aplicaciones de Gestión de Diálogo es el empleo de menús y preguntas más o menos cerradas. De esta manera, se puede acotar el rango de posibles respuestas para que estén contempladas en las gramáticas que la aplicación maneja, es decir, formen parte de las posibles combinaciones que las gramáticas ofrecen en esa parte de la aplicación. En otras palabras, para comparar las palabras o frases del usuario con los candidatos que sirven para enrutarle por una u otra rama de la aplicación. Una de las principales limitaciones de la forma clásica es que el usuario está obligado a verbalizar ciertos términos o expresiones previamente propuestos y es obligado, a su vez, a recorrer parte de la

aplicación para ir perfilando lo que quiere. Así, sería imprescindible atravesar algunos menús antes de que el sistema esté en condiciones de ofrecer aquello que se viene buscando, lo que quizás sería un hándicap para los usuarios, en especial para los considerados expertos[6].

En la figura 4.5. se representa un ejemplo ficticio del diseño de la interacción entre sistema y usuario y con él podemos hacernos una idea de un diseño de estas características. Si buscásemos por ejemplo alquilar un vehículo, tendríamos que atravesar por lo menos dos menús. El primero sería el referente a la ruta Compras-Alquiler y seguramente, en lo sucesivo, atravesaríamos otro que identificase si se trata de una compra o de un alquiler. A su vez, por ejemplo, podríamos encontrar aún otra opción sobre si el tipo de vehículo a alquilar es un vehículo industrial, un vehículo de uso individual o vehículos de dos ruedas. Estas desventajas se dan, si cabe en mayor medida, para sistemas cuyos menús están regidos por tonos.

No obstante, algunas aproximaciones interesantes han tratado también de atajar el problema de la secuencialidad y abrir la posibilidad de que el usuario pueda saltarse o cambiar el orden de menús seleccionando directamente opciones que están presentes en niveles más profundos de la arquitectura de la aplicación. Destaca el uso de los Formularios de Iniciativa Mixta (*Mixed Initiative Forms*) que pueden ser implementados con estándar VXML *(Voice eXtensible Markup Language version 1.0 W3C Note 05 May 2000)* o la Interpretación Robusta del Lenguaje natural (*Robust Natural Language Interpretation*) propuesta en la plataforma NUANCE (NUANCE, 2001, p196).

[6] Si bien es verdad que la personalización de la aplicación a partir de la identificación del tipo de usuario (experto o no-experto) y la posibilidad de interrumpir las locuciones antes de que estas terminen puede mitigar la incomodidad generada por los continuos menús, esta no desaparecerá ya que aunque con menús abreviados y sin el acompañamiento de explicaciones redundantes, la navegación seguirá siendo secuencial y jerárquica.

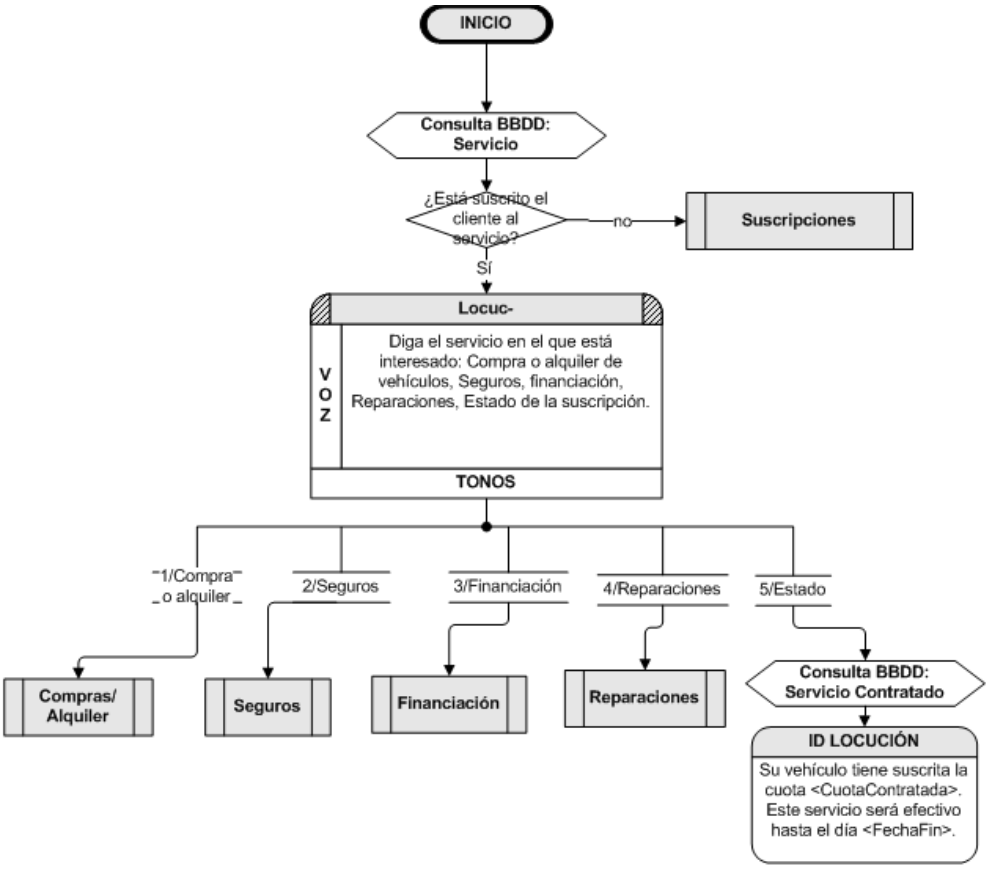


Figura 4.5.- Ejemplo de un diseño supuesto de la interacción entre sistema y usuario.

Esto permite que el usuario pueda tener la opción de demandar una información o acción que en ese momento no es ofrecida explícitamente y saltar literalmente a otra parte de la aplicación donde sí es ofrecida. Además, también se pueden proporcionar varias porciones de la información requerida en un mismo momento y en el orden que se desee[7]. Este último efecto es parecido al que se obtiene cuando se introducen datos de búsqueda en la caja de texto de "Google Maps". Se pueden introducir los datos en el orden que se quiera e incluso obviando algunos de ellos (http://maps.google.es/maps). Al igual que se reducen en "*Maps*" las cajas de texto, también se reducen en este tipo de aplicaciones el número de diálogos que requieren verbalización de datos.

Con todo, esta forma de implementación tiene varios inconvenientes, a saber, se hace inoperante cuando la variabilidad de los valores que pueden

[7] Merece la pena revisar http://www.developer.com/voice/article.php/3413361 para hacerse una idea de la filosofía de este tipo de diseños y como el usuario puede ofrecer la información requerida variando el orden inicial de la aplicación u ofreciéndola toda junta.

tomar las entradas es extremadamente grande (haciendo que una gran parte de estas entradas queden fuera del alcance de las gramáticas) y además, estas gramáticas están diseñadas para tener en cuenta el reconocimiento de los nexos sintácticos y el orden de las palabras lo cual agrava más la dificultad en un medio de extrema variabilidad.

Una forma de solventar parte de estos problemas son las técnicas basadas en modelos estocásticos del significado o modelos semánticos del lenguaje. Estas técnicas no son tan sensibles a las degradaciones léxicas y sintácticas pues no suelen conceder importancia al reconocimiento de palabras cerradas ni al orden y ocurrencia de las abiertas[8] y debido a la representación vectorial de un vocabulario muy amplio, pueden someter a categorización semántica estructuras de gran variabilidad. Existen herramientas cerradas para integrar modelos semánticos en aplicaciones de gestión de diálogo como, por ejemplo, la aplicación "*Call Steering*" de *NUANCE* (http://www.nuance.com/callsteering/) o "*Natural Language*" de *INFINITY* (http://www.naturalanguage.es/). Sin embargo, este artículo se ocupará de analizar las experiencias llevadas a cabo con LSA. Como técnica basada en un modelo semántico del lenguaje, LSA tiene las mismas ventajas de los paquetes anteriormente mencionados[9], pero con la particularidad de que se puede controlar todo el proceso de modelado del lenguaje. Al igual que las demás aproximaciones de "*Call routing"*, implementar un sistema con LSA significaría que no sería necesario proponer posibles entradas al usuario (posibles palabras a verbalizar), sino que simplemente, se le ofrecen preguntas iniciales y abiertas del tipo: "Bienvenidos a nuestros servicios bancarios, ¿en qué podemos ayudarle?". A partir de esta pregunta, la respuesta del usuario será identificada entre unas posibles categorías las cuales, condicionarán las rutas a seguir.

[8] Se apela a la distinción clásica entre palabras cerradas o de función y palabras abiertas o de contenido. Las primeras, determinantes, preposiciones, etc., carecen de significado y sirven como nexo entre las segundas: verbos, adjetivos, adverbios, etc.

[9] Hágase la salvedad en cuanto a que los paquetes cerrados facilitan el reconocimiento de producciones espontáneas y su trascripción en base a las mismas muestras del lenguaje que servirán para la formación de los modelos semánticos. Entiéndase que LSA no es una herramienta ni un sistema, sino un tipo de técnica en la que se basa la arquitectura de algunos sistemas.

4.5.3.-LSA como técnica de categorización de documentos

El LSA/LSI fue originalmente descrito por Deerwester, Dumais, Furnas, Landauer y Harshman (1990) cómo un método de Recuperación de la Información (*Information Retrieval*). Fueron más tarde Landauer y Dumais (1996; 1997) los que propusieron este modelo cómo un modelo plausible de la adquisición y la representación del conocimiento. Desde ese momento ha sido empleada para modelar algunos fenómenos cognitivos (Landauer, 1999; Kintsch,1998; Kintsch, 2001; Kintsch y Bowles, 2002), además de aplicaciones más directas como son la corrección de textos en el ámbito académico (Trusso, 2005), para medidas de cohesión y coherencia textual (Graesser ,McNamara, Louwerse and Cai, 2004), para simular modelos de usuarios potenciales en usabilidad WEB (Blackmon, Polson, Kitajima, y Lewis, 2002; Blackmon y Mandalia, 2004; Jorge-Botana, 2006a y b) o como complemento a las ontologías (Cederberg y Widdows, 2003).

Para llevar a cabo la técnica, se procesa un texto de grandes dimensiones, lo que se conoce como el corpus lingüístico. El corpus se representa en una matriz cuyas filas contiene todos los términos distintos del corpus (palabras) y las columnas representen una ventana contextual en la que aparecen esos términos (habitualmente párrafos) (ver figura 4.6). De este modo, la matriz contiene sencillamente el número de veces que cada término aparece en un documento. Esta matriz sufre una ponderación que resta importancia a las palabras excesivamente frecuentes y la aumenta a las palabras moderadamente infrecuentes (Nakov, Popova y Mateev, 2001) con la idea de que las palabras demasiado frecuentes no sirven para discriminar bien la información importante del párrafo y las moderadamente infrecuentes sí. El siguiente paso es someter esta matriz ponderada a un algoritmo llamado Descomposición del Valor Singular (SVD), variante del análisis factorial (figura 37). El SVD se aplica con la idea de reducir el número de dimensiones de la matriz original en un número mucho más manejable (en torno a 300), sin que se pierda la información sustancial de la matriz original (Landauer y Dumais, 1997; Olde, Franceschetti, Karnavat y Graesser, 2002; Kurby, Wiemer-Hastings, Ganduri, Magliano, Millis y McNamara, 2003). No obstante, el número

de dimensiones depende de la naturaleza del corpus, por lo que puede ser muy variable y depender de varios criterios (Wild, Stahl, Stermsek y Neumann,2005)

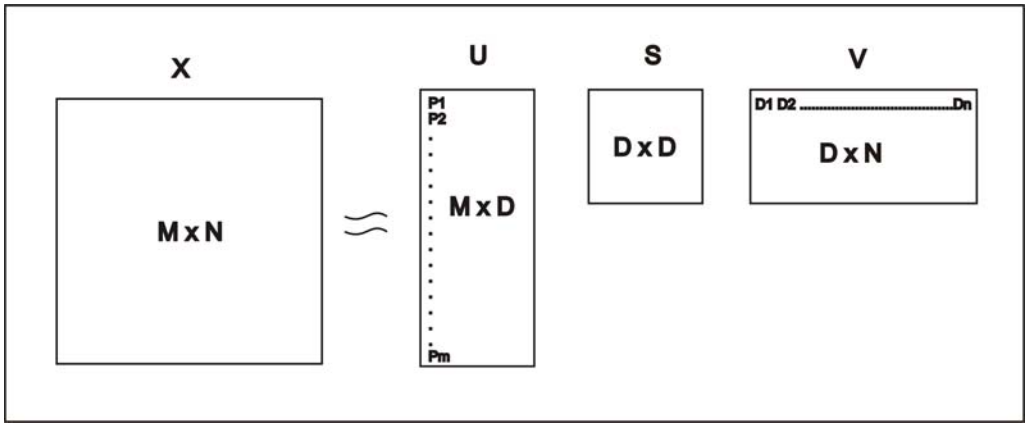


Figura 4.6.- Desglose de la matriz principal en las dos matrices de vectores singulares y una matriz diagonal de valores singulares. Será a partir de este desglose desde donde se reducirán las dimensiones tomando sólo las que más capacidad tienen para diferenciar regiones semánticas.

Lo interesante de esta reducción de dimensiones no es únicamente mejorar el manejo de una matriz tan grande como la original, sino crear un espacio semántico vectorial en el que tanto términos como documentos están representados por medio de vectores que contengan sólo la información sustancial para la formación de conceptos. La nueva representación de los términos y documentos en este espacio semántico ha mostrado ser muy exitosa simulando comportamientos humanos. La ventaja de representar el lenguaje vectorialmente es que éstos son susceptibles de comparaciones por medio de cosenos, distancias euclídeas u otras medida de similitud. Además, a partir de las coordenadas de los términos ya representados pueden introducirse en el espacio nuevos vectores que representen textos producidos a posteriori y que se suelen llamar pseudodocumentos (Landauer, Foltz y Laham, 1998). Será a partir de estos pseudodocumentos como se lleve a cabo la categorización de textos, transcripciones y diálogos[10]. En la medida en que se obtengan cosenos altos entre dos pseudocumentos, se podrá inferir que ambos versan sobre una temática similar. En nuestro caso, tendremos un vector con la verbalización del usuario y otro con el texto que represente una categoría en la

[10] Para familiarizarse con la técnica e incluso hacer algunas pruebas, puede visitarse el sitio LSA que mantiene, aunque algo desactualizado, la universidad de Boulder http://lsa.colorado.edu/ . También merece la pena echar un vistazo a http://www.cs.utk.edu/~lsi/. Además, en nuestro grupo de interés hemos creado un sitio en el que exponemos alguna documentación y se pueden realizar pruebas sobre algunos espacios semánticos específicos de dominio http://www.elsemantico.com/ .

línea de negocio de modo que cuando sean similares, se considerará que se refieren a los mismos contenidos y se dirigirá al usuario a los diálogos y acciones pertinentes.

4.5.4.-Algunos casos concretos de LSA y Call Routing

Una de las primeras experiencias en el uso de LSA en servicios de telefonía fue llevado a cabo en los laboratorios de *Lucent Technologies* por Chu-Carroll y Carpenter (1999). Tomaron como corpus de referencia 4.497 transcripciones telefónicas en las que los clientes interactuaban con los operadores de un "Call Center" de un servicio bancario. Analizaron primero las características de las trascripciones y, en especial, las primeras producciones verbales del cliente, de las cuales elaboraron una taxonomía según fuesen los datos aportados por él (1. - Nombre del destino - ej: "alguien en *leasing*, por favor"; 2. – Actividad -ej:"Querría hablar con alguien sobre cuentas de ahorro"; 3. - Demanda indirecta dónde se dan rodeos - ej: "Un amigo me dijo que si llamaba y había comprado un coche como él…"). Además, consignaron dónde suelen enrutan los operadores a los clientes dadas esas primeras demandas o producciones verbales. Siguiendo con su análisis del corpus, encontraron que el 20% de las llamadas necesitaban más información para ser desambiguadas y que requerían de mayor información. De estas llamadas que requerían más información, un 75% de ellas se produjeron como consecuencia de la falta de especificación en los nombres de las frases (e.g., "crédito para coche", sin especificar si se trata de un crédito de un coche que ya existe, o de un crédito para uno nuevo). El 25% restante de estas llamadas que requerían de una mayor información se produjeron por la poca especificación de los verbos (e.g., "depósito directo", sin especificar de si lo que desea es abrir un depósito o cambiar uno existente). Basado en esta falta de especificidad, Chu-Carroll y Carpenter (1999) introdujeron un módulo posterior que contenía LSA y con el que se solicitaba al usuario más información para desambiguar su demanda y de nuevo categorizar la reformulación de la demanda con el sistema LSA.

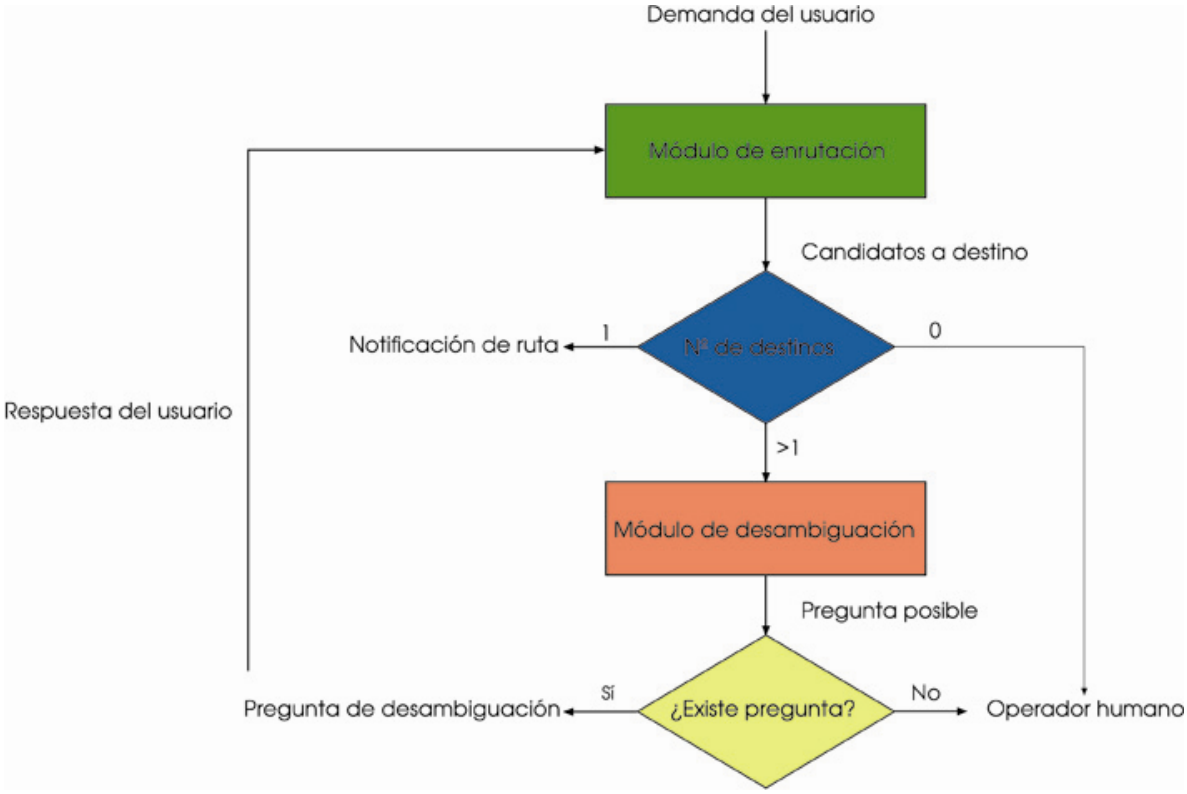


Figura 4.7.- Tomado de Carroll y Carpenter (1999)

En resumen, Chu-Carroll y Carpenter propusieron un sistema en que ante la respuesta del usuario a una pregunta abierta del tipo "Diga algo"(*Say Anything*), el módulo de LSA categorice dicha producción y proponga algunos candidatos de enrutación (si sólo hay uno se enruta directamente y si no hay ninguno se le pasa con un comercial). Si es el caso que hubiese diversos candidatos, el siguiente módulo, el de desambiguación, dada una respuesta de usuario concreta, formulará una pregunta para que sea el mismo usuario el que reformule su demanda. En la figura 4.7 se puede observarse la arquitectura de dicho sistema.

La forma de entrenar el módulo LSA es la siguiente: como cada demanda del usuario está marcada con el destino final donde fue dirigido, todas las frases que fueron dirigidas a un destino forman un documento único. De 3.753 llamadas se formarán 23 destinos que representarán cada uno un documento[11]. Antes de haberse formado estos documentos, se han suprimido del corpus la lista de palabras no deseadas como lo son las palabras más

[11] La técnica LSA parte de una matriz términos-documentos en la que se consignan las ocurrencias de los primeros en los segundos.

comunes o llamadas listas-stop (*stop list*) y las palabras que forman parte del ruido introducido en el lenguaje espontáneo (*fillers*) cuya lista se llama "lista a ignorar" (*ignore list*). Una particularidad del tratamiento del texto es la que viene dada por la formación de bigramas y trigramas. Los bigramas y trigramas son términos que por su uso conjunto forman una unidad. Ejemplos serían "*car+loan*", "*check+acount+balance*", etc. Los autores introducen una lista de estas palabras (si su ocurrencia en conjunción es significativa) para encontrar en el corpus la ocurrencia de estos términos y unirlos de manera que salgan en el corpus como un término indiferenciado. La forma de hacerlo es buscar las ocurrencias de los unigramas , bigramas y trigramas pero de manera que si es encontrada una de orden mayor como un trigrama, también son conservadas las de orden menor. Si por ejemplo, es encontrada "*check+acount+balance*", también serán introducidos en el corpus final los términos "*check+acount*", "*check*", "*acount*" y "*balance*". De esta manera se preservan las apariciones de los órdenes menores. Además, como habitualmente se hace, se calcula el corrector IDF[12]. Calculados todos los pasos de LSA incluido SVD y la reducción de dimensiones, obtenemos las matrices de consulta[13].

Obtenidas estas matrices de consulta y dada una demanda del usuario, es calculada su representación vectorial como pseudodocumento y es establecida la comparación con cada uno de los documentos destino. Los documentos con alta similitud bajo un umbral serán los candidatos que se introducirán en el módulo de desambiguación. Hay que resaltar que para el cálculo de la similitud entre destino y pseudodocumento emplean como medida los cosenos pero corregidos con el propósito de maximizar las diferencias. Para ello, emplean la función sigmoidea extraída de la distribución de las similitudes (medidas con el coseno) entre cada llamada y cada destino, dividido entre 1 si fue enrutado a ese destino o 0 si no lo fue. Dada esta distribución, se ajusta a una función logística y se obtiene la función concreta (índice de confianza) que

[12] IDF es uno de los métodos usados en la llamada fase de preproceso antes de someter la matriz a SVD. La motivación de este método es ponderar cada término en base a su importancia para representar supuestos dominios semánticos. Se infiere que si un término ocurre en un número muy alto de documentos será mal predictor del dominio al que puede pertenecer. Imagínese el lector un término como "dolor" en un corpus basado en una taxonomía médica. Este término no nos ofrecería gran información sobre tipos de enfermedades, no así por ejemplo "inmunodeficiencia". IDF trata de menguar el influjo de términos muy frecuentes y poco informativos como "dolor".
[13] Matrices de consulta: son las matrices que contienen la representación vectorial de términos y documentos.

modula los cosenos crudos.

$$Conf(d_a, d_b, x) = 1/(1 + e^{-(d_a x + d_b)})$$

Donde X es el coseno entre la demanda concreta del usuario y un destino concreto; d_a y d_b son coeficientes de la función sigmoidea para el vector destino.

Empleando la función sigmoidea se obtiene una reducción del error del 16.7% de llamadas bien dirigidas. Una vez obtenidos los índices de similitud se calcula empíricamente el umbral sobre el cual se considera que el destino es un posible candidato. Como resultado se obtiene que 0.2 representa el mejor umbral para el índice de confianza. Si más de un candidato supera dicho umbral se pone en marcha el módulo de desambiguación. El módulo de desambiguación intenta que dados dos destinos posibles, el usuario reformule la demanda para que uno de esos dos destinos se descarte. Para ello, hace uso de la filosofía de LSA en cuanto a la representación vectorial. Si el vector demanda está muy parejo a dos vectores destino, se habrán de encontrar términos que representen a los vectores-diferencia del vector-demanda con cada uno de los vectores-destino. De esta forma se comprueba qué términos son los que pueden ajustarse a esos vectores diferencia. Una vez encontrados estos, sólo servirán como candidatos los que pueden formar un n-grama con la demanda original. Es decir, para una demanda como “loans please”, y extrayéndose dos posibles destinos: “loan services” y “costumer lending”, se han de extraer los términos que se ajustan a los vectores diferencia entre “loans please” -“loan services” y “loan please” -“costumer lending” respectivamente. Dada la lista de términos que se ajustan a los vectores diferencia, serán sólo relevantes para la construcción de la nueva pregunta aquellos que pueden formar un n-grama con “loan”. De esto resultarán términos como “auto-loan” y “loan-payoff”. De aquí se crearán preguntas más o menos estándar a modo de moldes. En este ejemplo y dado que todos comparten “loan”, la pregunta propuesta será “*What kind of loan?*”. De esta manera, el usuario reformulará la demanda, volverá a pasar por el módulo de enrutamiento

y si es el caso de que hay más de dos destinos, volverá también al módulo de desambiguación y sufrirá el mismo proceso hasta ser refinada del todo. Si en el módulo de desambiguación, se pudiese formar sólo un n-grama como "exist+car+loan", entonces se propondría una pregunta del tipo "sí-no" como "Is this about an exiting car loan'". Este sistema consigue un porcentaje de aciertos del 93.8% e incluso tiene buenos resultados teniendo en cuenta los errores propios del reconocimiento del habla (97%-92.5% en su máximo rendimiento y 75%-72% en su rendimiento más modesto).

A partir de este anterior trabajo, Cox y Shahshani (2001) realizaron un análisis de la conveniencia o no de emplear LSA para "call routing". En primer lugar criticaron la forma en que Chu-Carroll y Carpenter (1999) formaron los documentos. Cox y Shahshani (2001) propusieron dos formas de crear los documentos en LSA y "call routing". La primera, la que ellos denominan *T-Route* y que es la empleada por Chu-Carroll y Carpenter (1999), y se basa en que un documento congrega todas las llamadas que fueron dirigidas a un mismo sitio. Compilando los documentos de esta manera, se crearán entonces tantas columnas o documentos como rutas posibles haya en nuestra lógica de negocio. La otra manera a la que llaman *T-Trans*, se basa en que cada trascripción configure en un documento por separado. De esta manera, habrá tantos documentos como llamadas. El problema de aplicar *T-Route* es que el número de columnas o documentos en (M, N) es muy pequeño, lo que hace que los términos queden representados con una dimensionalidad que puede ser muy baja (la dimensionalidad de los términos no podrá superar al número de columnas). Recordemos que la elección de N dimensiones para representar términos y documentos es arbitraria y viene dada por la propia técnica de Descomposición del Valor Singular (SVD). Otro problema de *T-Route* es el siguiente: el número de términos de un documento consulta o demanda suele ser muy reducido por lo que al formarse surgirá un vector (antes de introducirlo en el espacio vectorial) con la mayoría de sus índices ocupados por ceros. Esto contrasta con los documentos con los que se quieren comparar, los cuales, surgen de la compilación artificial de todas las llamadas que se enrutaron a un destino en un solo documento, lo que provoca que haya abundancia de valores no-cero e incluso valores abultados que muestran una sobre-representación.

Tomando medidas de porcentajes de error en varios espacios semánticos formados de diferentes formas (con *T-Route* o con *T-Trans*, Sin dimensionalizar, dimensionalizado con LSA o Análisis discriminante después de LSA), Cox y Shahshani (2001) encontraron que *T-Trans* obtuvo en todo momento menores porcentajes de error que *T-Route*. También descubrieron que los espacios no dimensionalizados obtenían mejores rendimientos que cuando se reducen las dimensiones, excepción hecha de la combinación que permite mejores resultados: dimensionar con LSA a 350 dimensiones y posteriormente empleando 31 dimensiones en el análisis discriminante. Estos resultados mostraban de alguna forma la aseveración de Cox y Shahshani (2001), quienes sostienen que en "call routing", el escenario varía frente a otras utilidades de LSA. Para "Call Routing" el número de dimensiones viene dado a priori y es igual al número de rutas posibles, por lo que puede ser innecesario descubrir cuál es la dimensionalidad óptima con LSA. Esto parece confirmarse con el buen comportamiento de los espacios sin dimensionar aunque puede deberse, tal como nosotros mismos hemos comprobado (Olmos, León, Escudero y Jorge-Botana, 2007; en prensa), a que no exista suficiente variabilidad en los corpus de referencia, es decir, que no estén representados en el corpus términos que representan información tangencial, lo cual no invalidaría la técnica LSA para corpus de diálogos telefónicos de mayor cobertura. Como concluyen Franceschetti, Karnavat, Marineau, McCallie, Olde, Terry y Graesser (2001), los corpus en los que se conserva esta información tangencial, funcionan mejor a la hora introducir comparaciones por medio de pseudodocumentos. En cualquier caso, el experimento de Cox y Shahshani (2001), muestra que aunque todos los espacios tienen un comportamiento aceptable (16%-6% de error), existen formas de LSA en combinación con otras técnicas que pueden resultar muy beneficiosas

Otra extensión interesante de LSA para clasificación de diálogos es la propuesta de Serafín y Di´Eugenio (2004). En su experiencia emplean FLSA (*Featured Latent Semantic Analysis*) para llevar a cabo clasificaciones de diálogos. Para ello prueban el comportamiento de tres corpus marcados previamente: *CallHome*, un corpus de llamadas telefónicas en español, *MapTask* que contiene diálogos en torno a las instrucciones en torno a un

mapa y *Diag-NLP* que versa sobre diálogos sobre el aprendizaje del uso de ordenadores. Todos estos corpus están marcados con etiquetas que aluden a varios criterios. El método FLSA computa dos matrices que luego concatena: la matriz términos documentos y la matriz etiquetas-documentos (véase la tabla 21 donde se resumen las ventajas de cada una de estas aportaciones).

	Ventajas	Desventajas
Chu-Carroll y Carpenter(1999)	-Módulo de desambiguación en base a n-gramas -Corrección de los cosenos en base a valores empíricos.	-Formación de documentos: Cada destino en la línea de negocio se considera un documento. Cada uno de los documentos lo forman todas las llamadas que fueron enrutadas a ese destino. Hay un reducido número de documentos los cuales resultan abultados artificialmente.
Cox y Shahshani(2001)	-Argumentación de la conveniencia o no de la reducción de dimensiones en el caso de enrutación de llamadas (call routing) -Rediseño de la forma de extraer documentos para ser representados en la matriz en relación a Chu-Carroll y Carpenter(1999). -Buenos resultados en combinatoria con otras técnicas estadísticas	-En la mayoría de los casos no encuentra mejores resultados con la reducción de dimensiones lo que puede deberse a varias causas (ver texto).
Serafín y Di´Eugenio(2004)	-Integración en el análisis de etiquetas dadas por jueces humanos. (*FLSA o Featured Latent Semantic Análisis*) -FLSA obtiene mejores resultados que LSA y que las líneas base. -También obtienen ventaja frente a resultados obtenidos anteriormente.	

Tabla 21: Resumen sucinto de las aportaciones y desventajas de las tres aproximaciones presentadas.

La matriz etiquetas-Documentos está formada por las etiquetas de los propios corpus e identifica a cada documento. Esta matriz resultante[14], (w+t)*D es tratada de la misma forma que se trataría en la forma clásica de LSA. De alguna manera, las etiquetas son tratadas como términos ocupando también filas en la matriz de datos. De esta forma se consigue forjar más cohesión entre los propios documentos y términos en coalición con las etiquetas artificiales.

[14] (w+t)*D se refiere a la concatenación de la matriz de palabras por documentos WxD y la matriz de etiquetas documentos TxD.

Ambos casos LSA y FLSA se comportan de una forma muy efectiva a la hora de categorizar diálogos pero los resultados muestran que FLSA se comporta algo mejor.

4.6.- Conclusiones

Se han presentado en este artículo algunas observaciones en torno al lugar que pueden ocupar las técnicas basadas en LSA en el diseño y desarrollo de agentes virtuales. LSA puede clasificar las demandas de los usuarios de un servicio telefónico en las categorías que identifican cada una de las líneas de negocio. En otras palabras, LSA emitirá un juicio de cuán parecido es el texto-demanda del usuario con cada uno de los documentos que representan las posibles rutas. Por tanto, las técnicas basadas en LSA pueden ser implementadas como herramientas en los módulos de Gestión del Diálogo. Esta parte de la aplicación empieza una vez se ha producido el reconocimiento del habla espontánea. Dada una entrada del usuario, LSA se encargará de clasificarlo en alguna de las categorías semánticas. Las técnicas basadas en espacios vectoriales tienen algunas ventajas sobre las técnicas clásicas de menús y tonos, a saber, dada la posibilidad de valorar entradas de discurso libre, ofrecen al usuario la flexibilidad de no seguir unas pautas marcadas exclusivamente por el sistema, evitando atravesar un excesivo número de menús . Otra ventaja es que toda interacción parte de una pregunta inicial del tipo “Diga algo” (“*Say Anything*”). De esta manera, el usuario percibe más naturalidad en los diálogos que mantiene. En el caso de querer reconocer la respuesta a una pregunta abierta, implementar tal sistema definiendo gramáticas, conllevaría un riesgo para el propio sistema de reconocimiento de voz o un excesivo número de ítems en los menús. Aún así, para llevar a cabo un sistema de enrutamiento basado en LSA es preciso conocer que tipo de materia prima estamos empleando por lo que serían de mucha utilidad más estudios sobre el tipo de corpus, su preproceso y la manera de representarlo en la matriz de ocurrencias como variables que inciden en la efectividad del enrutamiento. Por ejemplo, una cuestión importante es la que sugieren los resultados de Cox y Shahshani (2001) y es la relativa a si en todos los corpus o bajo todos los preprocesos y tratamientos es beneficiosa la reducción de

dimensiones o si por el contrario, hay ocasiones en que no es necesaria e incluso contraproducente. Por otro lado, el futuro de este tipo de técnicas pasa ahora por implementar algoritmos que empleen como base el espacio semántico que emana de LSA y que extraigan el sentido de estructuras concretas insertas en los textos. Un ejemplo de esto es el algoritmo de predicación (Kintsch, 2001) donde trata de encontrar el sentido a estructuras predicativas y metafóricas. En nuestro grupo de interés, estamos trabajando tanto en encontrar las propiedades de los corpus que elevan la eficiencia de LSA (Olmos et al,. 2007, 2007; en prensa; León, Olmos, Escudero, Cañas, Salmerón, 2006) como los parámetros involucrados en la representación de estructuras basadas en predicaciones (Jorge-Botana, Olmos, León y Molinero, 2007). En definitiva, el reto actual se encuentra en desarrollar parseadores que localicen cierto tipo de estructuras y aplicar sobre ellos algoritmos que operen sobre el espacio semántico resultante del proceso LSA.

Capítulo 5
La herramienta

El autor de este trabajo conoció a su más estrecho colaborador, Ricardo Olmos Albacete, hace ya algún tiempo, por mediación mi director de tesis, José Antonio León (el suyo también). Ricardo, al igual que yo, venía desarrollando en el ámbito de su tesis una herramienta capaz de llevar a cabo las funcionalidades de LSA con algunos métodos de parseo muy interesantes y complejos. En paralelo, yo también había desarrollado mi propia herramienta de tal manera que al encontrase nuestro caminos, ambos contábamos con nuestra propia herramienta y tuvimos la oportunidad de validar una con la otra, encontrando que se extraían valores muy similares cuando se comparaban semánticamente textos. Utilizamos una herramienta como criterio de la otra, obteniéndose correlaciones entre los resultados por encima siempre de 0,90. Por ejemplo, se obtuvieron 30 comparaciones semánticas de un corpus de psicología clínica procesado por ambas herramientas. La correlación entre los resultados fue de 0,96, ($p < 0,001$). Más tarde, ambos seguimos desarrollando sendas herramientas por separado con la esperanza de que en algún momento íbamos a poder integrar todo en una única aplicación. Al momento de la redacción de esta tesis, ese momento no ha llegado.

Por tanto, uno de los objetivos que hemos cumplido en esta tesis, es desarrollar el LSA de tal forma que se controle de principio a fin todos los procedimientos que requiere el LSA (entre otras controlar los métodos de ponderación, el número de dimensiones o las medidas de similitud). Tarde o temprano y de forma natural en una tesis que profundice en el LSA, uno termina creando su propio LSA. De no hacerlo, la opción que queda es la utilización de la herramienta estándar de Boulder, Colorado (véase en http://lsa.colorado.edu/). Sin embargo, el uso de la herramienta oficial impide la manipulación de variables esenciales para la comprensión y la profundización de la naturaleza del modelo LSA. Por ejemplo, entre las limitaciones más serias está la imposibilidad de entrenar los propios corpora, la imposibilidad de acceder a cada una de las coordenadas que conforma el vector de un término o texto y, tal vez, más importante, la imposibilidad de generar nuevos algoritmos que dinamicen la herramienta (al modo como hace Kintsch, 2001).

En conclusión, era condición “sine qua non” desarrollar enteramente el LSA. Como primer objetivo se planteó su diseño y programación, a poder ser con tecnología moderna y con paradigmas eficientes, como es el caso de la Programación Orientada a Objetos. Además, empezar con tecnología moderna y extendida podía facilitar en el futuro convertir con facilidad todos los desarrollos para el formato WEB y más aún, preparar servidores que proporcionen servicios WEB, bien sea por solicitudes http simples o por tecnologías que poco a poco están implantándose con fuerza, como es el caso de los protocolos SOAP. De esta forma, las clases implementadas en el desarrollo de esta tesis podían ser aprovechadas para posteriores desarrollos industriales. Es objeto de este capítulo de esta tesis decribir todo lo que concierne a la herramienta que fue implementada.

5.1.- Implementación de LSA con la plataforma .NET y MATLAB

Son muchas las opciones a la hora de elegir el lenguaje de programación y por ende, la plataforma de desarrollo que se empleará para llevar a cabo una aplicación que materialice (LSA). Incluso, como no pasa desapercibido al lector de estas líneas, en ciertas ocasiones, esto es dependiente del conocimiento que se tenga sobre unas plataformas de desarrollo y no otras. Se podrán elegir desarrollos con lenguajes orientados a objetos y de ámbito general como es Java, C++, VB.NET, C# , J# , Python, hasta lenguajes circunscritos a ámbitos más restringidos al cálculo como R, MATLAB, etc. Cada una de estas opciones tiene sus ventajas e inconvenientes y generalmente, estas vienen impuestas por el desarrollo de nuevas funciones y librerías dentro del marco de programación. Por ejemplo, si bien es verdad que en torno a –R-, está surgiendo una comunidad muy activa y comprometida en el desarrollo de funciones de código abierto, también es verdad que Java y .Net son opciones más solventes y nutridas dentro del mundo empresarial y que permiten una fusión mucho más directa con aplicaciones de bases de datos como ORACLE y SQL SERVER. Una buena reflexión sobre lenguajes y sistemas operativos en torno a LSA la podemos encontrar en Quesada (2006). En lo que se refiere a este apartado, nos vamos a dedicar a uno de los marcos de trabajo más

extendidos en el mundo empresarial y quizás, menos tenido en cuenta a la hora de implementar desarrollos de estas características en el mundo académico: .NET

5.1.1.- Plataforma .NET

Visual Studio .NET es una plataforma para el desarrollo de aplicaciones Windows y aplicaciones WEB. Es un entorno de trabajo que no va unido a ningún lenguaje específico, sino que sirve de recurso a un abanico amplio de lenguajes. Todos los .Net utilizarán las mismas herramientas, independientemente del lenguaje elegido, con el objeto de ofrecer la máxima funcionalidad a la aplicación en forma de código. La plataforma con la que se ha desarrollado el proyecto ha sido *Visual Studio .NET 2003*, constituida en su base por cuatro lenguajes de programación:

- Visual Basic .NET 2003
- Visual C# .NET 2003
- Visual C++ .NET 2003
- Visual J# .NET 2003

5.1.2.- Librería de clases de Álgebra Lineal. Trabajando con matrices.

Para implementar el procedimiento estándar SVD y los cálculos que se realizan sobre las diferentes matrices es necesario contar con alguna librería suplementaria de cálculo de álgebra lineal. Generalmente, estas funciones están disponibles en paquetes especializados como MATLAB o sueltos en lenguajes compartidos y comunitarios como R, pero no se suelen encontrar en los lenguajes generales como un procedimiento incorporado al entorno de trabajo. Dada la implantación de este tipo de entorno de trabajo en el mercado (tomemos por caso VISUAL STUDIO.NET). Algunas firmas han optado por dotar a los desarrolladores de estos lenguajes de librerías de clases

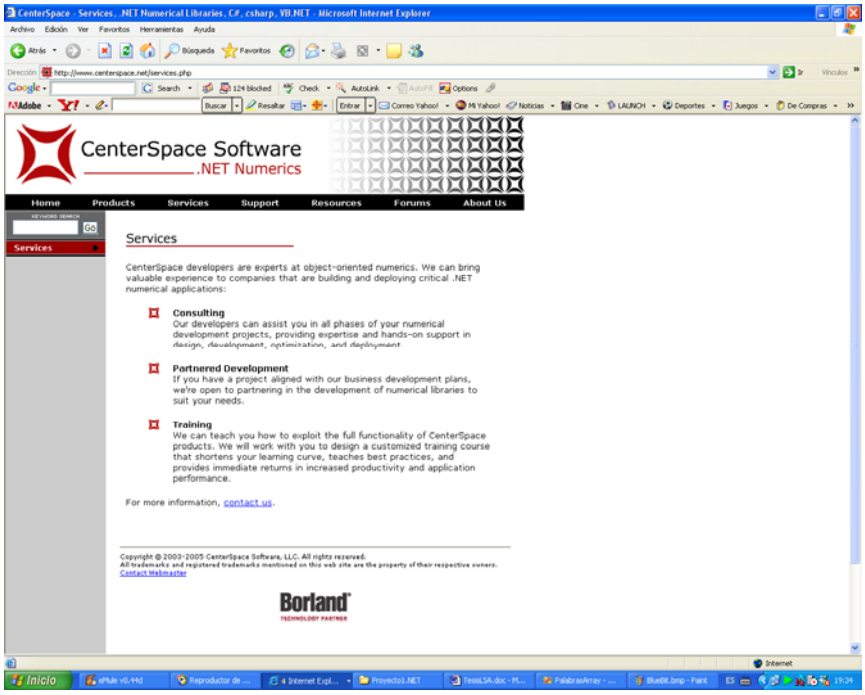
relacionadas con los cálculos de álgebra lineal. Incluso algunas compañías han optado por hacer que los desarrolladores .NET puedan incluir en sus proyectos código con un lenguaje totalmente orientado a las ciencias como es el caso de FORTRAN. Tal es el caso de -*Fortran for .NET Language System*- de Lahey/Fujitsu (http://www.lahey.com/lf71/lfnet.htm) **o** PVF que integra FORTRAN en la nueva versión de visual estudio 2005 **(**The Portland Group, http://www.pgroup.com)**.** En otras palabras, al igual que los componentes que se le añaden al entorno general cuando se agregan nuevas referencia y se hacen disponibles nuevos controles o librerías, también aquí, se importaría una nueva librería de clases de cálculo de álgebra lineal sobre matrices y se agregaría a nuestro proyecto.

En nuestro caso concreto, necesitamos generar matrices de gran tamaño y calcular sobre ellas toda la funcionalidad que ha sido expuesta en apartados anteriores: producto escalar, multiplicación de matrices, sumatorios, SVD, etc., y necesitamos hacerlo dentro de un entorno en que sea fácil implementar siguiendo la filosofía de la Programación Orientada a Objetos, lo cual nos facilitará diseñar en forma de clases todas las entidades que vamos a ir necesitando, a saber, pseudodocumentos, términos, redes con sus nodos y conexiones capaces de implementar algoritmos como el de predicación, etc. Además, esto ayudará a hacer que el código sea reutilizable y que las correcciones se leven a cabo sólo en las clases implicadas. Además, podemos hacer uso de propiedades como la agregación por las cuales por ejemplo, un pseudodocumento puede contener varios objetos de tipo término que a su vez contenga entre otros objetos, uno de tipo vector.

Para la plataforma .NET hemos encontrado dos fabricantes que tienen entre sus ofertas este tipo de productos. Uno de los fabricantes es -*Center Space Software-,* cuyas librerías pueden encontrarse en la página http://www.centerspace.net. La librería que se ajusta a nuestro propósito lleva el nombre de *NMath Matrix* y se puede adquirir en la misma página WEB. Nosotros personalmente no la hemos utilizado ya que optamos por la segunda opción que ahora presentamos. Esta decisión fue tomada en base al buen soporte técnico que presta y al interés de sus foros. Se trata de un fabricante

griego que se dedica también a dotar de componentes a los desarrolladores. El nombre de la casa es *BlueBit Software* (puede consultarse en su página http://www.bluebit.gr).

Para álgebra lineal en .NET puede adquirirse el paquete *.NET Matrix Library 2.2* .A día de hoy está disponible la versión 5, que implementa "*Sparse Matrices*". En ambos fabricantes existen versiones de prueba y en este último además nosotros adquirimos una versión académica de un año de duración en la cual no está limitada ninguna de sus funcionalidades. También existe una librería para los desarrolladores que requieran agregar la referencia como componente *COM: Matrix ActiveX Component Version 3.1.*



Otra alternativa para el desarrollo de este tipo de aplicación es contar con librerías que han sido desarrollados en el marco de proyectos de código abierto. La desventaja de este tipo de librerías es que no están avaladas por una licencia que contenga una garantía de buen funcionamiento lo que viene especificado en la propia licencia. Además, los procedimientos de clases suelen ser menos y con un menor desarrollo que las que se comercializan en el mercado.

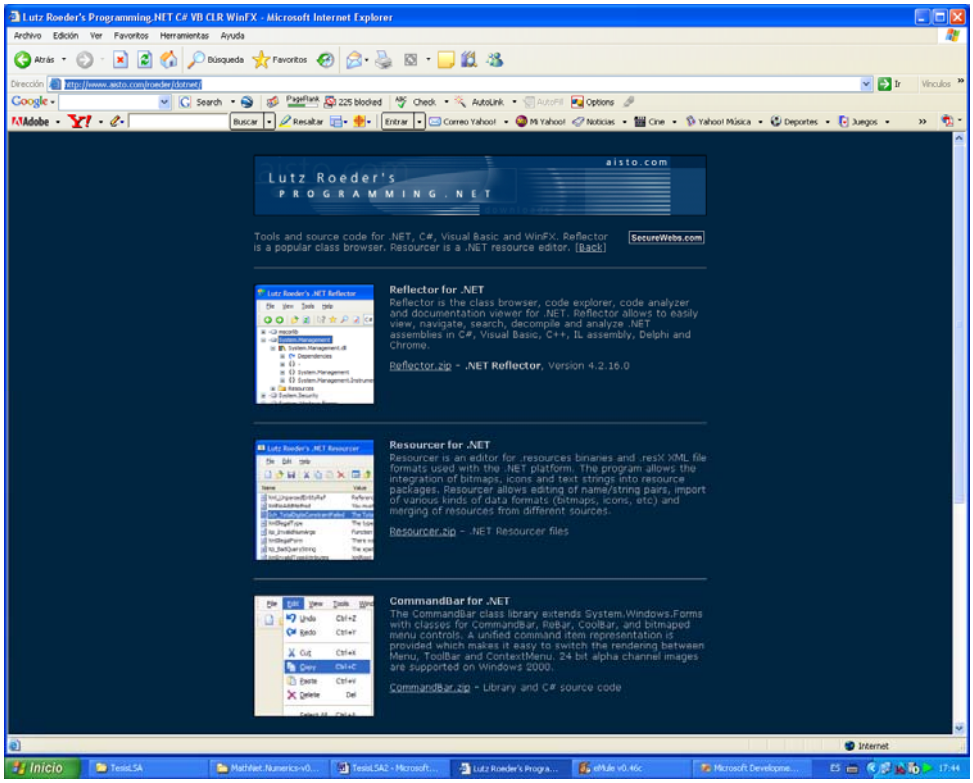
Un ejemplo de este particular es el desarrollo dentro de la clase -Matrix- de una clase vector en el paquete .NET Matrix Library 2.2 (de http://www.bluebit.gr) y su facilidad para convertir vectores en matrices y viceversa además de multitud de métodos que facilitan el desarrollo. Sin embargo, una gran ventaja de este tipo de librerías es que suelen venir acompañadas del código fuente de las clases que contienen. De esta manera, los procedimientos de las clases que involucremos en el desarrollo de las aplicaciones, pueden ser manipulados por nosotros mismos, si es el caso de que necesitamos variaciones en su funcionamiento. Para la plataforma .net, hemos encontrado algunos proyectos que contienen librerías de álgebra lineal de código abierto.

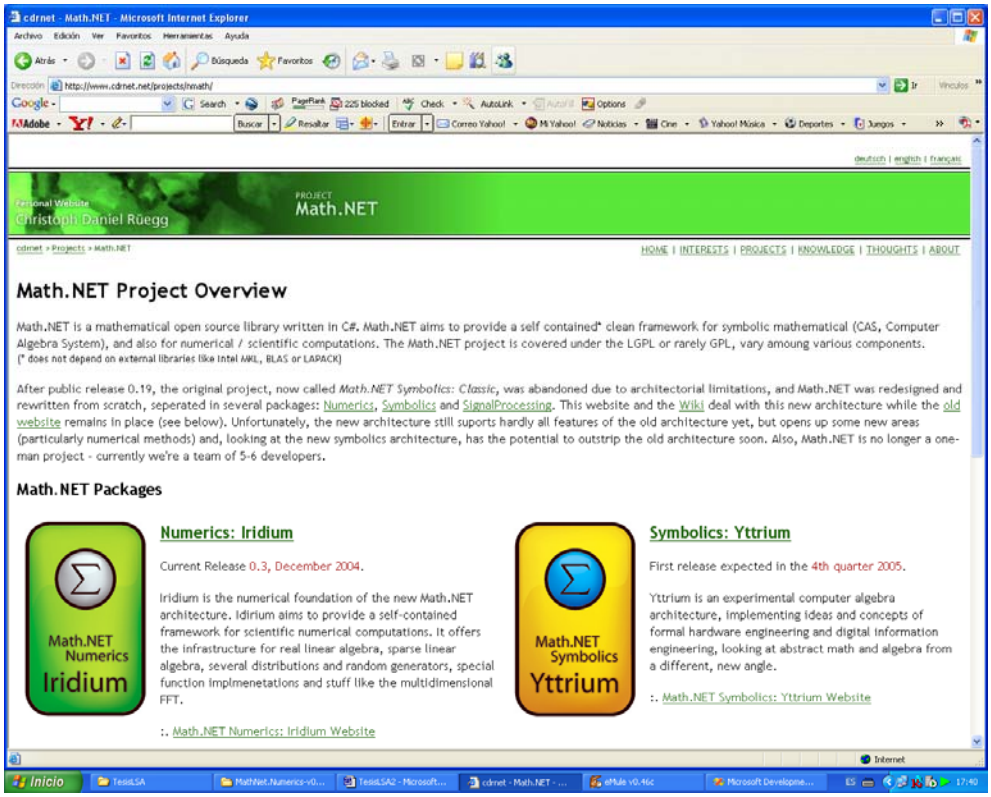
Resaltamos aquí las que nos parecen más interesantes bajo el criterio de que contengan el desarrollo de la -descomposición del valor singular- (SVD) y que sean fácilmente incorporables a los proyectos que se codifican con lenguajes VB.NET y C#. Ambas permiten descargar un archivo en formato .Zip que contiene tanto las librerías (dll) como las clases (.vb) y algún ejemplo. Mapack for .Net (www.lutzroeder.com/dotnet/) es una adaptación al entorno .net de las librerías -Mapack for COM-, -Lapack- y -Java Matrix Package-. No es muy amplia en lo que se refiere al número de las clases y los métodos pero si es de fácil uso. Las librerías (.dll) son acompañadas por el código fuente y un ejemplo sencillo.

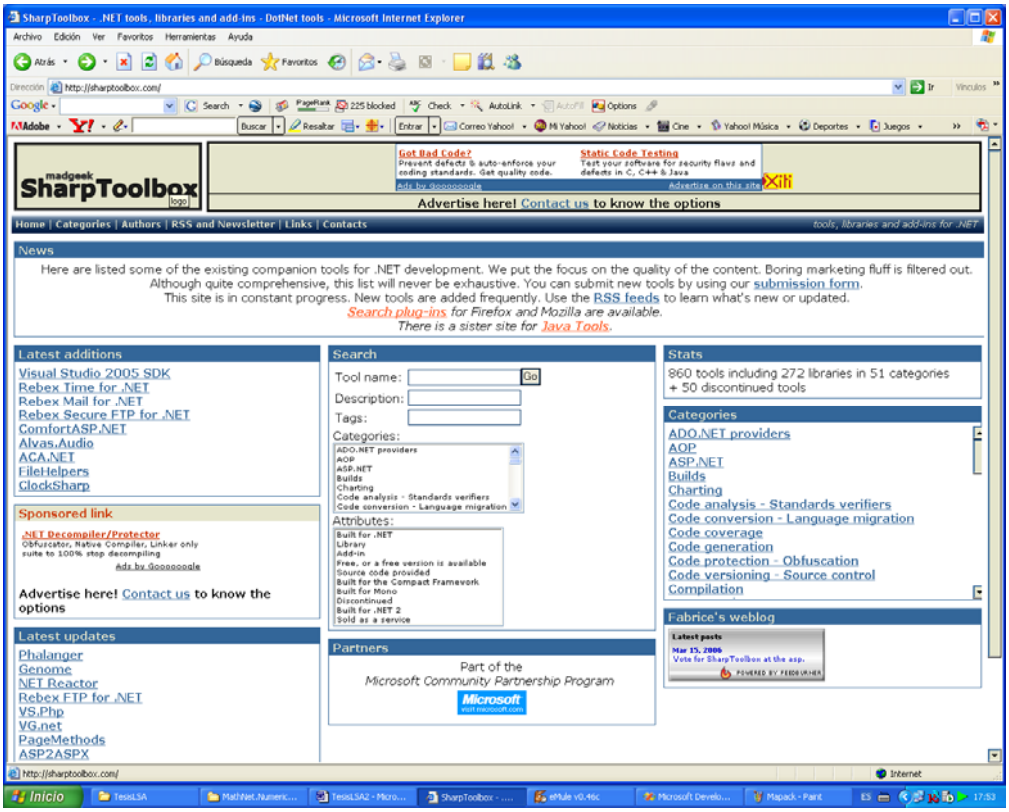
El desarrollador responde al nombre de Lutz Roedor. Aunque contiene las funcionalidades básicas para operar sobre matrices(multiplicaciones, sustracciones, submatrices, etc.) y contiene la clase SVD (descomposición del valor singular) acompañada de su código, no contiene demasiadas funcionalidades como por ejemplo la capacidad de aislar directamente los vectores que contienen las matrices instanciadas y calcular sobre ellos por ejemplo el producto escalar (útil por ejemplo para la comparación de similitud entre términos, documentos y pseudodocumentos).

Aunque no llega a las librerías comerciales, un desarrollo más extenso es el proyecto Math.NET Iridium (www.mathdotnet.com/Iridium.aspx). Su autor es el suizo Christoph Rüegg e implementa un mayor número de clases y funcionalidades que el anterior. Además de contener las funcionalidades básicas del cálculo de matrices y de contener la clase SVD (descomposición del valor singular) con su código fuente, su ventaja sobre otras es que contiene a su vez otras clases cuyas funcionalidades van desde la trasformación de Fourier hasta el tratamiento de distribuciones probabilísticas. El desarrollo de estas librerías están en continuo desarrollo por lo que es probable que a lo largo del tiempo, acojan muchas más funcionalidades. Como fuente de recursos .net para desarrollar aplicaciones que contengan cálculos de diversos tipo, (recomendamos también http://sharptoolbox.com/Category3f9caa41-7171-4d11-be8e-b79ef7532830.aspx en el que se podrá encontrar un gran número de

recursos relacionados con implementación .net en torno a redes neuronales artificiales, cálculos matemáticos y estadísticos, inteligencia artificial, etc).







Mención aparte merece el sitio WEB referencia de los desarrolladores tanto C#, VB, ASP.NET o C++ de la plataforma .net. Se trata de la página http://www.codeproject.com que aunque es de propósito general, se pueden encontrar multitud de librerías de clase que pueden ser utilizadas para multitud de proyectos en los que se necesite desde funciones de álgebra lineal hasta la implementación de redes neurales artificiales. Está dividida en multitud de apartados dependiendo del lenguaje de programación(C#, VB, C++, ASP.NET) y del propósito del desarrollador (algoritmos, librerías, acceso a datos, etc.). Un ejemplo de esto son las siguientes:

- http://www.codeproject.com/useritems/brainnet.asp: librerías y código en VB.net para implementar redes neurales artificiales con diseños orientados a objetos (Madhusudanan, 2006). En el estudio de esta librería nos hemos basado para construir la red necesaria para implementar el algoritmo de predicación.

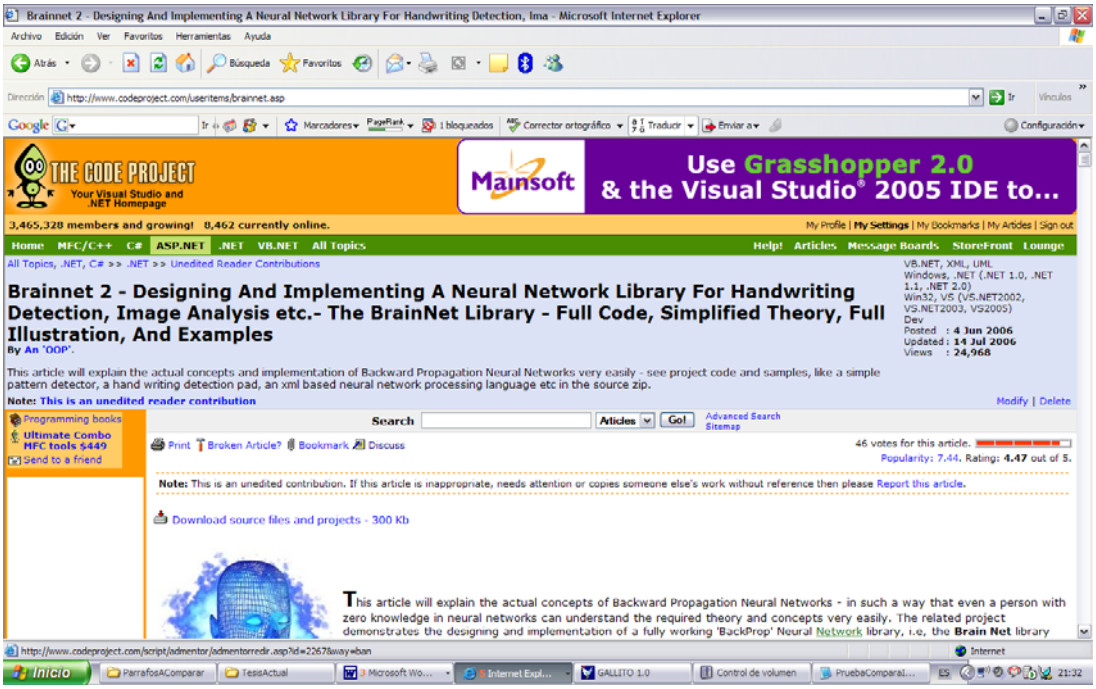
- http://www.codeproject.com/useritems/HopfieldNeuralNetwork.asp: librerías para implementar una red neural tipo Hopfield (Magomedov, 2006).

- http://www.codeproject.com/useritems/nxml.asp : Se explica el uso de NXML, un sistema estándar de marcado basado en XML para guardar formalizar cualquier aplicación de redes neurales. Se explica su uso y se propone una apliación POO y sus clases que emplea este tipo de formalización (Madhusudanan, 2006)

- http://www.codeproject.com/vb/net/algebraclass.asp :librerías y código (en VB) de álgebra lineal (Vander Haeghen Y,2003).

- http://www.codeproject.com/csharp/psdotnetmatrix.asp : librerías y código(C#) de álgebra lineal (Selormey, P.,2004).

- http://www.codeproject.com/samples/matlab_cpp.asp : librería y código (Matlab/C++) de álgebra lineal (Riazi, A.,2003).



5.1.3.- Nuestra solución

Para beneficiarse de la de la Programación Orientada a Objetos se empleó la plataforma .NET (VB.NET y en ocasiones C#), entre otra cosa para implementar una buena interfaz de usuario. Además, para realizar algunos cálculos se empleo la librería ofrecida por BlueBit Software (http://www.bluebit.gr). Referente al SVD, se empleó el ofrecido por MATLAB, integrando dicha función en la plataforma .NET (véase la página http://www.codeproject.com/KB/dotnet/matlabeng.aspx). Esta última librería permite ejecutar funciones de *Matlab* desde el código de .NET y el intercambio de vectores y matrices entre la aplicación principal implementada en .NET y las ejecuciones en MATLAB. Esta solución nos proporcionó poder aprovechar las funcionalidades de las matrices *sparse* de *matlab*, pero implementadas con la filosofía de Orientación a Objetos en una plataforma de carácter general como Visual Studio .NET.

5.2.- Funcionalidades principales a resolver

5.2.1.- El tratamiento del texto

Cuando decimos tratamiento de texto nos referimos a la parte que concierne a la parte que la aplicación que se encarga de suprimir ciertas estructuras que no interesa que estén representadas en la matriz de coocurrencias. Estas estructuras pueden ser desde signos básicos hasta estructuras sintagmáticas más complejas. Lo primero de todo es depurar el texto y dejarlo sin signos que puedan contaminar la formalización del texto en la matriz de coocurrencias. El producto final sería un archivo nuevo que no contuviese ni uno solo de los signos no deseados y dispuesto a ser procesado en busca de estructuras gramaticales no deseadas. La siguente fase es eliminar del corpus todas aquellas estructuras léxico-gramaticales que contaminen de alguna forma la extracción de la semántica. En apartados anteriores se analiza que tipo de estructuras conviene eliminar del análisis. Un ejemplo claro es el representado por los determinantes y adverbios. Si deseamos quitar este tipo de estructuras, lo primero es definir exactamente las ocurrencias que queremos suprimir. Para ello podemos crear una o varias listas de las palabras y estructuras que queremos eliminar. Puede haber varias listas según queramos que se deje opción a eliminar unas y otras no. Estas listas pueden estar insertas en ficheros de texto para ser leídas y utilizadas por la propia aplicación. A su vez hay que insertar en archivos diferentes las estructuras formadas por una sola palabra y las que están formadas por dos o más palabras. Esto es debido a que en un primer escaneado, habrá que eliminar las estructuras complejas para más tarde retirar las simples. Por ejemplo, la estructura “a las primeras de cambio” está compuesta por palabras que también son susceptibles de eliminar como son “a", "las”, “de”, y quizás “primeras”. Si eliminàramos primero estas estructuras parciales no seríamos capaces después de encontrar este tipo de estructuras complejas y eliminarlas del corpus con éxito. Por eso, es imprescindible seguir este orden. Un ejemplo de una parte de un archivo de estructuras complejas sería el siguiente:

adverbioscomplejos.txt

a largo plazo
a las claras
a las mil maravillas
a las órdenes
a las primeras de cambio
a las puertas de la muerte
a las veces
a la antigua
a la carrera
a la chita callando
a la cola
a la defensiva
a la desbandada
a la desesperada
a la funerala
a la hora
a la intemperie
a la inversa
a la izquierda
.

Respecto a las estructuras simples, una parte ejemplo podría ser la siguiente:

Adverbios.txt

.
jamás
tampoco
Acaso
quizás
tal vez
probablemente
Aquí
entonces
ahora
así
luego

tal
tanto
Donde
como
cuanto
cuando
Cuándo
dónde
cómo
cuánto
qué
palabra
abdominalmente
aberrantemente
.

Una vez se tienen las listas en archivo de texto de las palabras que se eliminarán, hay que crear una rutina que abra esos archivos, los deposite en arrays y haga un escaneado en busca de la ocurrencia de cada una de las estructuras contenidas en el array. No hay que olvidar que hay que seguir estrictamente el orden de más a menos complejas. Una forma de suprimir las estructuras complejas puede ser cogiendo frase a frase o párrafo a párrafo el corpus (dependiendo de que es lo que hayamos tomado como separador de documentos) y suprimir en ese subtexto almacenado en una variable cadena cada una de las estructuras que contiene el array.

Cuando se han cumplido todos estos pasos, el producto final es un fichero de texto que contiene el corpus originario pero sin ninguna de las estructuras no deseadas. Este es el fichero sobre el que se construirá la matriz de coocurrencias.

5.2.2.- La matriz de coocurrencias

La matriz de coocurrencias ha de representar exclusivamente las ocurrencias de cada uno de los términos del corpus ya depurado en cada uno de los documentos. Por lo tanto se implementará un algoritmo para que recorra

cada uno de los documentos y contabilice la ocurrencia de cada uno de los términos. Reconocerá una sucesión de caracteres como documento por medio del signo que hayamos escogido como delimitador. Si la unidad de documento fuese la frase, el delimitador coincidiría con el punto, si fuese el párrafo, lo más normal es delimitarlo artificialmente introduciendo como separador un símbolo (por ejemplo "#"). Así las cosas, primero rellenaremos un array con todos y cada uno de los documentos teniendo en cuenta su delimitador. A su vez, en esta misma rutina se introduce una función para cargar un array que contiene los términos únicos. Una rutina que toma como argumento cada uno de los documentos contenidos en el array de documentos va introduciendo en el array de términos todas las palabras que no estén ya contenidas en él. Es decir, divide cada documento en palabras e introduce en el array sólo aquellas que aún no estén en él.

5.2.3.- Eliminación de términos que no aparecen en n documentos

Una de las recomendaciones para reducir ruido del análisis es eliminar desde un primer momento los términos que no salen en un determinado número de documentos. Con esto, también reduciremos el tamaño de la matriz lo que reducirá el uso de la memoria. Con estas tres rutinas obtendríamos un array de términos y otro de documentos que son los que utilizaríamos para rellenar y dar forma a la matriz de coocurrencias.

5.2.4.- El ajuste lingüístico

Se llevarán a cabo los cálculos que posibilitan calcular cada celda en base a la importancia que se estima al término para predecir un tema. Se implementarán dos tipos de cálculos opcionales. Log-Entropía y Log-IDF. Las explicaciones y las fórmulas están insertas en el capítulo 2, en el que se explica la técnica de LSA.

5.2.5.- SVD

Después de tener la matriz de ocurrencias y si ha sido el caso,

ponderada conforme a las fórmulas de la entropía, puede ya llevarse a cabo la descomposición de dicha matriz en vectores y valores singulares singulares. Simplemente recordar que la SVD (singular value decomposition) devuelve un desglose de tres matrices. Dos de ellas son los vectores filas y columnas independientes a su vez de la interacción de la matriz. La otra es una matriz diagonal de valores singulares en orden descendente según la participación que cada valor tenga en interacción de la matriz. Para realizar este cálculo se pasará la matriz de ocurrencias a una rutina en Matlab y se recuperarán de nuevo las tres matrices resultantes a la aplicación principal.

5.2.6- Consultas

Una vez tenemos la matriz factorizada, podemos empezar a realizar consultas sobre ella. Las consultas más importantes pueden ser la comparación término con término, documento con documento y comparación documento con pseudodocumento. Todos los sistemas basados en LSA tienen como base este tipo de consultas. Este tipo de comparaciones se llevarán a cabo por medio de los cosenos u opcionalmente por medio de las distancias euclídeas.

5.3. La herramienta

5.3.1. Introducción

Las aplicaciones funcionales asociadas a este tipo de sistemas pueden ser:

- Creación de espacios semánticos.
- Recopilador de palabras insertas en documentos definidos.
- Cálculo de funciones de importancia de términos (Entropía o IDF)

5.3.2. Instalación

5.3.2.1. Requisitos

Para el correcto funcionamiento del Gallito 1.0 es necesario tener instalados los siguientes componentes, (Algunas se adjuntan en el CD de instalación o pueden descargarse desde la página web de Microsoft):

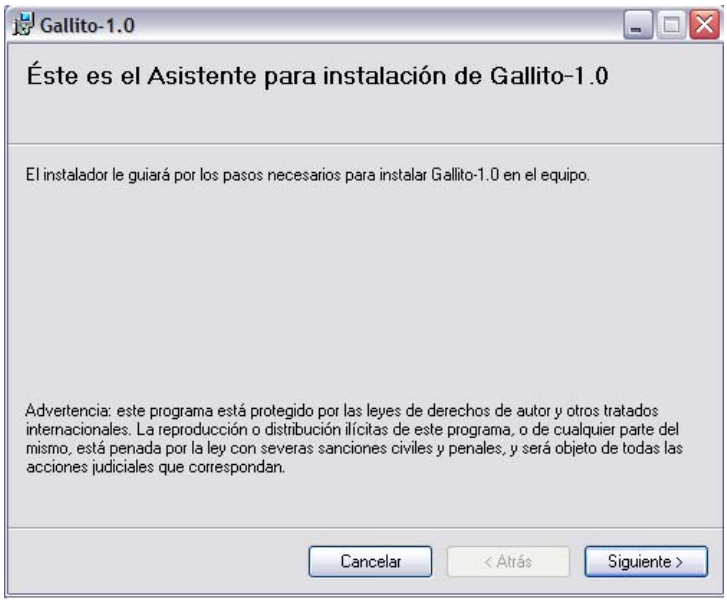
- Microsoft SDK 1.1
- MATLAB 6

5.3.2.2. Procedimientos de instalación

- Acceder al directorio y ejecutar el programa Setup.exe.
- Aparecerán las siguientes pantallas que le guiarán en el proceso de

instalación:

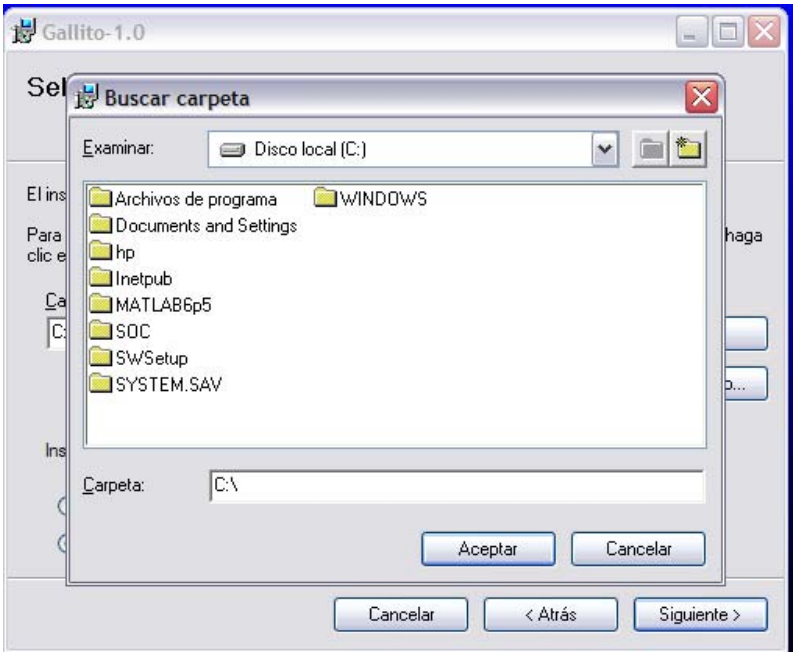
Primer paso. Pantalla de bienvenida a la instalación del producto, en la que se especifica el producto a instalar (Gallito 1.0). Deberá pulsar el botón siguiente para continuar con la instalación o el botón cancelar para salir del proceso de instalación.



Segundo paso. En esta pantalla debe seleccionarse el directorio para la instalación del producto que por defecto será C:\Archivos de Programa\Guill\Gallito-1.0\

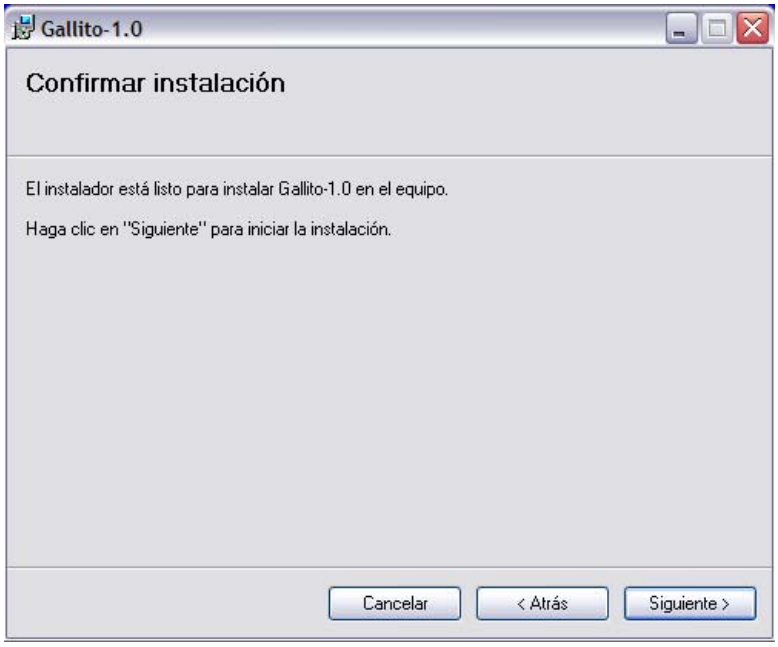
En caso de que se desee instalar el producto en otro directorio diferente al directorio por defecto este deberá ser seleccionado pulsando el botón “Examinar” que aparece en la ventana de instalación. Pulsando este botón se accede a una ventana de exploración estándar de directorios de Windows a través de la cual se podrá

seleccionar el directorio deseado para la instalación de Gallito 1.0 (ver siguiente imagen).



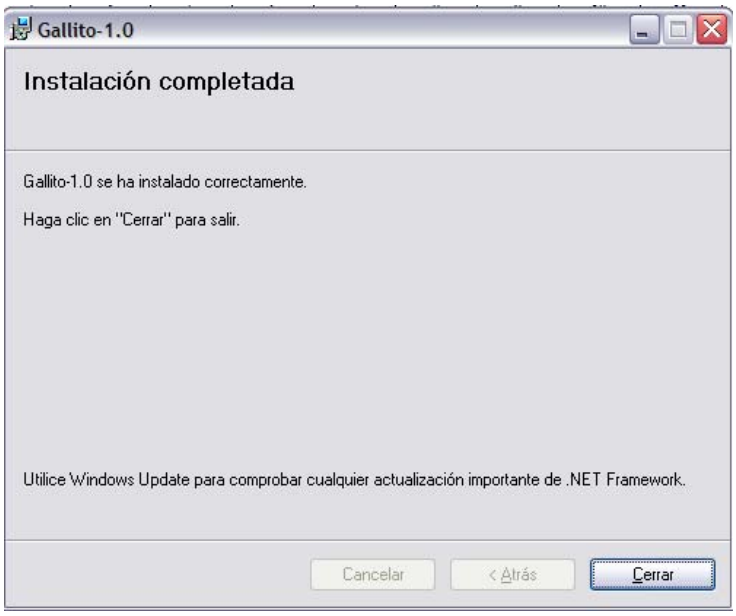
Una vez seleccionado el directorio para la instalación deberá pulsarse el botón "siguiente" para continuar con el proceso de instalación.

Tercer paso. En este paso se avisa al usuario de que pulsando el botón "siguiente" se comenzará la instalación del producto. Pulsando el botón "atrás" se podrá volver a introducir la información recabada hasta el momento para realizar la instalación.



Cuarto paso. La ventana mostrada indica el progreso de la instalación en tanto por ciento y los archivos que están siendo copiados en la máquina en la que se está instalando el producto. Una vez alcanzado el 100% del proceso de instalación se accederá al siguiente paso de la misma (ver siguiente epígrafe) tras informarse de que ha sido introducida en el registro la información necesaria para el correcto funcionamiento de la aplicación. Pulsando el botón cancelar de la ventana de progreso se detiene el proceso de instalación.

Quinto Paso. Fin de la instalación. En esta pantalla se avisa de la finalización la instalación del producto. Pulsando el botón terminar se accede a la última pantalla del proceso de instalación.



5.3.2.3. Componentes instalados

La aplicación permite:

- Generar espacios semánticos bajo diferentes parámetros.
- Consultar la semejanza de términos, documento y pseudodocumentos.
- Generar listados de vecinos bajo diferentes parámetros (con posibilidad de ser exportados a Microsoft Excel).
- Cargar y guardar espacios semánticos en diferentes formatos.
- Generar ficheros .txt con las matrices importantes.

Para ello la aplicación constará de una única pantalla o panel de control gestionada con pestañas y pequeñas subpantallas. La aplicación se divide en dos funcionalidades básicas:

- Generación de espacio semántico.
- Operaciones sobre un espacio semántico (Calcular Semejanzas, Guardar espacio, Cargar espacio, etc.).

5.3.3. Funcionalidades

Al iniciar la aplicación se desplegará una pequeña presentación seguida de la pantalla de control. Paralelo a esto, podemos observar que se abre el editor de MatLab cuya ventana quedará disponible en la parte inferior. No es necesario manipular la pantalla del editor de MatLab para realizar ninguna operación, ¡Por favor, no cierre este editor!. A partir de aquí, Ud. Podrá optar por crear un nuevo espacio semántico o cargar un existente que esté guardado en el disco duro. La pantalla que nos aparecerá será la siguiente:

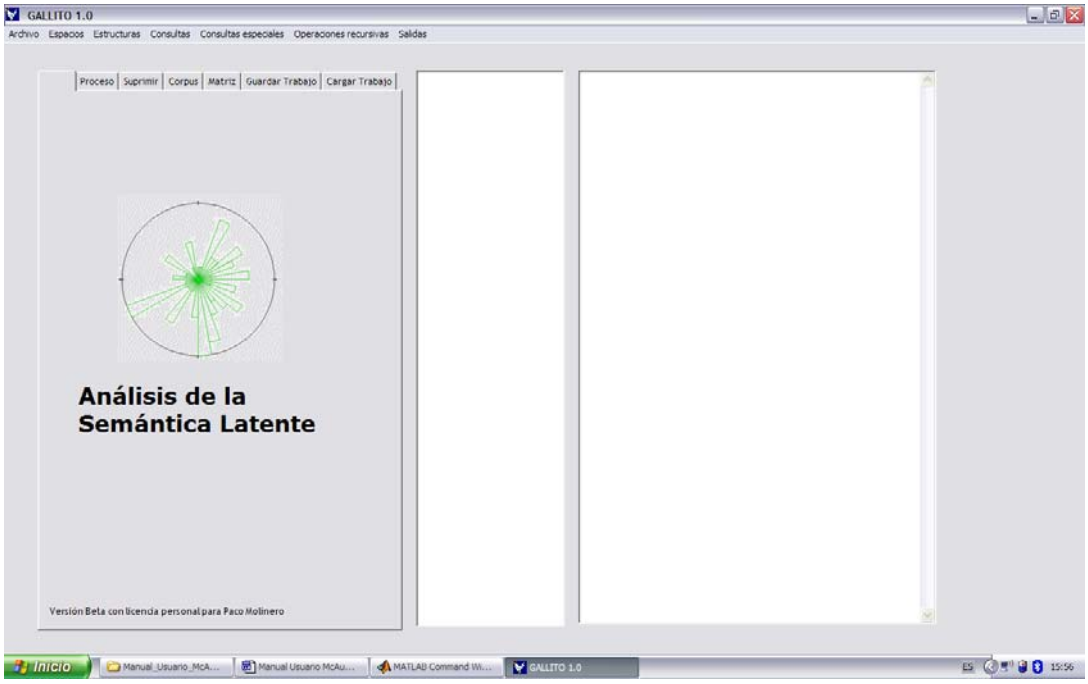


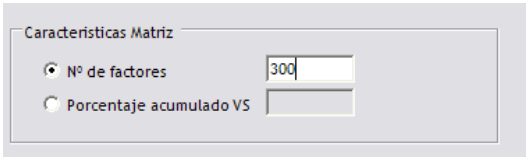
Figura 5.14.-

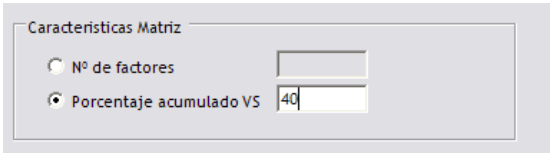
Con las pestañas y desplegables se podrá empezar a realizar operaciones. Para ello la pantalla estará dividida en tres zonas donde se tendrán los siguientes elementos:

5.3.3.1. Crear un espacio semántico

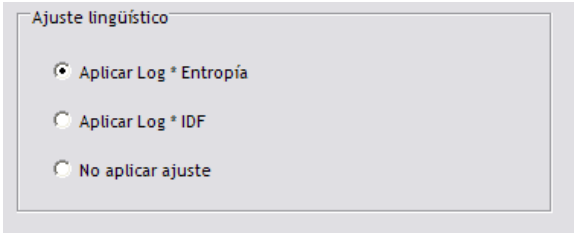
Para crear un espacio semántico se necesitarán los siguientes parámetros:

- **Nº de dimensiones o Valor singular acumulado**: El nº de dimensiones no debe superar el nº total de documentos. Respecto al valor singular acumulado se expresará en porcentaje (sin el carácter "%"). Este porcentaje reflejará la dimensionalidad conservada, es decir, el porcentaje de dimensionalidad (en orden descendente de importancia) que se conservará. De esta manera, un 40% se referirá a un nº de dimensiones que corresponderá con esa dimensionalidad. En corpus extremadamente grandes no será posible el cálculo de dicho porcentaje por lo que se emplearán 300 dimensiones.



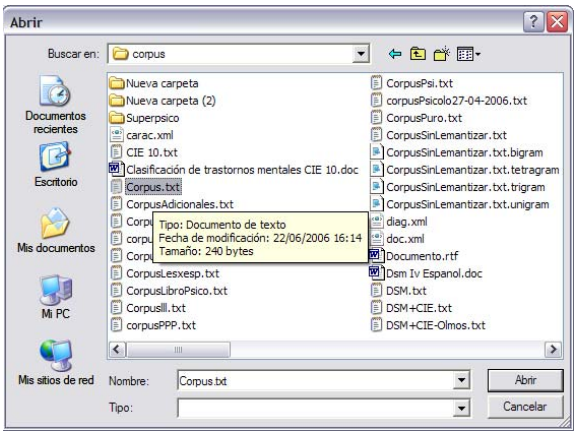


- **Ajuste lingüístico**: Esta opcionalidad se referirá al cálculo de importancia de cada término en el corpus. Podrá seleccionarse Entropía o IDF. También la ausencia de estos cálculos.

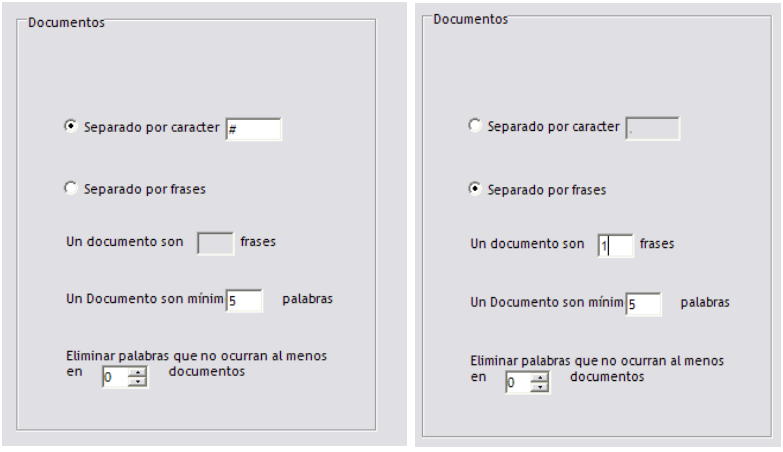


- **Corpus de referencia**: Ruta del archivo de texto en donde se encuentra el corpus lingüístico (en un formato legítimo) Pulse el botón para "examinar" los directorios.

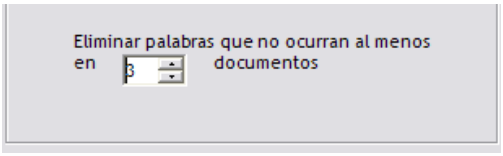
En ventana de exploración estándar de directorios de Windows podrá seleccionar el corpus deseado.



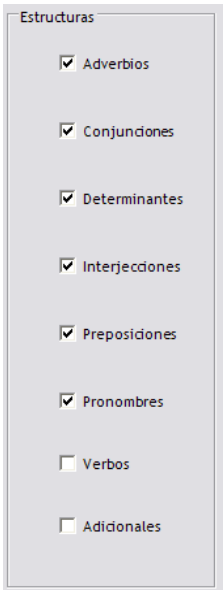
- **Separación de los documentos**: Los documentos podrán estar separados por un carácter o simplemente por frases naturales. En el primer caso habrá que especificar el carácter separador (generalmente "#"). En el segundo caso habrá que configurar cuantas frases forman un documento (generalmente 1).



- **“Un documento son mínimo ...”**: Cual es el número mínimo de términos para que un documento sea introducido en el análisis.
- **“eliminar palabras que no ocurran al menos en ...”**: Cual es el número mínimo de documentos en el que un término concreto tiene que aparecer para ser incluido en el análisis.

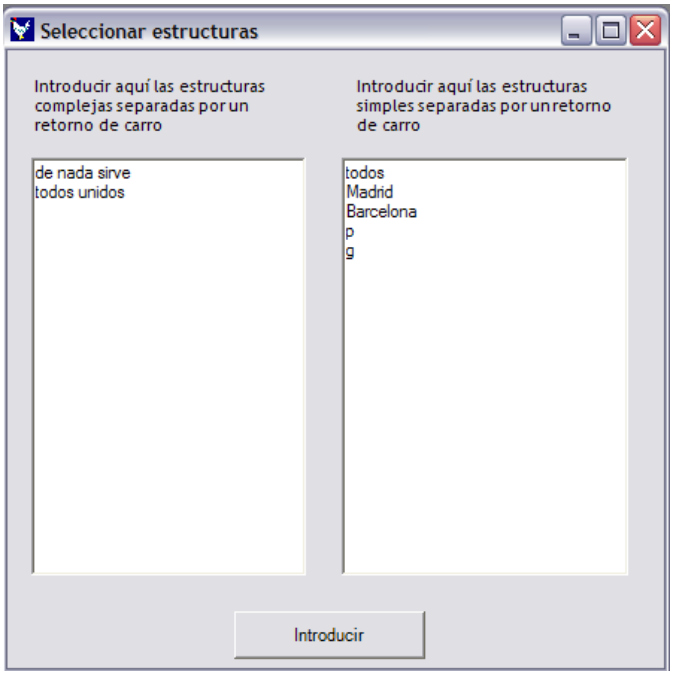


- **Suprimir**: Se suprimen las apariciones literales de cada una de las estructuras propuestas.

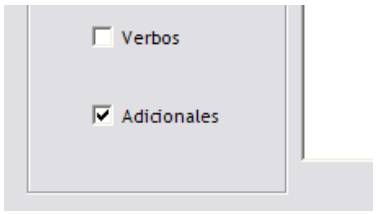


- **Generar “stop list”:**

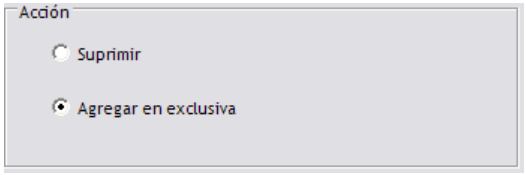
  1) Seleccione en los desplegables estructuras > elegir e introduzca las estructuras que desea eliminar. En la parte izquierda las estructuras compuestas por más de un término. En la derecha, las estructuras simples o compuestas por un solo término.



2) Seleccione en la pestaña de suprimir la casilla “Adicionales”

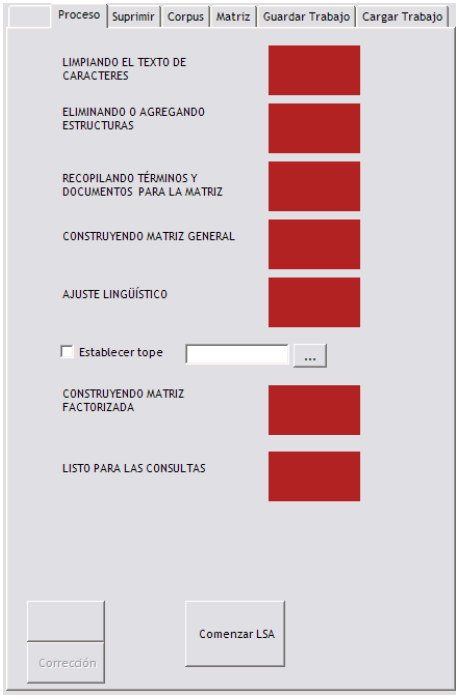


- Crear un espacio sólo con una “go list” : El procedimiento es semejante al anterior. Se selecciona en la pestaña de suprimir la casilla “Adicionales” y se introducen en estructuras > elegir las estructuras de la “lista pase”. También ha de habilitarse la opción “Agregar en exclusiva” .

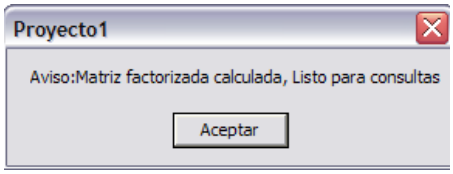


Mediante este método se pueden seleccionar las demás estructuras para que formen parte de análisis, es decir, un análisis con palabras de función y adverbios por ejemplo y las estructuras adicionales seleccionadas.

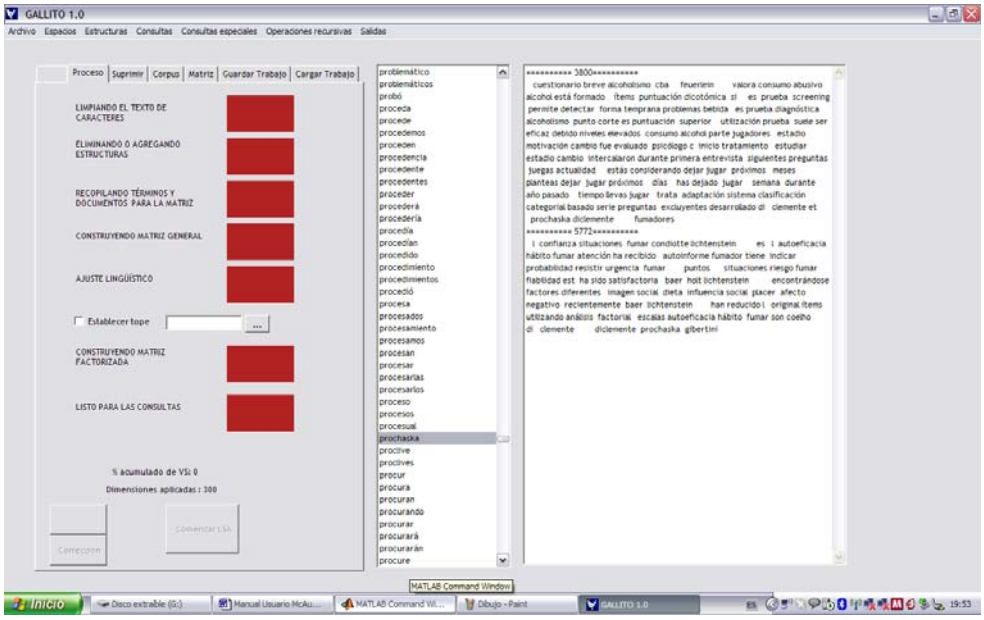
Una vez seleccionados los parámetros para crear el corpus se abrirá la pestaña de “Proceso” y se pinchara en el botón de “Comenzar”. A raíz de esto, los procesos se mostrarán encendidos conforme se vayan ejecutando.



El proceso final acabará con el siguiente mensaje:



Aceptando este aviso, términos y documentos se cargarán en la parte de la derecha y se podrá proceder a realizar operaciones sobre el espacio semántico.



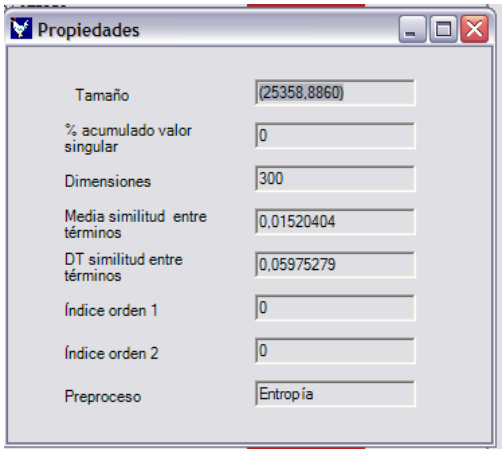
5.3.3.2. Operación sobre el espacio semántico

Una vez cargado creado un espacio semántico, se podrá proceder a realizar operaciones sobre él. Estas operaciones pueden ser: Comparar dos términos, Comparar dos documentos identificados por un número, comparar dos textos libres introducidos por el usuario, extraer vecindarios semánticos (con cosenos simples o corregidos, o con predicación simple o corregida). También se podrá guardar en espacio en un directorio del disco duro para ser cargado en otra ocasión.

5.3.3.2.1. Propiedades del espacio

En esta opcionalidad se podrán consultar las propiedades de los espacios sobre los que se trabaja. Algunos de los índices estará deshabilitados en las aplicaciones destinadas a grandes corpus lingüísticos.

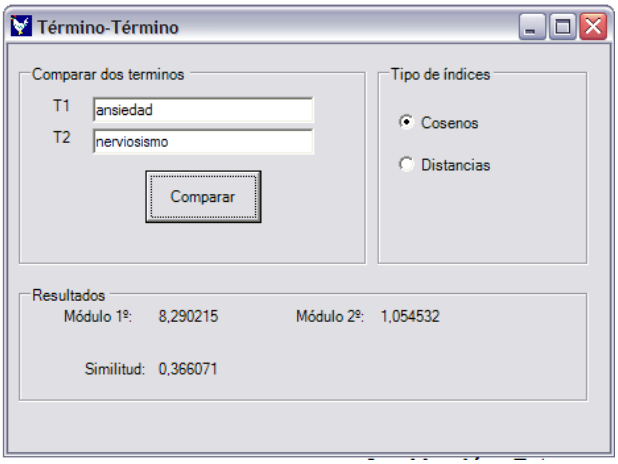
Espacio>Propiedades



5.3.3.2.2. Comparar dos términos

Se podrán comparar mediante el coseno o la distancia euclídea dos términos concretos.

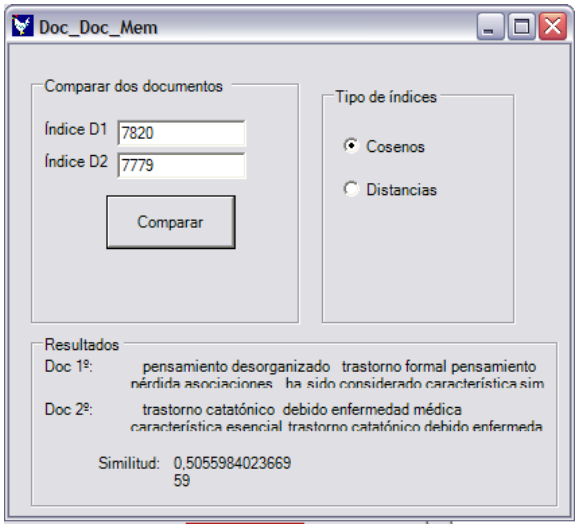
Consultas > Término-Término



5.3.3.2.3. Comparar dos documentos en base a su índice

Se podrán comparar mediante el coseno o la distancia euclídea dos documentos concretos.

Consultas > Documento-Documento



5.3.3.2.4. Comparar dos textos libres

Se podrán comparar mediante el coseno o la distancia euclídea dos textos libres (En espacios de gran tamaño este proceso tardará algunos segundos. El proceso será indicado mediante una barra de estado).

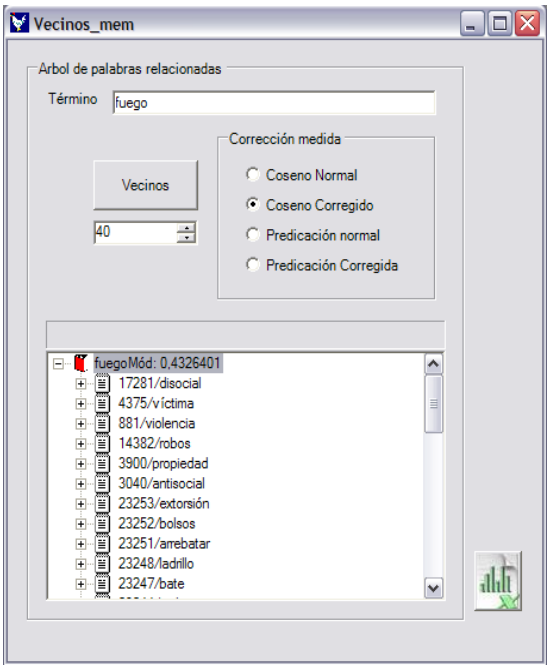
Consultas > Textos libres

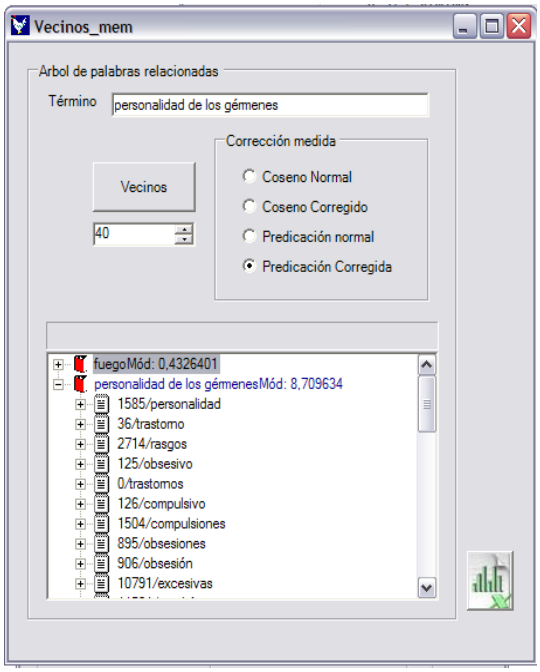
5.3.3.2.5. Extracción del vecindario semántico

Se extraer mediante diferentes métodos (cosenos, cosenos corregidos, predicación, predicación corregida) los vecinos semánticos de un término concreto. Se desplegarán árboles de vecindario según se extraigan vecinos de los términos.

Además, se podrá seleccionar el número de vecinos a extraer. También se podrán exportar los resultados a Microsoft Excel.

Consultas > Vecindario semántico

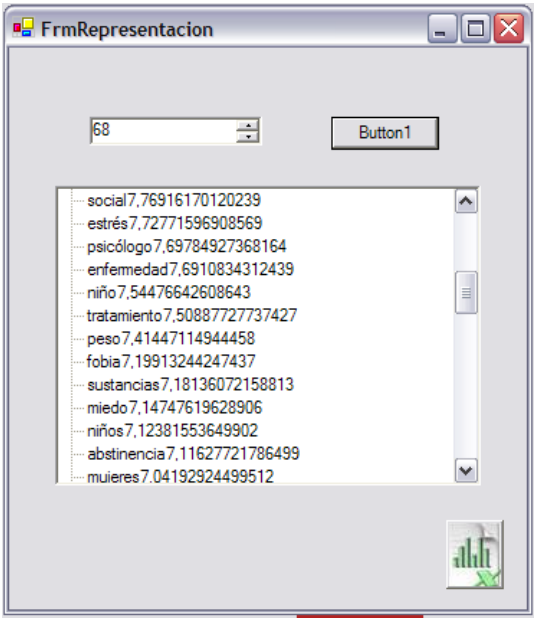




5.3.3.2.6. Extracción de los términos más representativos del espacio semántico:

Se extraer mediante diferentes métodos los vecinos semánticos de un término concreto.

Consultas > Representación

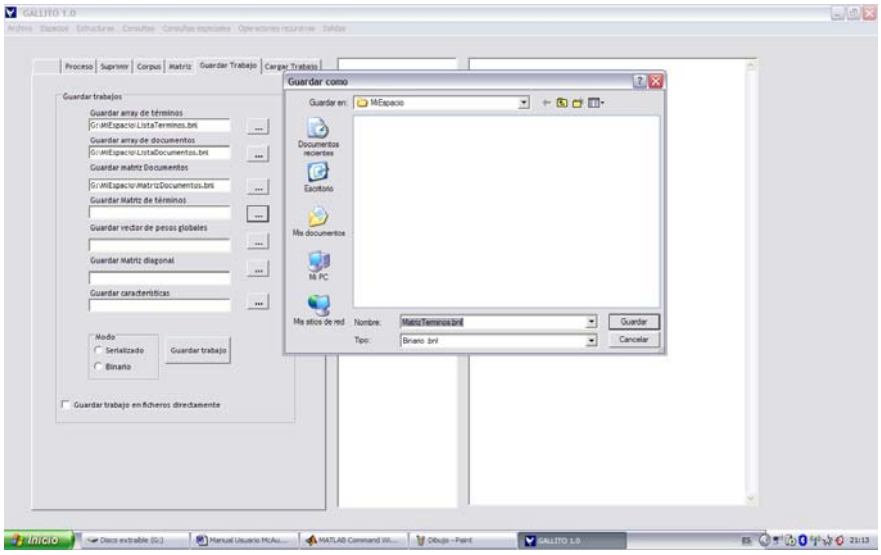


5.3.3.3. Guardar el espacio semántico

Un espacio semántico puede guardarse en el disco duro y volverse a cargar cuando lo necesitemos sin necesidad de volverlo a crear.

Pestaña "guardar trabajo"

A continuación le pediremos que seleccione un nombre y una ruta para guardar 7 variables. Recomendamos que guarde todos en un mismo directorio que podrá crear en la misma ventana de exploración de directorios de cada una de las variables. Al finalizar, seleccione el formato (recomendamos "binario") y pinche el botón de "Guardar".

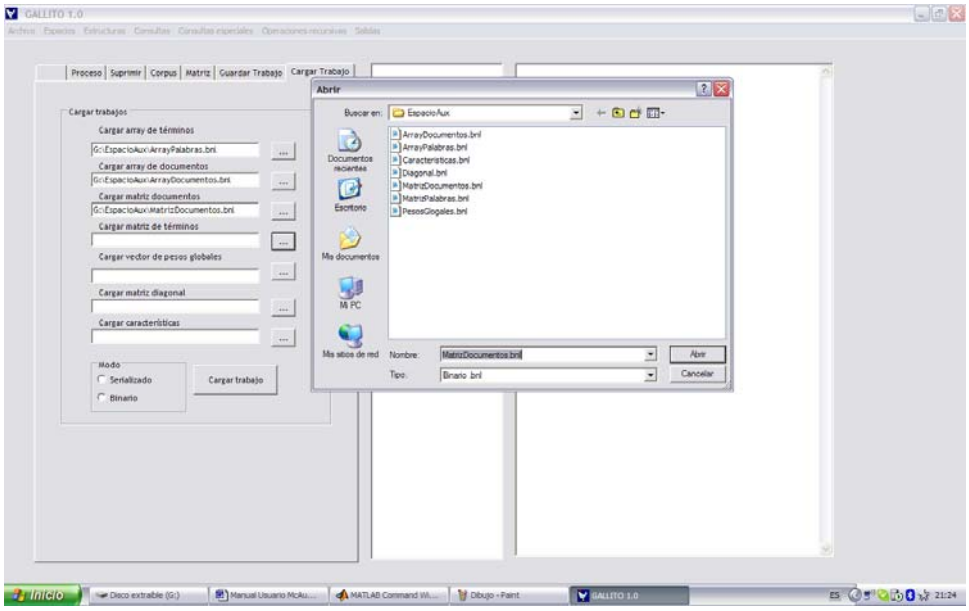


5.3.3.4. Cargar un espacio semántico

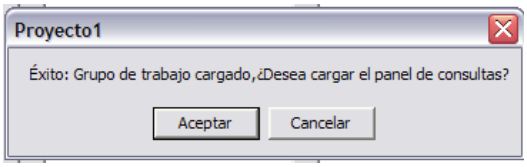
De la misma manera, un espacio semántico puede cargarse desde el disco duro sin necesidad de volverlo a crear.

Pestaña “Cargar trabajo”

A continuación le pediremos que seleccione un nombre y una ruta para localizar las 7 variables del espacio a cargar. En la misma ventana de exploración de directorios de cada una de las variables, puede buscar y seleccionar la ruta de la variable correspondiente. Al finalizar, seleccione el formato con el que se guardó (recomendamos “binario”).



Finalizado esto, pinche el botón de “Cargar” y espere a que salga el siguiente mensaje.



Seleccione aceptar y el espacio será cargado en memoria para realizar las operaciones antes descritas.

Capítulo 6

Objetivos del presente trabajo de tesis

6.1.- Objetivos básicos de esta tesis

El objetivo general de esta tesis ha sido analizar de manera exhaustiva tanto teórica como aplicada la herramienta informática del Análisis Semántico Latente (*Latent Semantic Analysis*, LSA, en inglés) y abordar su alcance de esta herramienta para simular diversos aspectos psicológicos en lengua española, superando algunas limitaciones como es el problema de la polisemia. Si bien este fue siempre el objetivo general de esta tesis, para desarrollarlo nos obligaba a partir de otros tres objetivos previos y más básicos. El primero de ellos fue crear un LSA propio (con una herramienta propia). Mediante la creación de este sistema basado en LSA, hemos podido generar espacios semánticos en español, extraer la similitud de los términos y textos de estos espacios y llevar a cabo ciertos procedimientos sobre ellos como, por ejemplo, el algoritmo de predicación Kintsch (2001) o simular correctores de respuestas abiertas. El desarrollo del LSA nos ha permitido controlar de principio a fin todos los procedimientos que se requieren para su buen funcionamiento, procedimientos entre los que se encuentran el control de los métodos de ponderación, el número de dimensiones o las medidas de similitud, por poner algunos ejemplos. El desarrollo de un LSA propio se hace, por tanto, un paso obligado para profundizar en el uso de la herramienta y simular procesos cognitivos humanos ya que, de no hacerlo, la opción alternativa que se hubiese optado hubiese sido la utilización de la herramienta estándar de Boulder, Colorado (véase en http://lsa.colorado.edu/). Como es sabido, el uso de esta herramienta oficial impide la manipulación de variables esenciales para la comprensión y la profundización de la naturaleza del modelo LSA. Entre las limitaciones más serias se encuentra la imposibilidad de entrenar los propios corpora, la dificultad de acceder a cada una de las coordenadas que conforma el vector de un término o texto y, tal vez, más importante, la imposibilidad de generar nuevos algoritmos que dinamicen la herramienta (al modo como lo ha realizado Kintsch, 2001).

En resumen, este objetivo previo era una condición *sine qua non* desarrollar enteramente el LSA. Se planteó su diseño y programación, optando por una tecnología moderna y con paradigmas eficientes, como es el caso de la

Programación Orientada a Objetos (POO), o matrices SPARSE para el procesado de los corpus. Este objetivo conlleva también un “paseo” exhaustivo por la técnica analizando, además, cómo ha sido desarrollada por otros investigadores y cómo éstos han resuelto algunos problemas relativos a su puesta en marcha como son la selección de los corpus, la eliminación de ciertas estructuras léxicas y núcleos temáticos, el número de dimensiones, el tipo de medidas de similitud vectorial que han empleado, la creación artificial de corpus, etc. Además, el hecho de empezar a desarrollarla con una tecnología moderna y extendida, podría facilitar en el futuro convertir con facilidad todos los desarrollos para el formato WEB y más aún, preparar servidores que proporcionen servicios WEB (Web Services), bien sea por solicitudes http simples o por tecnologías que poco a poco están implantándose con fuerza, como es el caso de los protocolos SOAP (*Simple Object Access Protocol)*.

Un segundo objetivo básico de esta tesis se ha dirigido a recopilar los fenómenos psicológicos que han podido modelarse con cierto éxito. Se propone una reflexión sobre cómo se justifican los modelos en psicología cognitiva y cómo se contrastan con los datos empíricos. Para cumplir este objetivo se enumerarán también los fenómenos a los que LSA ha podido servir como modelo plausible. Estos son: pobreza de estímulo, captación de las relaciones de distintos órdenes, sinonimia y antonimia, polisemia y homonimia, la representación única del término, comprensión de predicaciones, comprensión de metáforas, intentos de introducción de morfología y sintaxis, isomorfismo de segundo orden y la evaluación de resúmenes (emulación del juicio de un experto). Al final se analizan también algunas limitaciones.

El tercer y último objetivo básico es describir las aplicaciones prácticas que se han sustentado en la técnica del LSA. El autor de esta tesis lleva dedicando su carrera profesional al ámbito de las tecnologías donde es jefe de proyecto en una consultora dedicada a llevar a cabo proyectos tecnológicos relacionados con los sistemas IVR (*Interactive Voice Response*). Dichos sistemas necesitan para su desarrollo, además de tecnología de carácter general, desarrollo de herramientas lingüísticas como es el caso de gramáticas para los reconocedores de voz (*grxml, gsl o abnf*) y estándares para la creación

de flujos de diálogos, como es el caso de VXML (VoiceXML). Además, poco a poco, el mercado está haciendo más patente la necesidad de herramientas de tratamiento semántico para estimar las demandas de los usuarios de un call-center y llevarles donde éstas se despachan o simplemente para ponderar los índices de confianza de las salidas de los reconocedores de voz. Por esta causa, me es muy grato hacer un repaso sobre algunos casos de éxito, como es la creación de autotutores, las aplicaciones que simulan modelos de usuario en aras a perfeccionar la búsqueda y un uso eficiente (usabilidad) de las aplicaciones, de la detección de estados de ánimo y la implementación de enrutadores en sistemas de gestión de diálogos en el ámbito de la IVR (*Interactive Voice Response*). Por ello, el alcance de esta tesis podría también servir de base sobre posibles y posteriores desarrollos industriales o tecnológicos.

6.2.- Objetivos específicos de esta tesis

Sobre la base de estos tres objetivos básicos se han desarrollado otros objetivos específicos del presente trabajo de tesis y que se pueden desglosar en cuatro bloques, coincidiendo con los cuatro estudios experimentales de esta tesis. Aunque los objetivos operativizados y las hipótesis de trabajo se detallan en cada trabajo experimental, aprovechamos aquí para hacer una breve descripción de la secuencia que se ha seguido:

6.2.1. - Parámetros en torno a LSA y la evaluación de resúmenes

En primer lugar y como objetivo de largo alcance, se intentó acotar la incomensurabilidad con la que se ha de afrontar la manipulación del LSA. En este sentido, el primer trabajo experimental se trató de evaluar los parámetros con los que el LSA se mostraba más eficiente. Este trabajo se realizó con corpus específicos de dominio y, por lo tanto, con espacios semánticos de reducidas dimensiones. Concretamente, se trata de corpus basados en informaciones diagnósticas construidos de diferentes formas.

Previo a este trabajo y mediante un análisis informal de nuestro grupo de trabajo, se había puesto de manifiesto el hecho de que con corpus pequeños, la reducción de dimensionalidad no siempre aporta beneficios mayores que el no hacerlo. Esta conclusión fue clave para comprender el funcionamiento de LSA, ya que se plantea la cuestión de si la reducción de dimensiones es necesaria, es decir, si necesitamos ir a lo *latente* del significado, a la propia esencia del LSA, de cuando éste analiza corpora pequeños. Se pone a prueba esta idea junto con otros estándares del LSA: cálculos de *Log-entropía*, SVD, reducción de dimensionalidad y coseno como medida de similitud. Para valorar la eficiencia de todos los parámetros que se estudian de la herramienta, se utiliza el paradigma de la evaluación automática de resúmenes en el ámbito académico, con datos provenientes de evaluadores expertos, de LSA y de los resúmenes de alumnos expertos y no-expertos. La similitud LSA-expertos se toma como variable dependiente para valorar la bondad de cada combinación de parámetros manipulada.

6.2.2.- Extracción de sentidos en corpus específicos de domino

Concluido este primer objetivo, nos propusimos usar uno de los corpus de este primer trabajo para comprobar cómo algunos algoritmos y ajustes sobre LSA pueden ayudar a ubicar con mayor precisión las distintas acepciones teniendo en cuenta su contexto léxico. En concreto, pusimos a prueba palabras aisladas como por ejemplo “fobia”, además de estructuras en las que el sentido de una palabra era dependiente de otra que le acompaña. Por ejemplo, la palabra “fobia” tendrá unas connotaciones u otras, según vaya acompañada por la palabra “sangre” o por la palabra “hablar”. En el primer caso estará relacionado con las “fobias específicas” y el segundo con las “fobias sociales”. En este sentido, fueron puestas a prueba de dos formas. En la primera se utilizó la longitud de vector junto con la del coseno, extrayendo así los vecinos semánticos de las palabras. De esta forma se evitaba que los vecinos semánticos extraídos fuesen poco representativos o que las relaciones con la palabra de la cual se extraen fuesen espurias. En la segunda prueba se empleó del algoritmo de predicación desarrollado por Kintsch (2001). El empleo de este algoritmo nos permitió comprobar la utilidad de este tipo de

procedimientos para extraer el sentido de un término en base al contexto léxico de recuperación.

Puesto que el corpus entrenado versó sobre información diagnóstica, el trabajo realizado tendrá cierto interés para los métodos de recuperación de información en el ámbito médico y psicológico. Además, se abordó otro un tema apasionante dentro de la psicolingüística y que no es otro que la gestión de la polisemia/homonimia. Se puede afirmar que en un texto de información diagnóstica no suele abundar la polisemia (en un sentido estricto), pero si es cierto que las categorías adquieren una acepción u otra dependiendo del contexto circundante. Como se ha ejemplificado más arriba, "fobia" cobra un sentido u otro según vaya acompañado de "gente" o de "tormentas". Por tanto, los mismos mecanismos que dan cuenta de la gestión de los sentidos en las polisemia puras pueden justificarse también en este tipo de estructuras. Todo esto se discutirá también aquí.

6.2.3 Extracción de los sentidos de las polisemias en corpus de dominio general

Siguiendo una secuencia lógica, empleamos los mecanismos del anterior trabajo (corrección en base a la longitud del vector y algoritmo de predicación) pero esta vez sobre un corpus de carácter general, es decir, una muestra del lenguaje empleado en la conversación coloquial. El corpus entrenado fue el Lexesp (Sebastián et al., 2000). Sobre este corpus contrastamos el tratamiento de polisemias como "*partido"*, "*planta", "papel", etc*. En base a esto, analizamos la polisemia sobre un modelo computacional como LSA para preguntarnos sobre si está justificado o no pensar que la mente actúa de manera parecida a como lo hace el LSA. Téngase en cuenta que LSA representa de manera única cada término y no contempla tantas entradas como sentidos posea una palabra. La recuperación de cada sentido se hace simplemente en base a la interacción con el contexto. Además, tratamos también de predecir efectos no deseados y que podrían estar operando en los modelos espacio-vectoriales en a la hora de extraer diferentes significados a una palabra polisémica (inundación del significado predominante, falta de

precisión y bajo nivel de definición), y proponer ideas basadas en el algoritmo de predicación como forma de evitarlos. Para hacer gráfico el proceso, se muestran algunos ejemplos con técnicas modernas de visualización además de realizar un análisis de varianza para comparar los efectos y medir el impacto de las posibles soluciones.

6.2.4 Modelado de fenómenos empíricos: Dificultad de asociación de las palabras ambiguas y su ventaja ante la decisión léxica

El cuarto de nuestros objetivos es el que más se puede ajustar a los parámetros de la psicología cognitiva ya que simularemos con LSA fenómenos que se producen en los datos empíricos. Desde hace mucho tiempo, la ambigüedad de las palabras y la consiguiente desambiguación ha sido uno de los temas de interés de muchas disciplinas. Desde la psicolingüística, el estudio de la ambigüedad se ha focalizado dentro de los modelos de la representación. Mientras unos modelos han defendido representaciones separadas en la que cada sentido posee una representación diferenciada (Klein y Murphy, 2001) otros modelos, en cambio, defienden la existencia de una representación común o "core" (Rodd, Gaskell y Marslen-Wilson, 2002; Klepousniotou y Baum, 2006). Además, desde los modelos espacio-vectoriales, como LSA (Deerwester, Dumais, Furnas, Landauer y Harshman, 1990) , HAL (Burgess and Lund, 2000) o el modelo de tópicos (Griffiths y Steyvers, 2004; Steyvers y Griffiths, 2007), se ha propuesto que una única representación para cada unidad léxica y la participación del contexto, es suficiente para dar cuenta de este fenómeno.

Una dimensión menos explorada del fenómeno de la ambigüedad es lo referente a la diferenciación del fenómeno en base a la ocurrencia de una palabra en diferentes contextos. Por definición, una palabra que ocurra en diversas unidades contextuales (definiendo cada unidad de alguna forma operativa) tendrá mayor grado de ambigüedad. La distribución de cada palabra en estas unidades tiene que estar especificada en su representación, cualquiera que sea esta. Los modelos espacio-vectoriales pueden ser muy útiles para objetivar algunos parámetros para definir la ambigüedad de una

manera más mesurable, dado su representación léxica objetiva y discreta. Un modelo como LSA puede ayudar a dar una definición de la señal lingüística que es procesada por el sistema humano. Encontrar este tipo de definiciones de la ambigüedad (definiendo objetivamente las distribuciones) tiene impacto sobre las teorías de la representación léxica pues puede identificarse que tipo de distribuciones propician la efectividad en ciertas tareas (nombrado, decisión léxica, tareas semánticas, etc) y así justificar los datos empíricos en base a ella. Pueden propiciar que por ejemplo, fenómenos como polisemia puedan ser entendidos en base a la forma en que sus distintos sentidos saturan las dimensiones de su representación o que por ejemplo, los datos empíricos en torno a la concreción/abstracción (si se considera la abstracción un tipo de ambigüedad) puedan ser explicados o no en base a una distribución únicamente lingüística dentro de un marco en el que no sea necesaria la alusión a representaciones primarias.

Una de las características que definen a las palabras ambiguas es el tipo de relaciones que promueven con los demás términos y como se comportan en la tarea de decisión léxica. En este estudio planteamos la hipótesis de que estos fenómenos pueden ser explicados desde un modelo de representación única cómo es LSA. Para el primer fenómeno, a saber, las palabras ambiguas parecen no mantener relaciones semánticas sino asociativas, nos preguntamos si es la propia distribución de los vectores de las palabras ambiguas la que penaliza las relaciones semánticas. Operativamente, esto se puede definir por el hecho de que los primeros vecinos semánticos de ambos tipos de palabras tienen mayores cosenos en las palabras no ambiguas, es decir, las palabras no-ambiguas se relacionan más estrechamente con sus primeros vecinos que las ambiguas. Para el segundo fenómeno, el comportamiento de las palabras ambiguas en la LDT lanzamos la hipótesis de que a partir de un umbral (el definido por sus n primeros vecinos) la penalización se invierte y las palabras ambiguas parecen relacionarse con su segunda tanda de vecinos de una manera más estrecha, es decir, con unos cosenos mayores. Las relaciones con esta segunda tanda de vecinos, mucho más numerosa que la primera, es la manera de operativizar la activación inespecífica que se define en los estudios como causa de la superioridad de las palabras ambiguas en la decisión léxica.

Además, se contempla también la hipótesis de que los patrones conseguidos también se ajustan a las palabras abstractas vs concretas. El hecho de que se adaptasen generaría dos asunciones teóricas: Primero que la abstracción podría ser considerada como una variante extrema de la ambigüedad donde existe una total carencia de focalización estable en cuanto a sus referentes. Segundo, que al repetirse los patrones de las palabras polisémicas-monosémicos pero no seguir el mismo patrón en cuanto a la decisión léxica. Las palabras concretas podrían estar recibiendo activación inespecífica de otro tipo de representaciones. Este tipo de activación recibida, sería suficiente para contrarrestar la activación de las palabras abstractas debida a la distribución de los vectores en los distintos contextos lingüísticos. Es decir, la superioridad de las palabras concretas en la LDT no se puede justificar en base a las propiedades distribucionales por lo que sería necesario apelar a un añadido de activación que proviene de una representación no lingüística. De esta manera tendría justificación la alusión del modelo dual a otro tipo de representaciones y aunque no distintas redes, si se podría proponer que se recibe activación desde otro tipo de representaciones, al igual que otros modelos han formalizado conexiones entre las representaciones semánticas y las representaciones fonológicas, y contextuales (Seidenberg & McLelland, 1989). De igual manera, se podría postular de manera hipotética que si las representaciones extralingüísticas promueven la activación necesaria para la LDT, los procesos semánticos pueden estar eminentemente promovidas por las propias distribuciones de las palabras en los contextos lingüísticos, sin mucha mediación de las representaciones primarias. Todo esto se discutirá en este cuarto trabajo.

Parte empírica: Estudios y Simulaciones

Capítulo 7

Latent Semantic Analysis Parameters for Essay Evaluation using Small-Scale Corpora

(Artículo publicado en Journal of Quantitative Linguistics 2010, Volume 17, Number 1, pp. 1–29)

Journal of Quantitative Linguistics
2010, Volume 17, Number 1, pp. 1–29

Latent Semantic Analysis Parameters for Essay Evaluation using Small-Scale Corpora

Guillermo Jorge-Botana, José A. León, Ricardo Olmos & Inmaculada Escudero
Universidad Autónoma de Madrid, Spain

Abstract

Some previous studies have pointed to a need for additional research in order to firmly establish the usefulness of LSA (Latent Semantic Analysis) parameters for automatic evaluation of academic essays (Van Bruggen et al., 2004). The extreme variability in approaches to this technique makes it difficult to identify the most efficient parameters and the optimum combination (Haley et al., 2005; Haley et al., 2007). With this goal in mind, we conducted a high spectrum study to investigate the efficiency of some of the major LSA parameters in small-scale corpora. We used two specific domain corpora that differed in the structure of the text (one containing only technical terms and the other with more tangential information). Using these corpora we tested different semantic spaces, formed by applying different parameters and different methods of comparing the texts. Parameters varied included weighting functions (Log-IDF or Log-Entropy), dimensionality reduction (truncating the matrices after SVD to a set percentage of dimensions), methods of forming pseudo-documents (vector sum and folding-in) and measures of similarity (cosine or Euclidean distances). We also included two groups of essays to be graded, one written by experts and other by non-experts. Both groups were evaluated by three human graders and also by LSA. We extracted the correlations of each LSA condition with human graders, and conducted an ANOVA to analyse which parameter combination correlates best. Results suggest that distances are more efficient in academic essays evaluation than cosines. We found no clear evidence that the classical LSA protocol works systematically better than some simpler version (the classical protocol achieves the best performance only for some combinations of parameters in a few cases), and found that the benefits of reducing dimensionality arise only when the essays are introduced into semantic spaces using the folding-in method.

1.- Introduction

Latent Semantic Analysis (LSA) is a computational linguistic model that offers a quantitative representation of a semantic domain. It was first described as an information retrieval method (Deerwester, Dumais, Furnas, Landauer & Harshman, 1990) derived from the Salton's vector-space model (Salton, 1983) but it was Landauer and Dumais (1997) who first demonstrated its ability to account for phenomena related to knowledge acquisition and representation; other authors have demonstrated its suitability for taking in account some additional cognitive phenomena (Kintsch, 2001). Basically, the protocol is as follows. 1) Analyze a corpus and construct a dimensional matrix where each row represents a unique digitalized word (term) and each column represents one document, one paragraph, one sentence, etc. (depending on the contextual window that has been chosen). 2) After some linguistic calculations on this matrix (local and global weighting of each term), reduce the original matrix via Singular Value Decomposition (SVD), a mathematical technique that transforms the occurrence matrix X into three other matrices (reduced to k dimensions which represents abstract concepts), a term-concept vector matrix, U, a singular values matrix, S, and a concept-document vector matrix V ($X=USV^T$), where in matrices US and SV it is possible to compare different sections of text (words, sentence, paragraph, essays, summaries) with adjoining units of the text to determine the degree to which the two are semantically related (figure 1). LSA usually measures the similarity between two pieces of text using the cosine between the two vectors. If the cosine is near to one, the two sections of text are very semantically similar, and if the cosine is near to zero the two sections are not semantically related at all.

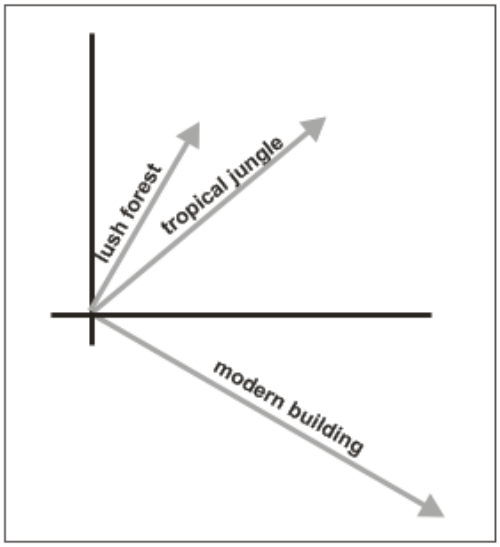


Figure 1. Graphical example of LSA representing three texts (vectors)

To summarise, LSA has been proposed as a model suitable for simulating the representation of the lexicon. LSA, though, does not constitute a single, consistent, well-defined stepwise method. LSA is dependent on the interaction of multiple parameters.

The theoretical aim of the present study is to establish the parameters which interact, limiting ourselves to the application of LSA as a tool for assessing academic essays, and evaluating its efficiency compared to human graders, as in some recent studies (Haley et al., 2005; Haley et al., 2007).

2. Variability of approach to LSA has produced mixed results in terms of effectiveness when assessing academic essays.

Many researchers have obtained positive results using LSA to emulate human graders. However, there are considerable differences between studies in terms of how LSA is conducted. This variability often prevents us from clearly identifying the parameters critical to the success or failure of each attempt (Haley, Thomas, De Roeck & Petre, 2005; Haley, Thomas, Petre, & De Roeck, 2007). Some examples of these technical parameters might be the elimination of certain structures (e.g. lists of stop words), weighting functions (an estimation of how representative a word is of the documents where it occurs), different dimensionality reductions applied to the term-document matrix

(truncating the matrix after SVD to k dimensions), different measures of similarity in the comparison of texts (for instance the habitual cosine), size of the corpora (normally measured in number of characters, number of bytes or number of terms and documents) and composition type (taking into account structure, type of text and number of themes treated in the corpora). Given the wide range of possible combinations, our question is whether any of these parameters of LSA are efficient (evaluating essays compared to human graders) in all conditions, or whether there are certain conditions under which their usage is invalid or even counterproductive.

2.1 Different dimensionality, different results

Dimensionality reduction is considered to be the core of the LSA technique. For this reason it is worthwhile investigating the contribution of SVD and subsequent dimensionality reduction to overall efficiency in generating semantic spaces that represent pieces of text well. Since we have the occurrence matrix (X), a matrix where each row represents a unique word (term) and each column represents documents in which that term occurs, LSA reduces such a matrix via SVD, a mathematical technique for reducing dimensionality. This process generates a vector-space that is not influenced by the irrelevant dimensions (which are removed) and allows us to avoid the noise from variability of usage of different terms that designate the same things. Theoretically, all terms and documents are then represented with the k most relevant dimensions (abstract and non-intuitive concepts). The value of k is an open, empirical parameter (Figure 2).

Wiemer-Hastings, P., Wiemer-Hastings, K., & Graesser (1999a) was the first to question the efficiency of such a reduction. He found that a version of LSA without SVD (using only weighting functions and geometric comparisons such as cosines) obtained similar results to the full version (with dimensionality reduction after SVD). Even a version with searches for literal words performed satisfactorily at automatic essay evaluation, although less so than the above methods. SVD made LSA more robust and capable of exploiting the contexts words are found in, as well as managing certain phenomena such as synonymy

and homonymy better. The authors suggested that dimensionality reduction and the weighting function's efficiency depend on the size of the corpus and the size of the paragraphs to assess. Wiemer-Hastings et al (1999a) said that the differences between the three conditions tested (key word searches, using only entropy weighting and geometric comparisons, and with dimensionality reduction) could possibly be greater using larger texts, as in earlier studies by Landauer and Dumais (1997). In fact, Landauer and Dumais offered us the following details: with no dimensionality reduction, only 15% precision can be achieved, while the precision grows to the 45% - 53% range with a dimensionality reduction around 300 factors. These initial studies led to a discussion of the differences in LSA behavior according to the conditions under which it was used, hinting at the schism between general domain corpora and specific domain corpora – the latter being smaller and more difficult to control.

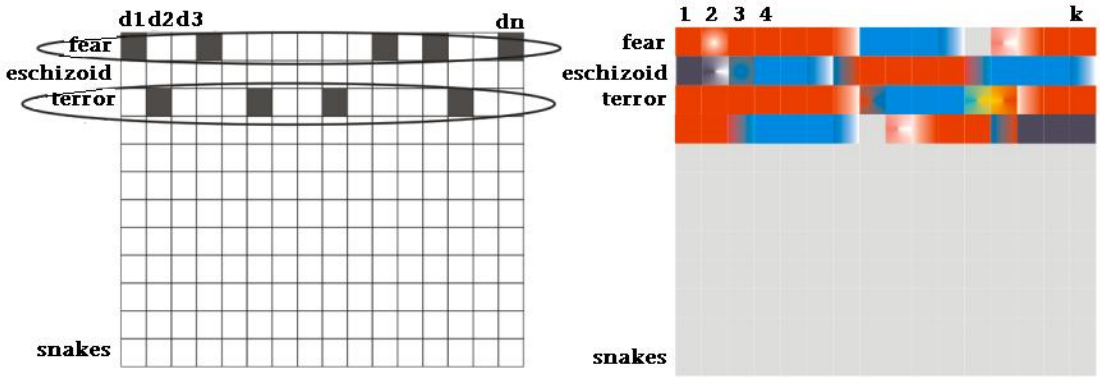


Figure 2. Graphical example of the occurrence matrix (left side). Such a matrix computes the occurrence of a term in each document. The final result (right side) is term representation in the k most representatives dimensions.

In specific domain corpora, some experiments has shown implicitly that the dimensionality reduction is no more efficient than the mere application of 100% of dimensionality (i.e. without any reduction) (e.g., Cox & Shahshani, 2001; Kontostathis, Pottenger & Davison, 2005; Kurby, Wiemer-Hastings, Ganduri, Magliano, Millis & McNamara, 2003; Olde, Franceschetti, Karnavat & Graesser, 2002; Silva, Martinez & Ruiz, 2004). One factor is common to these articles: the best conditions of dimensionality reduction are distributed in an asymptote without any improvement in efficiency until the largest possible

number of dimensions (no dimensionality reduction). For instance, Kurby et al., (2003) tested a range from 50 to 450 dimensions, and found optimum performance with 450. This leads us to ask whether this performance would be maintained above 450, up to the point where there was no reduction (all dimensions). Olde et al., (2002) proclaimed that around 300 dimensions produced the highest performance, and that from 300 to 500 (the maximum) no variation in performance was observed. Silva et al., (2004) used a dimensionality range from 100 to 1000. The best performance is obtained between 700 and 1000 dimensions (the maximum). These authors thought the effect might be due to the number of documents used to build the semantic space. Kontostathis et al., (2005) obtained similar results - taking precision as a measure of effectiveness, most of the trials show that the highest number of dimensions (usually with no dimensionality reduction) produces best results. Cox & Shahshani (2001) applied the technique to a corpus extracted from telephone conversation transcripts, and found that the spaces with no dimensionality reduction showed better performance than those with dimensionality reduction.

Paradoxically, for specific domain corpora a dimensionality below 150 is the norm, usually justified by the nature of the domains themselves (e.g., Wiemer-Hastings, P., Wiemer-Hastings, K., & Graesser, 1999b; Wolfe, Schreiner, Rehder, Laham, Foltz, Kintsch & Landauer, 1998; Foltz, Britt & Perfetti, 1996; Foltz, Kintsch & Landauer, 1993; Dumais, 1991). A very low dimensionality has even been used on occasions - around 30 (e.g., Nakov, Popova & Mateev, 2001; Nakov, 2000b). Although there have been attempts to unify approaches, using percentage values and recommending that the original matrix and the reduced matrix share 50%, 40% or 30% of the dimensionality (Wild, Stahl, Stermsek & Neumann, 2005), it is difficult to draw conclusions about the best dimensionality. The variability of dimensions in specific domain simulations leads to doubts over the extent to which reducing dimensions results in an improvement. Such doubts extend to studies that have imposed a dimensionality reduction a priori, assuming that it would be better than no reduction at all. These doubts increase when we consider the variability in the composition of the semantic spaces.

2.2 Weighting functions

Once an occurrence matrix (**X**) is constructed, and before SVD is carried out, local and global weighting functions can be applied to it. The weighting functions transform each raw frequency cell x_{ij} of the matrix, using the product of a local term weight, l_{ij}, and a global term weight, g_i. This process attempts to estimate the importance of a term in predicting the topic of documents in which it appears. There are different ways to calculate local and global weights. Navok et al. (2001) claim that both local and global weights affect final results, and for these authors it is always advisable to apply the local weight, for instance Log (in formula 2). This finding gained supported from simulations by Wild et al. (2005), although Navok (2000a) had achieved very good results using only frequency of occurrence as the local weight (formula 1).

(1) TermFrequency

$l_{ij} = tf_{ij}$

(2) Log

$l_{ij} = log(tf_{ij} + 1)$

where tf_{ij} is the number of occurrences of term i in document j

Regarding the implementation of global weights, Entropy (formula 3) or IDF (Inverse Document Frequency) (formula 4) have been the more common formulae used with LSA, but conclusions drawn vary considerably. Some claim that IDF seems more beneficial because it appears to offer best performance in more trials (Wild et al., 2005; Navok et al., 2001), although in Navok et al.'s (2001) experiment the Entropy function also seems to be consistently beneficial. Entropy is the most widely-used function in previous studies, and also appears to have offered very satisfactory results (Haley et al., 2005; Nakov, 2003; Dumais, 1990), although Wild et al., (2005) see results in a less positive light. In summary, the use of these functions appears to be more effective than not applying any function at all, but there is huge variability among results.

(3) IDF

$g_i = log_2 (n / df_i) + 1$

where $\boldsymbol{df_i}$ is the number of documents in which term ***i*** occurs.

(4) Entropy

$g_i = 1 + \Sigma_j (p_{ij} \log(p_{ij}) / \log(n))$

where $p_{ij} = tf_{ij} / gf_i$

where
$\boldsymbol{tf_{ij}}$ is the number of occurrences of term ***i*** in document ***j***
$\boldsymbol{gf_i}$ is the total number of times term ***i*** occurs in all documents
n is the number of documents

The final product of local and global weight ($\boldsymbol{x_{ij} = l_{ij} * g_i}$) will be the final value of each cell. In this study, we use Log as local weight and both IDF and Entropy as global weight (Log-Entropy and Log-IDF).

2.3 Similarity measures

Although other measures of similarity such as Spearman's correlation (Wild et al., 2005) have occasionally been used between vector-texts, as Haley et al. (2005) noted, the cosine measure (formula 5) is practically ubiquitous for assessing academic texts. The usual method in automatic grading extracts the cosine between the vector representing the response of each student and the vector that represents an "ideal" response written by an expert (a golden essay).

$$(5)\ Cos(Vw1, Vw2) = \sum_{i=1}^{K} (Vw1_i \cdot Vw2_i) / (|Vw1| \cdot |Vw2|)$$

where *Vw1* is the vector representing the first essay, *Vw2* is the vector representing the second essay and *k* is de number of dimensions

$$(6)\ Dis(Vw1, Vw2) = \sqrt{\sum_{i=1}^{K} (Vw1_i - Vw2_i)^2}$$

where *Vw1* is the vector representing the first essay, *Vw2* is the vector representing the second essay and *k* is de number of dimensions

But the cosine has some limitations, however. When we consider similarity indices based on the cosine, there is one problem that is especially evident in applications for grading automatically academic essays with LSA. A student can construct a response (for instance to an exam question) by introducing a very small number of highly representative terms, or even using simple repetition of the words of the question. In these cases, the vector that represents the answer given by the student is very similar to the vector representing the ideal response proposed by experts. Due to this similarity, the cosine will overstate the grading. A very small essay comprising only high-frequency critical words would be represented with a vector whose position is very close to that of the ideal answer vector. Nonetheless the two vector lengths are extremely different, since the vector that represents the student summary is extremely small. Rehder, Schreiner, Wolfe, Laham, Landauer & Kintsch (1998) proposed cosine as the best measure for assessing academic essays, but nevertheless became aware of this problem. They suggested enriching the cosine by using other measures, but left this task to future researchers. More recent studies have questioned the ability of the cosine to measure summaries, with high similarity possibly denoting only paraphrasing and repetition of words (Millis, Kim, Todaro, Magliano, Wiemer-Hastings & McNamara**,** 2004; Kurby et al., 2003; Wolfe & Goldman, 2003). Other studies have demonstrated that Euclidean distances (formula 6) have some advantages over cosines in the evaluation of academic essays (Olmos, León, Jorge-Botana & Escudero, in press) in essays written by non experts.

2.4 Pseudo-documents

Our aim is usually to compare two pieces of text which are not represented as documents in the semantic space, for instance, when we have to extract the similarity between each student response and the vector that represents an "ideal" response written by an expert. Imagine that a student answers *"is an anxiety disorder characterized by overwhelming anxiety and excessive self-consciousness in everyday social situations. Social phobia can be limited ….."* and following the expert method, we have to extract the cosine between such a response and the ideal response (golden response). But there is not a document in the semantic space (in the matrix VS) that coincides exactly with the text of the student response, nor of the ideal response. We thus need to represent a new document in the current semantic space generated by LSA, for the student response and also for the ideal response.

The two ways of representing a new document in the semantic space generated by LSA are *Vector Sum* and *Folding-In* (Berry, Dumais & O'Brien, 1995; Deerwester et al., 1990). The Vector Sum method is based on representing a text as the sum of the vectors of terms it contains, so that the resulting vector is another vector-term. Conceptually, this method implies that the meaning of a document is the sum of vectors of the words that constitute it.

In contrast, the Folding-In method projects the new document into the matrix of documents, as an extra document – a new vector-document. Following this method, new documents are introduced into the matrix $\boldsymbol{V}$ in the space of an existing LSA simply using the equation $\boldsymbol{d = e^T U S^{-1}}$. A new vector $\boldsymbol{d}$ can be created by computing an essay $\boldsymbol{e}$, (a new vector column in the ocurrence matrix $\boldsymbol{X}$ with all the terms that occur in it), and then multiplying it by $\boldsymbol{U S^{-1}}$. $\boldsymbol{e}$ is also computed by applying the same global and local weights as in the creation of the original space.

3. Objectives

The main objective of this study was to investigate the efficiency of a series of parameters associated with LSA, applied to grading student essays with small-scale corpora. In our experiment 120 variants of the combination of LSA parameters have been tested. These variants are constructed using the parameters that are considered crucial (see introduction). These parameters include 2 specific domain corpora, 3 weighting functions (including no weighting), 5 levels of dimensionality reduction (expressed as percentages), 2 approaches to building pseudo documents (centroid vector sum vs. Folding-in) and 2 measures of similarity (cosine vs. Euclidean distance). In addition, in order to assess LSA performance, two groups of students were evaluated: Experts and Non-Experts. The text compiled by Experts provided more data than the Non-Experts, who often answered poorly in terms of both content and length. Each of the 120 variants of LSA was compared to the human assessment, represented by the average of assessments by three experts in psychopathology. The student essays assessed by both LSA and human graders comprise a response to the question "What is a social phobia?"

4. Method

4.1 Material

The study uses essays from 80 students in answer to an open-ended question: "What is a social phobia?" with no word limit imposed. These essays were written by two groups, each of 40 students. The first group comprised experts in psychology (4^{th} year degree) with recently-acquired knowledge of anxiety disorders. The second group was made up of first year speech therapy students - a group that is considered inexperienced as they have marginal knowledge of anxiety disorders. Written responses were assessed by each of three expert professors in clinical psychology, and also by each of the LSA combinations. Regarding the assessment of the LSA system, 30 semantic spaces are created by combining the possible values training a space with LSA. Added to these training combinations we have the comparison stage variables

(2 approaches to building documents, and 2 measures of similarity). Our evaluation of LSA is carried out using the *expert method* (Foltz, Laham & Landauer, 1999; León, Olmos, Escudero, Cañas & Salmerón, 2006), which compares the vector representing the response of each student with the vector that represents an "ideal" response written by an expert. For the LSA calculations, we used Gallito ®, a tool programmed in Microsoft .Net framework ® (VB.NET, C#) and integrated with Matlab® developed in our research group www.elsemantico.com.

4.2 Parameters manipulated in the study

I) Linguistic corpus: Two corpora were processed. The structured one is a corpus extracted from the DSM-IV (Diagnostic and Statistical Manual of Mental Disorders)and ICD-10 (International Statistical Classification of Diseases) compendia of Mental Disorders. The information in this text is hierarchical and structured (e.g. social phobia is a subset of phobias, and phobias belongs to the category anxiety disorders), and there is little variability of terms (only technical terms are found). It is a typical corpus with no tangential information. The unstructured corpus was extracted from the Internet. It contains texts that focus on psychopathology and Mental Disorders, but which cover very different topics. They contain much tangential information and display more variability in term usage (see table 1). In both corpora, documents were manually split by the authors according to topics treated and terms that occurs at least in two documents. Tokens that appeared on a stop list were removed.

In order to somehow measure the structure of the spaces formed with these two corpora, we used a transformed metric derived from the first and second order relations given by Mill & Kontostathis (2004), and Kontostathis & Pottenger (2006). These measures provide a good estimation of whether the relations in a space are shared among only a few terms, or are shared among many terms, only a few of which occur frequently. Mill & Kontostathis (2004) estimated the first-order matrix by multiplying the matrix of co-occurrences by its transposed matrix ($\boldsymbol{XX^T}$). This operation resulted in a term-by-term matrix that

represents the number of times two terms occur together in a document. Superior orders are calculated using this matrix, transforming it cell-by-cell into binary scores (1 where terms co-occurred at least once and 0 where terms did not occur together), and the diagonal was also set to zero values resulting a new matrix called ***B***. The resulting B matrix is multiplied by itself to produce the second-order matrix (***BB***). The significance of the first- and second-order matrices is different. Each cell in the first order matrix represents how many times two words occur together in the documents, while each cell of the second-order matrix represents the number of words that act as a "bridge" between the terms representing the rows and columns - in other words, where two terms do not occur together but occur with a common term (a bridge term). To express the first- and second-order relations using an index, regardless of the size of the corpus, we propose the following formulae.

The first-order index is calculated using the matrix of binary scores. It is the average of the number of terms that occur with each term from the space. We then apply a percentage conversion bearing in mind the total number of terms in the corpus. A semantic space that scores 13.51, for example, means that each term from that space occurs on average with 13.51% of the terms.

$$I_1 = \frac{(\sum\sum x_{ij} / n) \times 100}{n}$$

where *Xij* is the number of times a term occur with another term and *n* is the number of terms. A percentage conversion is applied

The second-order index is calculated using the second-order matrix. It is the average of the number of words that act as a "bridge" between each pair of words (the average of all cells). A percentage conversion is then applied, depending on the total number of terms in the corpus. A semantic space that scores 3.52, for example, means that each pair of terms has on average 3.52% of the total number of terms acting as a "bridge" between them.

$$I_2 = \frac{(\sum\sum x_{ij} / n^2) \times 100}{n}$$

where *Xij* is the number of terms that serves as a "bridge" between each pair of terms and *n* is the number of terms. A percentage conversion is applied.

High scores for such measures show high density of the relationships between terms in the texts. This means that most of terms are found jointly with many other terms, both directly and indirectly. High scores are associated with a structured and cohesive corpus with few tangential terms. We have called this measure the "relation density". As we can see from the properties in Table 1, the unstructured corpus has a lower first and second-order score than the structured one. The repercussion of these indicators will be discussed later.

Other indices are the text length (number of words and characters) and the length of co-occurrence matrices. In terms of text length, the larger corpus is the structured one. In terms of matrix length, the unstructured corpus is the larger.

	Text size		Matrix Size	Order	
Corpus	Words	Characters	Terms (docs)	First-order	Second-order
STRUCTURED	162,517	918,016	5,416 (446)	13.51	3.52
UNSTRUCTURED	141,045	765,932	6,844 (717)	6.24	0.89

Table 1. Metrical properties of the two corpora used in the study.

II) Dimensionality: Instead of taking absolute values, we prefer to take the percentage of singular value accumulation that is preserved after SVD (a criterion proposed by Wild et al., 2005). This method is justified because it is relatively independent of the number of documents and terms in texts. Following this method, we obtained 5 conditions: 20 %, 40 %, 60 %, 80 % and 5. 100 % (this last condition is with no dimensionality reduction).

	Matrix Size					
Corpus	Terms (docs)	20%	40%	60%	80%	100%
STRUCTURED	5416 (446)	45	112	195	298	446
UNSTRUCTURED	6844 (717)	61	150	266	414	717

Table 2. Dimensions of the two corpora used in the study, according to the percentage of singular value accumulation.

III) Weighting functions: for weighting we used the habitual relationship between the local and global weighting of terms (see Navok, et al., 2001). Two possible variants have been chosen, both of which take into account local weighting. The three conditions are, then, Entropy, IDF, and No pre-processing.

IV) Method: When we translated student and "ideal" expert essays to the semantic spaces, we used the two methods Vector sum and Folding-in (see section 2.5).

V) Measure: As mentioned in section 2.4, the two measures used in the study are Cosine and Euclidean distance.

VII) Groups: Expert and Non-expert. This is the grouping variable. In order to understand the characteristics of the essays written by each group, we extracted the mean number of words in each kind of essay, and found that Non-Expert essays are significantly shorter (28 words) than those produced by Experts (88 words).

4.3 Procedure

Combining all the levels of all the parameters involved in creating a space and comparing the documents, we obtain a total of 120 LSA conditions [2(corpus) x 5(dimensionality) x 3(Weighting functions) x 2(Method) X 2(Measure)]. Each student essay is assessed using the 120 LSA conditions as well as the human graders. Thus, we have 80 essays by the two groups of students, assessed by 3 human experts and 120 conditions of LSA. Such assessment produced a matrix of essays by graders (LSA conditions and human experts).

The procedure is as follows. First, we used the Pearson correlation coefficient to compare the LSA assessment of each condition with the expert graders' assessment. Secondly, we obtained measures of coincidence (correlations) between LSA and human graders, in order to demonstrate the

suitability of each combination of LSA parameters. To do so, we calculated the mean of the three human experts' scores for each essay, we standardized the human expert score (1 to 10) and the score for each LSA condition (cosine and Euclidean distances) to homogenize them. We then subtracted each LSA condition score from the human markers' score to obtain a matrix of coincidences (comprising absolute values – the smaller the difference the more similar the assessments). To summarise, each column represents the coincidence of an LSA condition with human criteria in evaluating each essay. We conducted a repeated measure ANOVA on this matrix, to compare and observe the main and interaction effects which might reveal the key parameter combinations for using LSA on specific domain corpora. A group variable is added to this repeated measures analysis: experts and non-experts. Analysis was performed using the SPSS 15 statistical package.

5. Results and discussion

5.1 LSA and human grader correlations

We extracted the correlations between scores resulting from each of the 120 LSA conditions and human graders' scores, in order to evaluate the technique. We sought the parameter conditions that best simulate human behaviour (assessment in the case of the present study). As a starting point, then, it was important that LSA (in all conditions) generally correlates well with human graders. First, we obtained the correlation between the three graders themselves. We found high correlations between Grader1-Grader2 (0.82), Grader1-Grader3 (0.82), and Grader2-Grader3 (0.91). Next, general correlations were extracted between each of the LSA variants and the average score of the human graders. The distribution of these correlations is presented below (Figure 1). The maximum correlation was 0.89, achieved with the unstructured corpus, 20% dimensionality reduction, Entropy, Folding-In and Euclidean Distances, showing that some conditions can be very effective. The minimum correlation was 0.30, the variability between conditions underlining the fact that some parameter interactions work much better than others. The average correlation was 0.71.

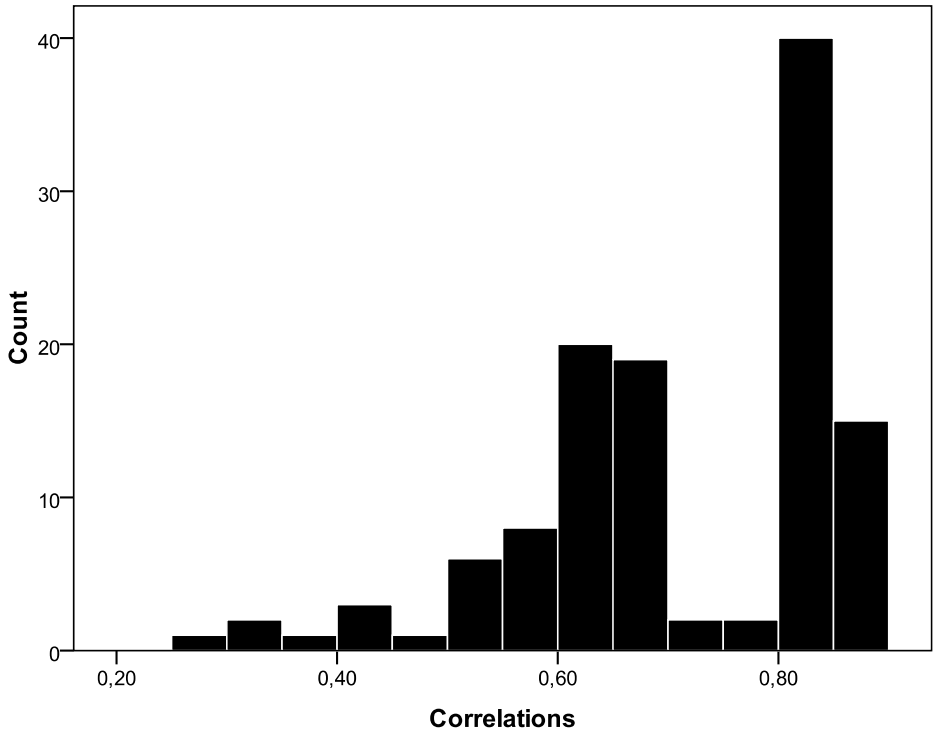


Figure 3. Histogram showing the correlation between human graders and LSA conditions

In order to extract a possible hypothesis, under which the distribution curve of correlations has a bimodal shape (as in Figure 3), we tentatively focused on the parameter Measure (Cosine and Euclidean distance). One of the hypotheses is that the Euclidean distance (measuring similarity) corrects the effect that occurs in essays that do not exceed a certain length and content, as seen in other studies (Olmos et al., in press). The effect in question is that in a short answer with little content, the mere occurrence of a key high-frequency term can produce a vector very close to that of the "ideal" response, and LSA scores are exaggerated.

Another way of thinking is that the distance takes into account both the amount of information and the content of the essay. The box graph in Figure 4, segmented by the variable "measure of similarity" explains such bimodality. From now on we will refer to this as the "distance corrector effect". According to this graph, distance is not only more effective, but is also more regular and stable across all LSA parameter combinations. In figure 4, we also extracted

the differential distribution between the two methods of constructing pseudo documents in the two measures conditions, and found no significant differences (although Folding-In displays more variability in efficiency, and more extremely good evaluations in Euclidean Distances). We can conclude that LSA scores are reliable compared to the graders' scores, depending on the condition – we will analyze variables in more depth in the following section.

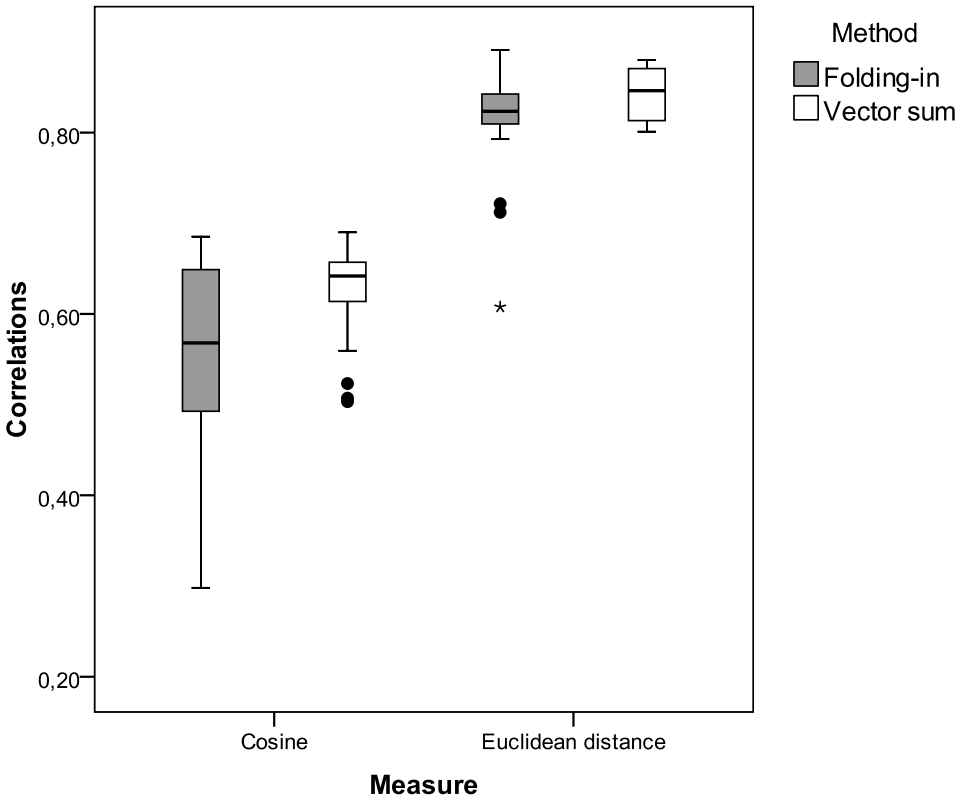


Figure 4. Distribution of correlations with cosine and Euclidean distance measures and Folding and Vector sum.

5.2 ANOVA: Effect of LSA parameters

First of all, we should bear in mind that the dependent variable of this ANOVA is the difference between LSA and average human grader score (previously standardized to put them on the same scale). So, a small difference between LSA and human graders mean a good LSA parameter combination, and vice versa, a high difference mean a poor LSA parameter combination. Secondly, although many parameters are involved, in this ANOVA, we have examined all possible interactions in order to gain an overview of the

regularities, including third order interactions where they exist and invalidate the second order ones.

Reviewing the results of the interactions, we find two phenomena worthy of discussion: Confirmation of what we referred to in paragraph 5.1 as the "distance corrector effect", and evidence that the advantages of reducing the dimensionality under some conditions are only found using the Folding-In method. Vector Sum is completely unaffected.

With respect to the measures of similarity, we found that Euclidean Distances behave significantly better than the cosine as a main effect. This confirms the results of the correlation measures (figure 4), which led us to postulate that the bimodality of the distribution is due to what we termed the **"**distance corrector effect". This beneficial effect in favour of the Euclidean distances is due to the fact that they combine two basic measures - the closeness (like the cosine) of two vectors in the n-dimensional space, and the strong prediction of knowledge represented in the length of the vector (Rehder et al., 1998). However, Distances behave differently depending on the condition. It would seem useful, then, to analyze these possible behaviours.

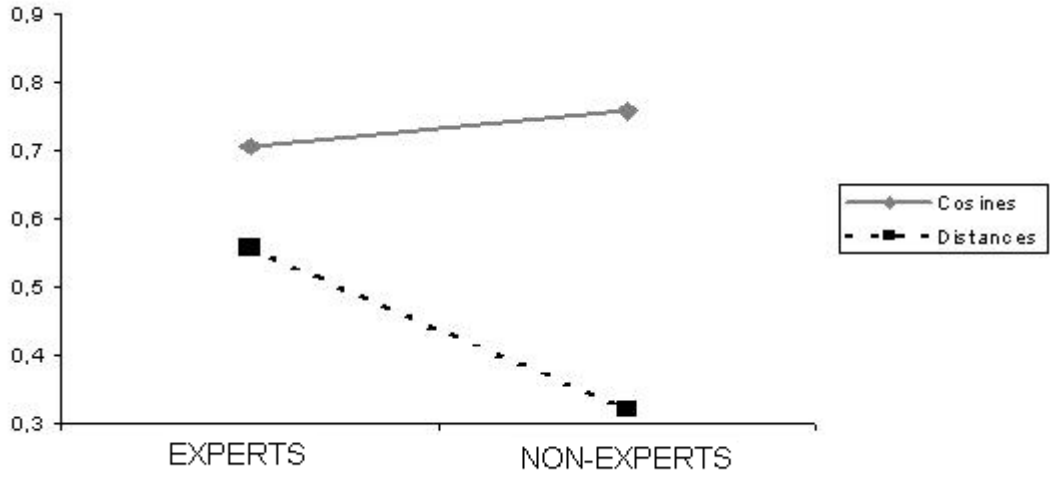


Figure 5. Interaction between Group and Measures of similarity

One of the more robust interactions occurs between group and measure of similarity [F(1,78)= 49.9; MSE= 3.47; p< 0.05]. In both groups, experts and

non-experts, the Euclidean distance correlates better than cosine with human graders. In the non-experts group this difference is particularly dramatic (see figure 5 and numeric values in table 3) due to an effect that occurs in short essays (which often coincide with the Non-experts group).

	EXPERTS		NON-EXPERTS	
	Mean	SEM*	Mean	SEM
COSINES	0,71	0,06	0,76	0,06
DISTANCES	0,56	0,04	0,32	0,04

* Standard Error of the Mean (SEM)

Table 3. Interaction between Group and Measures of similarity

To illustrate this phenomenon, let us take an example. As a response to the definition of "social phobia", we might find a short essay with only a few high-frequency key terms – for example “social phobia is a fear or a phobia of people”. A human grader would award a low score since it does not cover all possible content pertinent to the topic. However, due to the similarity of the vector for this response with that of the “ideal” answer vector, the score will be overstated by the cosines. The cosine measure does not take into account the length of the response vector, while the Euclidean distance does. Given that the non-expert students tend to produce answers with very few words, it may be that using cosines as a measure of similarity promotes an overestimation of evaluation scores. Euclidean distances mitigate this effect, and responses were graded in a manner more consistent with human graders, especially for the non-expert group where the differences between cosine and Euclidean distances are significantly larger.

In addition to this effect, the Euclidean distance tends to be more effective with spaces where its dimensionality has been reduced [$F(4,312)=30.33$; $MSE= 0.062$; $p< 0.05$].

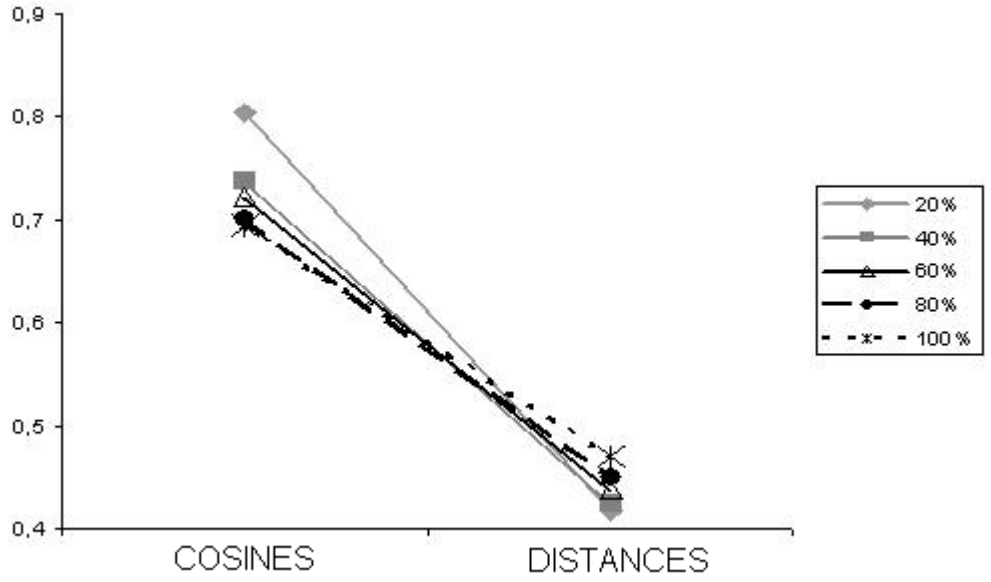


Figure 6. Interaction between Measures of similarity and Dimensionality. In the distance condition, there are significant differences ($p < 0.05$) between 100% and the other reductions.

From the interaction between Dimensionality and Measure of similarity (figure 6 and numeric values in table 4), it is clear that the superiority of the Euclidean distance tends to become more evident in spaces where dimensions have been reduced.

	20%		40%		60%		80%		100%	
	Mean	SEM	Mean	SEM	Mean	SEM	Mean	SEM	Mean	SEM
COSINES	0,80	0,05	0,74	0,04	0,72	0,04	0,70	0,04	0,69	0,04
DISTANCES	0,42	0,03	0,42	0,03	0,44	0,03	0,45	0,03	0,47	0,03

Table 4. Interaction between Measures of similarity and Dimensionality.

This is logical, as the goal of dimensionality reduction is to represent each word with substantial information and delete the dimensions that discriminate less effectively between content. We might say that this reduction deleted the dimensions that reduced the salience of the key terms, so that with dimensionality reduction the key terms become more crucial. The occurrence of a key term represented by a vector from a dimensionality-reduced space will have more impact on the texts it appears in, while a key-term's impact is diluted in spaces that keep all dimensionality. The key term's participation in similarity measures with the "ideal" vector will therefore be greater with dimensionality

reduction, and cosines will overstate scores in Non-expert texts and very short essays. In contrast, no such risk exists with Euclidean distances for the aforementioned reason (our "distance corrector effect"), and a reasonable dimensionality reduction offers only benefits (all dimensionality reduction options were significantly better than 100%).

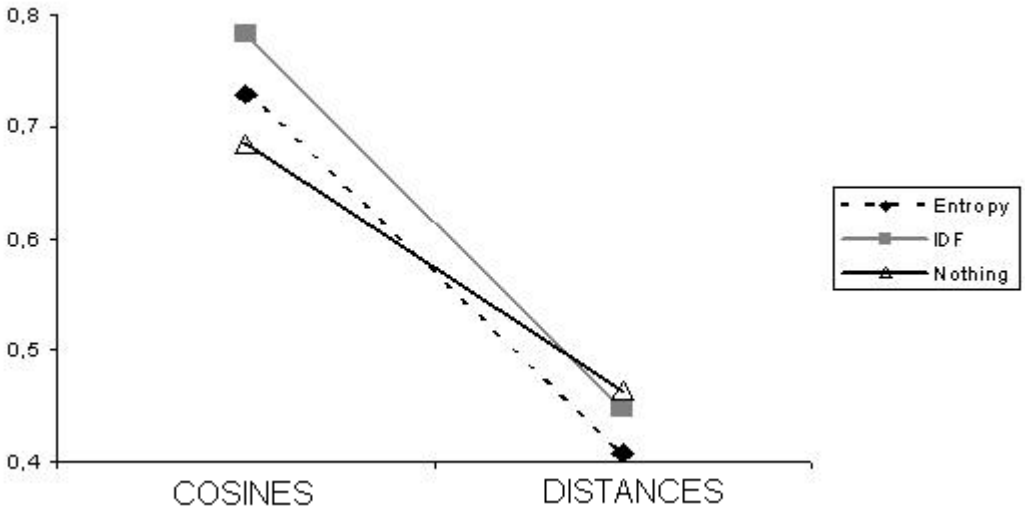


Figure 7. Interaction between Measures of similarity and Weighting function. With distances, there is significant difference ($p < 0.05$) between entropy and the others.

Similarly, we found that distances tend to behave better in spaces where some kind of Weighting function was applied (figure 7 and numeric values in table 5) [$F(8,624)=0.81$; $MSE= 0.32$; $p< 0.05$]. Again, key terms that come from a space with Weighting have more salience, and thus more impact within documents. Cosines are extremely sensitive to the risks of this impact in the essays of Non-experts. Entropy, on the other hand, seems to be the best option when Distances are used.

	ENTROPY		IDF		NOTHING	
	Mean	SEM	Mean	SEM	Mean	SEM
COSINES	0,73	0,04	0,78	0,04	0,68	0,05
DISTANCES	0,41	0,03	0,45	0,03	0,46	0,04

Table 5. Interaction between Measures of similarity and Weighting function.

2) In terms of the way pseudo documents are constructed, Vector sum and

Folding-in show no significant differences in overall efficiency. Vector Sum results in less variability in scores and less extreme cases (Figure 4). But all positive effects relating to dimensionality reduction are achieved only with folding-in: we found a third order interaction between method, dimensionality and other variables that indicate this effect. We found Corpus × Dimensionality × Method [$F(4,312)= 6.75$; $MSE=0.06$; $p< 0.05$], Group × Dimensionality × Method [$F(4,312)=7.83$; $MSE=0.07$; $p< 0.05$], and Weighting × Dimensionality × Method [$F(8,624)= 5.70$; $MSE=0.02$; $p< 0.05$].

The key to understanding such interactions is as follows: with Vector sum, dimensionality reduction did not improve results in any of the different corpora (Figure 8-left, data in table 6), with any of the different levels of expertise (Figure 9-left, data in table 8) or with any of the different pre-processes (Figure 10-left, data in table 10). This means that one reason for using LSA - the benefits of dimensionality reduction - tends to be eliminated when the Vector sum method is used. Using Folding-In on the other hand, there are conditions where dimensionality reduction proves more efficient than full dimensionality (Figure 8, 9, 10-right and numeric values in tables 7, 9, and 11 respectively).

The crucial question is whether it is better not to reduce the dimensionality and use vector sum in all cases, or whether in some conditions reduction and Folding-In are better. One reason to be optimistic about the benefits of dimensionality reduction with Folding-In was that one of the main effects showed that Distance is significantly better than Cosine measures (with an extremely large difference). Distances tend to work better in spaces that have been reduced (Figure 6), so there might be some conditions (measured with Euclidean Distance) where the benefits of dimensionality reduction were vital. In these cases, then, we should use Folding-In. As we can see from Figure 4, Folding-In displays more variability in efficiency, and more extremely good evaluations. In fact, the best two combinations found in this study (a correlation of 0.891 and 0.885 with human graders) was achieved under a condition using Entropy as a weighting function, Folding-In as the method and Distance as the measure of similarity.

There are other variables where we discover that dimensionality reduction plays a predominant role under Folding-In. Firstly, the effectiveness of dimensionality reduction is sensitive to the properties of the corpora (Figure 8-right and numeric values in Table 7). The structure of the corpus influences the benefit of dimensionality reduction via the "relationship density" between terms, which is what we measured using the first and second order indices. These indices are much lower in the unstructured corpus than in the structured one - proportionally the terms have fewer relationships because there are more sub-meanings and more tangential information. As Table 1 shows, the first order indices for the unstructured corpus are half those of the structured one, and the second order indices for the unstructured corpus are even smaller in proportion. Thus structure as well as size of texts contributes to the effectiveness of dimensionality reduction. It seems that the benefit of the dimensionality reduction can be traced to the presence of tangential information in unstructured texts.

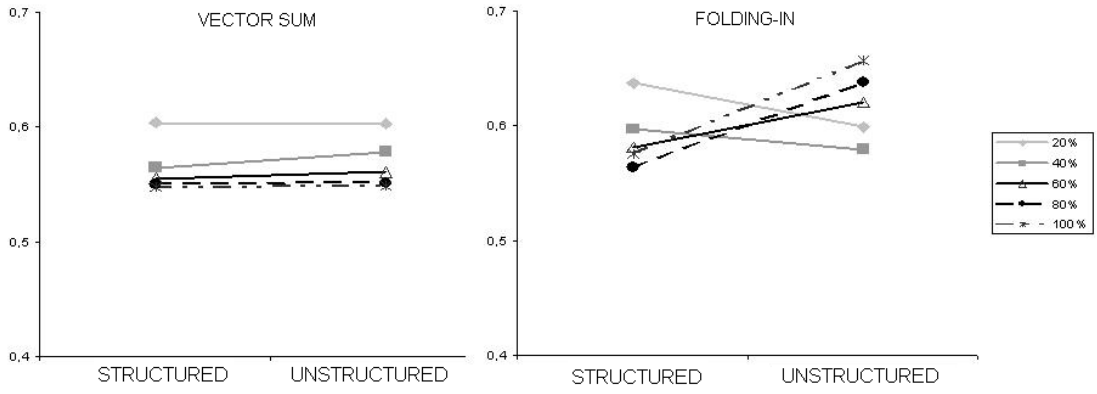


Figure 8. Interaction between Corpus and Dimensionality under the Vector sum and Folding-In condition

Secondly, we found that the effectiveness of dimensionality reduction is dependent on the level of expertise of the student evaluated. As we can see from figure 9-right (numeric values in table 9), LSA evaluation of Experts' essays tends to benefit from reduced dimensionality spaces. In the responses of such students the full dimensionality (100%) was less effective than some of the other dimensionalities, with significant differences between 100% dimensionality and 40% or 60% - these reductions offer substantial benefits.

	20%		40%		60%		80%		100%	
	Mean	SEM	Mean	SEM	Mean	SEM	Mean	SEM	Mean	SEM
STRUCTURED	0,60	0,03	0,56	0,03	0,56	0,03	0,55	0,03	0,55	0,03
UNSTRUCTURED	0,60	0,04	0,58	0,04	0,56	0,04	0,55	0,04	0,55	0,04

Table 6. Interaction between Corpus and Dimensionality under the Vector sum condition.

	20%		40%		60%		80%		100%	
	Mean	SEM	Mean	SEM	Mean	SEM	Mean	SEM	Mean	SEM
STRUCTURED	0,64	0,04	0,60	0,04	0,58	0,04	0,56	0,04	0,58	0,04
UNSTRUCTURED	0,60	0,03	0,58	0,03	0,62	0,04	0,64	0,04	0,66	0,04

Table 7. Interaction between Corpus and Dimensionality under the Folding-In condition.

Surprisingly, such dimensionality reductions coincide with the recommendation of Wild et al (2005) that the original and reduced matrices should share 50%, 40% or 30% of the dimensionality**.** In fact, 40% and 60% of the cumulated singular value obtained the best and most consistent behaviours in each group, with the greatest effectiveness for experts, and the same results as full dimensionality for non-experts (100% dimensionality is at least as good as any of the other dimensionalities for these responses).

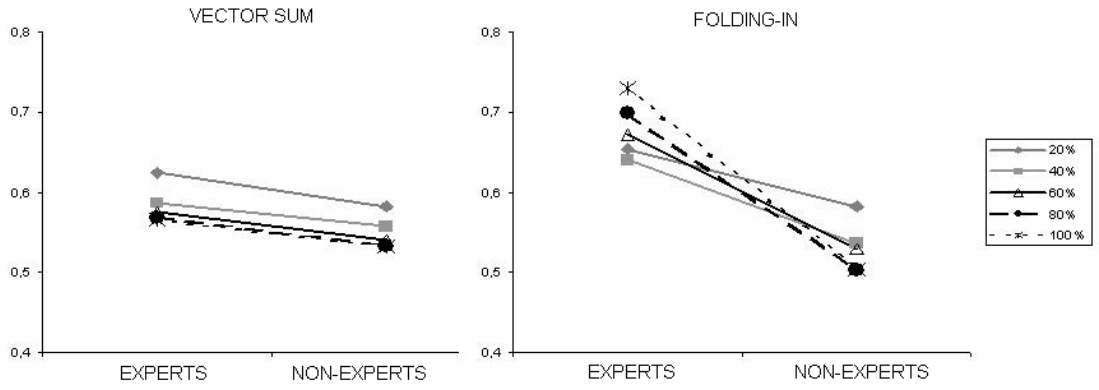


Figure 9. Interaction between Group and Dimensionality under the Vector sum and Folding-In conditions.

	20%		40%		60%		80%		100%	
	Mean	SEM	Mean	SEM	Mean	SEM	Mean	SEM	Mean	SEM
EXPERTS	0,62	0,05	0,59	0,05	0,57	0,05	0,57	0,05	0,57	0,05
NON-EXPERTS	0,58	0,05	0,56	0,05	0,54	0,05	0,53	0,05	0,53	0,05

Table 8. Interaction between Group and Dimensionality under the Vector-sum condition.

	20%		40%		60%		80%		100%	
	Mean	SEM	Mean	SEM	Mean	SEM	Mean	SEM	Mean	SEM
EXPERTS	0,65	0,05	0,64	0,05	0,67	0,05	0,70	0,05	0,73	0,05
NON-EXPERTS	0,58	0,05	0,54	0,05	0,53	0,05	0,50	0,05	0,50	0,05

Table 9. Interaction between Group and Dimensionality under the Folding-In condition.

If we consider the kind of responses offered by each group, it is possible that dimensionality reduction is useful for evaluating essays of some length and with a minimum content.

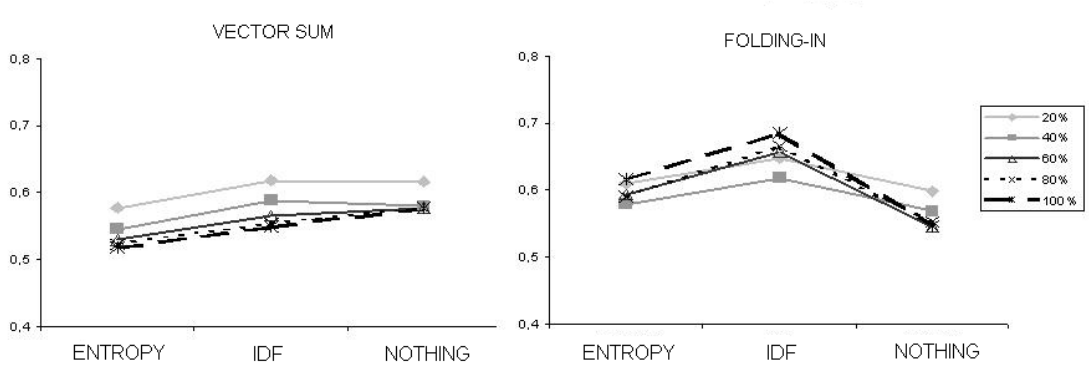


Figure 10. Interaction between Weighting function and Dimensionality under the Vector sum and Folding-In conditions.

Thirdly, we find that any benefits of Weighting tend to be enhanced when we apply a dimensionality reduction. As we see in figure 10 (numeric values in table 10 and 11), both IDF and Entropy tend to achieve the most efficient results if dimensionality reduction is applied. For these corpus sizes at least, then, we can see that the normal LSA protocol (Apply a Weighting function and reduce the dimensionality) only works under certain conditions, such when the essays have sufficient content and we use the Folding-In method.

	20%		40%		60%		80%		100%	
	Mean	SEM	Mean	SEM	Mean	SEM	Mean	SEM	Mean	SEM
ENTROPY	0,58	0,04	0,55	0,03	0,53	0,03	0,52	0,03	0,52	0,03
IDF	0,62	0,04	0,59	0,03	0,57	0,03	0,55	0,04	0,55	0,04
NOTHING	0,62	0,04	0,58	0,04	0,58	0,04	0,58	0,04	0,58	0,04

Table 10. Interaction between Weighting function and Dimensionality under the Vector sum condition

	20%		40%		60%		80%		100%	
	Mean	SEM	Mean	SEM	Mean	SEM	Mean	SEM	Mean	SEM
ENTROPY	0,61	0,04	0,58	0,04	0,59	0,04	0,59	0,04	0,62	0,04
IDF	0,65	0,04	0,62	0,04	0,66	0,04	0,67	0,04	0,69	0,04
NOTHING	0,60	0,04	0,57	0,04	0,55	0,04	0,55	0,04	0,55	0,04

Table 11. Interaction between Weighting function and Dimensionality under the Folding-In condition

6. General Discussion

The strength of correlations with human graders achieved with some of the parameter combinations in this study is remarkable. Some of these correlations (albeit with small-scale corpora) even match correlations between human scores, at least if we take Euclidean distance as our measure of similarity. But it is not easy to determine the true process behind LSA, or to establish which combination of parameters works best. Moreover, we found no clear evidence that the classical LSA protocol (using Entropy or IDF Weighting functions, dimensionality reduction and folding-in to incorporate the new texts into existing spaces) works better than some simpler version.

One piece of evidence drawn from this study is that the cosine is not the best measure for assessing academic texts, because it tends to overestimate the scores of essays with minimal length and content. This fault seems to be corrected with the use of Euclidean distances as a measure of similarity. We have called this phenomenon, previously observed in other studies (Olmos et al., in press), the "Distance corrector effect". Since Non-Expert responses do not meet minimum length and content requirements, the "Distance corrector effect" is critical in evaluating this group. Euclidean distances are more effective and consistent in all conditions, and we therefore suggest the use of this measure in place of the more commonly chosen cosines, at least, in assessment tasks.

Another conclusion is that the two methods of constructing pseudo documents (Folding-In and Vector sum) result in very different behaviour. There is no difference in terms of efficiency between the two, but the Vector

sum method is not susceptible to the benefits of dimensionality reduction. Using the Folding-In method, dimensionality reduction tends to improve results with some corpora, for some levels of expertise and using certain types of pre-processing.

The fact that the Vector sum method is not compatible with the benefits of dimensionality reduction is not necessarily a defect so long as reasonable and stable performance is obtained. However, reduction is beneficial for some efficient conditions, such as when we use Distance as our similarity measure. The interaction data showed that Distance is the best measure for comparing documents, so Folding-In and reducing dimensionality might be the best choice in these circumstances. In fact, the two best combinations in this study were achieved in a condition using Folding-In as our method for constructing pseudo documents and Distance as a similarity measure. To add to the benefits of Folding-In, it produces more variability and more extremely good results.

The sensitivity of Folding-In to interaction with dimensionality presents a quite different profile of results. In one corpus, the unstructured one, it seems that full dimensionality tends to achieve scores that are better than - or at least similar to - the most effective reductions. In the other, structured one, dimensionality reduction tends to achieve greater effectiveness.

So when does a corpus benefit from dimensionality reduction? Our data indicate that in addition to size, structure also plays a role here. By introducing first- and second-order relations (Mill & Kontostathis, 2004) into the equation, we found that less structured corpora are more favoured by dimensionality reduction. As has been argued before, structured corpora are often those that lack tangential information. This kind of corpus causes some terms to become key-terms (Franceschetti et al., 2001; Olde et al., 2002) and the occurrence of this kind of terms is of primary importance during evaluation. It is not then necessary to extract fine detail from the words through the elimination of noise, conducted by reducing dimensions. Thus 100% dimensionality tends to obtain the most effective results.

In the folding-in condition, we have also found that the effectiveness of dimensionality reduction is dependent on the student's degree of expertise. In the group of experts, reducing dimensionality tends to work more effectively and 60% and 40% of dimensionality provide the most consistent results, coinciding with Wild et al (2005) who recommend that the original and reduced matrices share 50%, 40% or 30%. In this line we have found that the optimal number of dimensions does not have to be extremely low, sometimes even approaching the 300 dimensions recommended by Landauer et al. (1997) for general domain corpora. However, in the group of Non-Experts, full dimensionality (100%) tends to achieve the same or better results than the most effective dimensionality reduction. This means that in the group of Non-Experts, assessment is focused on the occurrence of some high-frequency key-terms, without requiring a precise representation of the terms.

To summarise our findings using small-scale corpora, LSA shows great variability in its behaviour, and sometimes works better in versions that differ considerably from the classical LSA model (our data show that full dimensionality and sum of vectors is often a good combination). In spite of this, we have extracted good correlations between some versions and human graders, drawing several conclusions about the best combinations of parameters. It is difficult to determine what parameters should be used without obtaining empirical data from recurrent repetition but some patterns can be observed. For example, despite the variability found among the conditions, this study shows that the measure of Euclidean distance is more appropriate in assessing essays, and we therefore recommend increased use of this measure in the future. More simulations are necessary, but we hope to have thrown some light on the question of which parameters to apply in academic LSA implementations.

7. Acknowledgements

This work was supported by Grant SEJ2006-09916 from the Spanish Ministry of Science and Technology. The authors wish to thank Ramón Lopez-Higes, Jesús-Sanz and Jose Mª. Prados-Atienza from Universidad Complutense de Madrid for supporting the logistic of this research.

Capítulo 8

Using LSA and the predication algorithm to improve extraction of meanings from a diagnostic corpus.

(Artículo publicado en The Spanish Journal of Psychology 2009, Vol. 12, No. 2, 424-440)

The Spanish Journal of Psychology
2009, Vol. 12, No. 2, 424-440

Using LSA and the predication algorithm to improve extraction of meanings from a diagnostic corpus.

Guillermo Jorge-Botana, Ricardo Olmos & José A. León
Universidad Autónoma de Madrid.

Abstract

There is currently a widespread interest in indexing and extracting taxonomic information from large text collections. An example is the automatic categorization of informally written medical or psychological diagnoses, followed by the extraction of epidemiological information or even terms and structures needed to formulate guiding questions as an heuristic tool for helping doctors. Vector space models have been successfully used to this end (Lee, Cimino, Zhu, Sable, Shanker, Ely & Yu, 2006; Pakhomov, Buntrock & Chute, 2006). In this study we use a computational model known as Latent Semantic Analysis (LSA) on a diagnostic corpus with the aim of retrieving definitions (in the form of lists of semantic neighbors) of common structures it contains (e.g. “storm phobia”, “dog phobia”) or less common structures that might be formed by logical combinations of categories and diagnostic symptoms (e.g. “gun personality” or “germ personality”). In the quest to bring definitions into line with the meaning of structures and make them in some way representative, various problems commonly arise while recovering content using vector space models. We propose some approaches which bypass these problems, such as Kintsch's (2001) predication algorithm and some corrections to the way lists of neighbors are obtained, which have already been tested on semantic spaces in a non-specific domain (Jorge-Botana, León, Olmos & Hassan-Montero, under review). The results support the idea that the predication algorithm may also be useful for extracting more precise meanings of certain structures from scientific corpora, and that the introduction of some corrections based on vector length may increases its efficiency on non-representative terms.

Actualmente existe un amplio interés en la indexación y extracción de información provenientes de grandes bancos de textos de índole taxonómica. Por ejemplo, la categorización automática de diagnósticos médicos o psicológicos redactados de manera informal y su consiguiente extracción de información epidemiológica o incluso en la extracción de términos y estructuras para la creación de preguntas-guía que asistan de forma heurística a los médicos en la búsqueda de información. Los modelos espacio-vectoriales han sido empleados con éxito en estos propósitos (Lee, Cimino, Zhu, Sable, Shanker, Ely, & Yu, 2006; Pakhomov, Buntrock, & Chute, 2006). En este estudio utilizamos un modelo computacional conocido como Análisis Semántico Latente (LSA) sobre un corpus diagnóstico con la motivación de recuperar definiciones (en forma de listados de vecinos semánticos) de estructuras habituales en ellos (e.g., “fobia a las tormentas”, “fobia a los perros”) o estructuras menos habituales, pero que pueden formarse por combinaciones lógicas de las categorías y síntomas diagnósticos (e.g., “personalidad de la pistola” o “personalidad de los gérmenes”). Para conseguir que las definiciones sean ajustadas al significado de las estructuras, y mínimamente representativas, se discuten algunos problemas que suelen surgir en la recuperación de contenidos con los modelos espacio-vectoriales, y se proponen algunas formas de evitarlos como el algoritmo de predicación de Kintsch (2001) y algunas correcciones en el modo de extraer listados de vecinos ya experimentadas sobre espacios semánticos de dominio general (Jorge-Botana, León, Olmos & Hassan-Montero, in review). Los resultados apoyan la idea de que el algoritmo de predicación puede ser también útil para extraer acepciones más precisas de ciertas estructuras en corpus científicos y que la introducción de algunas correcciones en base a la longitud de vector puede aumentar su eficacia ante términos poco representativos.
.

1. Introduction

Latent Semantic Analysis (Henceforth LSA) is a computational model that analyzes semantic relationships between linguistic units automatically. It is currently one of the key computational models in cognitive psychology, especially in psycholinguistics, where its usage is especially high because of its suitability in a range of applications. It was first described by Deerwester, Dumais, Furnas, Landauer and Harshman (1990) as a means of *information retrieval*, but it was Landauer and Dumais (1997) who demonstrated its ability to account for phenomena related to knowledge acquisition and representation.

LSA constructs a vector space using an extensive corpus of documents, taking into account meaning and not grammar. A word or a combination of words is represented by a vector in this "semantic space". To establish the semantic relationship between two words or documents, LSA uses the cosine of the angle between the two. A cosine close to one reveals a strong semantic relationship. A cosine close to zero reveals no semantic relationship between the two words. This same principle can be applied when examining the semantic relationship between two documents or between a document and a term. Furthermore , the LSA model uses vector length or modulus of the term, which shows how well-represented the word is in the semantic vector space. In any case, the interpretation of vector length has been the subject of some disagreement (Blackmon & Mandalia, 2004; Blackmon, Polson, Kitajima & Lewis, 2002; Rehder, Schreiner, Wolfe, Laham, Landauer & Kintsch, 1998).

Psychologically speaking, inference processes in the LSA model have been formulated as the indirect relationships, the relationships between one set of words and another, beyond simply coinciding in documents (Landauer, 2002; Lemaire & Denhière, 2006; Mill & Kontostathis, 2004). With spaces drawn from LSA, it has even been possible to study the rate of knowledge acquisition relating to a term using exposure to documents in which it does not appear (Landauer & Dumais, 1997). For example, knowledge regarding the term "lion"

is acquired by reading documents, even those in which this term does not appear, regardless of whether or not they have anything to do with the semantic field of lions. This study concludes that acquisition of this type of inferential knowledge is greater for high frequency terms. In summary, these latent links between words might explain why language learning seems to take place much more rapidly than direct exposure to it would seem to allow (Landauer & Dumais, 1997). This fact can even be extrapolated to the modeling of overly literal interpretation of meaning in disorders such as autism (Skoyles, 1999) or problem solving processes (Quesada, Kintsch & Gómez-Milán, 2001).

Semantic spaces formed using LSA have also offered pleasing results in a synonym recognition task (Landauer & Dumais, 1997; Turney, 2001), even simulating the pattern of errors found in these tests (Landauer & Dumais, 1997). LSA shows that antonyms share a degree of *relative* similarity, which has led to the modeling of the psychological nature of antonymy as a type of synonymy and not in terms of the existence of *absolute* opposites (Landauer, 2002).

As with word recognition models based on artificial neural networks (e.g., Mandl, 1999; Rumelhart & McClelland, 1992; Seidengberg & McClelland, 1989), LSA vector space models contain a single vector representation of each term[15]. This has also been used in an attempt to emulate the phenomena of homonymy and polysemy. Whilst there are certain etymological differences between homonymy and polysemy, they share a common definition. We say that two words are homonyms if their signifier is the same, in other words if they comprise the same phonemes or graphemes, or their phonetic or written forms coincide. One of the ways that has been used to identify them is to break this single vector representation rule and consider each meaning or each grammatical condition as a representation in the semantic space. Wiemer-Hastings (2000) and Wiemer-Hastings & Zipitria (2001) experimented with a method that discriminated between the different morphological roles played by

[15] In fact, in the models proposed by Seidengberg and McClelland there are two entries, one phonological and another orthographic. The two ideally interact with the semantic and contextual layer, but there are no representational differences for the different meanings a term might have. In Mandl's case, the LSA vectors themselves and their single representation are the network input.

words with similar spellings. For example the word *plane* may take the value of verb or noun. The author's aim was to introduce them into the LSA space in differentiated forms, and to achieve this, each word was flagged with a termination that identified it as being one form or another. For example, *plane-VB* and *plane-NN* were inserted into the corpus to denote verb or noun. The result is that corpora flagged in this way perform worse than those that are not flagged, and this has been confirmed in other studies (Serafín & DiEugenio, 2003). This suggests that LSA takes advantage of the usage of words in different contexts. If each meaning is previously differentiated and is processed in a different way, the variability in usage of the orthographic representation of a term in LSA diminishes, and the vectors that represent it are less rich. Besides, this endorses the idea that it seems reasonable to think of a single mental representation of a term, and not the differentiated representations of its uses and meanings.

Nonetheless, Deerwester et al. (1990) indicate some limitations of LSA in representing the phenomena of homonymy and polysemy, and the disambiguation of each of its meanings depending on the context. These authors state that although the phenomenon of synonymy is faithfully represented by LSA simulations, the same is not true with polysemy. A term, even if it has more than one meaning, is still represented in a single vector. This vector has certain coordinates. Since it has several meanings, these are represented as an average of its meanings, weighted according to the frequency of the contexts where it is found. These authors provide the key: "If none of the real meanings is like the average meaning" it may create a bias in the representation, producing an entity that does not match any actual term usage. This recalls the criticisms leveled at prototype-based models which proposed the existence of a prototypical form, which was the sum of the typical features of the members of that category. The criticism argues that if the prototype is a cluster of features of a category, and bearing in mind the variability of the typical elements, paradoxically the resulting prototype used to establish similarity is in fact a very atypical member (Rosch & Mervis, 1975). This, however, is not the only criticism. Since the meanings of a signifier are extracted based on a context, we may find that the less frequent features never

gain enough weight to emerge in this context, and the most common meaning dominates. Following this line of argument, it seems plausible to think that LSA models are fairly efficient at representing some effects observed in polysemy and homonymy, but they alone are not capable of representing the phenomenon in all its aspects and extracting the exact meanings of each usage.

These and other criticisms such as LSA's inability to represent some categorization phenomena (e.g. Schunn, 1999) have led authors such as Burgess (2000), to respond to Glenberg and Robertson's (2000) criticisms by arguing that LSA is only a model of acquisition and representation and not a model of language processing. According to Burgess, LSA serves as a starting point from which models of processing might be proposed, and algorithms implemented to simulate psycholinguistic processes. Therefore the model of knowledge provided by LSA should be biased by a context that acts as a facilitator of some content, and thus simulate the processes observed in real subjects. One of these proposed algorithms that utilize LSA knowledge representation is that carried out by Kintsch (2001). The author explains how using independent representations of the actual context greatly simplifies the treatment of different meanings of words. Beyond deciding how many meanings and usages a term has or when one and not another should be retrieved, Kintsch points out that the only thing we need bear in mind is a single vector-term and a process that generates the meanings that emerge from this vector in each context (predication algorithm). Other proposals based on this approach are a simulation of the use of prior knowledge and working memory for comprehension of texts (Denhière, Lemaire, Bellissens & Jhean-Larose, 2007) and the modeling of web navigation (Juvina & van Oostendorp, 2005; Juvina, van Oostendorp, Karbor & Pauw, 2005).

2. Problems with LSA in extracting meaning: working toward precise, representative definitions.

There is currently a widespread interest in indexing and extracting taxonomic information from large text collections. One pertinent example is the

automatic categorization of informally written medical diagnoses, followed by the extraction of epidemiological information or even terms and structures needed to formulate guiding questions as a heuristic tool for helping doctors. Vector space models including LSA have been successfully used to this end (Lee et al., 2006; Pakhomov et al., 2006). Nonetheless, results from this type of models are at the mercy of the vectorial dynamics involved and the representational bias of some terms.

One of the main limitations of LSA is that it involves no analysis of word order relationships, nor of the roles terms take on within a given phrase. Perhaps for this reason, LSA is demonstrably more efficient at paragraph level, where word order plays a lesser role or is irrelevant (Kurby, Wiemer-Hastings, Ganduri, Magliano, Millis & McNamara, 2003; Landauer, 2002; Rehder et al., 1998; Wiemer-Hastings, Wiemer-Hastings & Graesser, 1999). Another limitation is that the vectorial sum calculation to represent structures involving several terms is often conditioned by how much or how little the terms are represented in the corpus. This makes it unlikely that the resulting vector represents their true meaning if any of the terms have a much lower occurrence. This is the case, for example, of predicate structures. Figure 1 shows a graphical representation of the predicate structure “the winger crossed”, with the argument (A) “winger” of much lower length than its predicate (P) “crossed”. Owing to this difference, the end result of the predication [P(A)] will be dependent on the dominant content of the predicate (P).

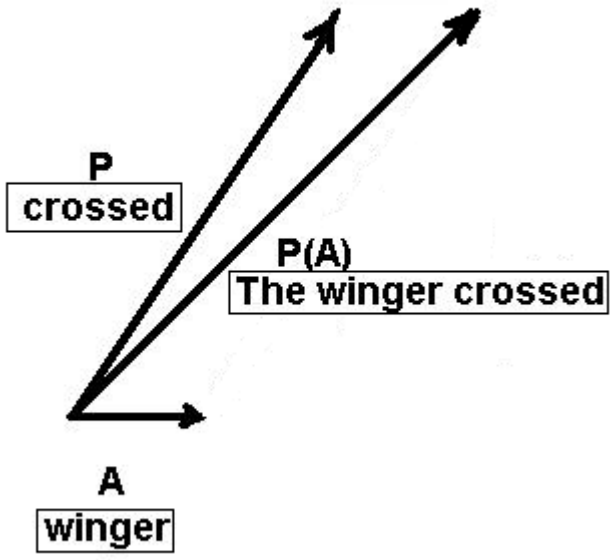


Figure 1. Centroid method. Bias in the vector sum due to length or modulus of the predicate vector.

Kintsch (2001) proposes that the exact meaning of a predicate depends on the arguments that go with it, and that both predicate and argument are constrained by a syntactic order that introduces a bias into each of them. If we take the previous example, the verb “to cross”:

Our paths crossed.
The lines crossed.
The pedestrian crossed.
The winger crossed.

All these phrases have the verb to cross as a common denominator, while this same verb takes on different meanings. We all know that in “our paths crossed” the verb “to cross” does not have the same meaning as in “the winger crossed”. The same verb acquires one or other set of properties according to the arguments that accompany it, in other words the properties that give meaning to this verb are dependent on the context formed by its arguments.

Let us take the proposition PREDICATE [ARGUMENT], assuming that the predicate takes on some set of values depending on the arguments. Both PREDICATE and ARGUMENT would be represented by their own vectors. To calculate the vector that represents the whole proposition, the common form of LSA would simply calculate a new vector as the sum or the “centroid” of the ARGUMENT vector and the PREDICATE vector. Thus, if the representation of the vectors according to their coordinates in the LSA space were:

PREDICATE vector= {p1,p2,p3,p4,p5,…,pn}
ARGUMENT vector= {a1,a2,a3,a4,a5,…,an}

Then, the representation of the whole proposition would be:

PROPOSITION vector = {p1+a1, p2+a2, p3+a3, p4+a4, p5+a5,…, pn+an}

This is not the best way to represent propositions, as it does not take into account the predicate's dependence on the arguments. In other words, to compute the vector of the entire proposition we do not need all the properties of the PREDICATE (to cross), only those that relate to the meaning of the subjects (paths, lines, pedestrian, winger). What the centroid or vectorial sum does using the LSA method, then, is to take all the properties - without discriminating according to arguments - and add them to those of the argument. Among the resulting effects is LSA's poor ability to represent the phenomenon of polysemy (Deerwester et al, 1990). All the properties of the verb “to cross” should be taken into account when it comes to calculating the new vector. If, as in the above example, the argument has a much lower vector length than the predicate, and other arguments are better represented in the predicate, the vector that represents the predication will not capture the actual intended meaning. The meaning will be closer to the sense of the predicate most represented in the LSA space. With this simple vector sum method, the length of the term-vectors involved dictates which semantic properties the vector representing the predication will take on.

Therefore we can assume that the centroid method fails to account for the true meaning of certain structures, and tends to extract definitions of a given structure that are subordinate to the predominant content. For the sake of argument we have chosen to name this problem Predominant meaning inundation.

Another common problem is that even when the pertinent meaning of the structure has been retrieved, the list of pertinent terms is not representative enough for the possible definition to cover all aspects. A previous study (Jorge-Botana et al., under review) confirms that when extracting semantic neighbors we obtain only neighbors that have low representativity in the semantic space, meaning that extraction of neighbors with the cosine needs to be corrected. The neighbors extracted using the cosine normally have a perfect positive association with the term they are extracted from, in other words in the corpus the neighboring terms always occur with the term in question, but never in its absence. This capacity for representativity of terms, phrases and paragraphs has also been formalized in previous studies such as that of Kintsch (2002). Kintsch compared different structures from the texts (headings, sub-headings and paragraphs), in order to find structures that themselves represent the other parts of the text. It can be thought of as extracting the abstract representation of the macrostructure. Even in the field of academic assessment, it has been suggested that the simple measure using the cosine is not enough to determine the extension of knowledge the author has of a trial, and should be enriched with some other measure of representativity (Rehder et al., 1998). We have decided to name this representativity problem for semantic neighbors Low-level definition. In summary, if the definitions (in the form of a list of semantic neighbors) do not show some degree of representativity, the sense retrieved will be too restricted.

Thus, to obtain a good definition of a structure such as “paranoid personality”, we need to retrieve semantic neighbors that both match the structure and are representative to an acceptable level. For this reason we must bear in mind the impact of the aforementioned effects, defining them in order to

operationalize the aims and general procedure of this study: Predominant meaning inundation and Low-level definition.

3. General aims

The aims of this study are concerned with solving the problems described above, but this time in a semantic space that represents clinical diagnoses and descriptions.

For the first problem (Predominant meaning inundation), we will use an adaptation of the predication algorithm (Kintsch, 2001) to filter out irrelevant content. The difference between applying this type of algorithm on generalist corpora (where it has already been applied) and scientific corpora (used for the first time in this study), is that the former contain representations of terms with different meanings that are totally independent of one another, forming pure polysemic structures. In contrast, this type of entries seldom appears in scientific corpora, as all content is restricted to the topic in question. However, it is possible to simulate the extraction for some structures where a particular term may express different meanings, even though these are not completely independent of one another. As in the example “a bird is a pelican” (Kintsch, 2001), “storm phobia” may represent one such structure in which the meaning of the word “phobia” takes on one of a range of meanings according to the context, in this case the word “storm”. With the use of this algorithm we predict that the content retrieved from this type of structures is closer to what is in fact sought.

For the second problem (Low-level definition), we will use a correction to the neighbor extraction mechanism. Whilst semantic neighbors are normally extracted using the cosine, we will correct the measure by introducing vector length as a modulating factor. The aim is for these same neighbors to have a more representative content in the semantic space used, and not be

constrained to a perfect positive correlation with the term in question[16]. Combining this technique with the cosine method, we predict that the terms extracted will better cover all topic-related content.

4. General procedure

We will take a semantic vector space produced by LSA, using a psychopathological corpus to extract content in different ways. We will begin by extracting neighbors for isolated terms, without any other accompanying term that might modulate their meaning toward a sub-category. This will provide a baseline and give us an idea of the predominant content of the terms. In addition we will extract the neighbors, correcting the cosine using vector length. This will show the effect of the correction on neighbors retrieved, and indicate how we might avoid the problem of Low-level definition described in section 3.

Secondly we will extract semantic neighbors for structures such as "airplane phobia", where the first term modulates the meaning of the second. As a baseline we will use the centroid (simple sum of vectors - see figure 1) and predict that the content extracted will be very similar to the predominant content extracted previously (problem referred to as Predominant meaning inundation). We will also use an adaptation of the predication algorithm (Kintsch, 2001) to improve the results. To avoid the problem that terms in the list are not sufficiently representative of the subject matter (Low-level definition), we will also apply a correction to the cosine using vector length.

Lastly, all neighbors extracted from each of the complex structures using each of the different methods will be compared (using an ANOVA) with definitions extracted from digitalized texts relating to mental disorders. In this way we will be able to see whether different meanings extracted under each condition are better matched to the meaning sought - in other words whether

[16] Perfect positive correlation: Relationships where the neighbors always occur with the term whose neighbors we seek to extract, and never in its absence.

the predication algorithm really is sensitive to the nuances that each of the arguments introduces into the predicate.

5. Simulation

5.1 Semantic space for testing

For this experiment a domain-specific corpus was created: a scientific corpus based on the classification and description of mental disorders following the DSM-IV structured classification system, together with 900 paragraphs of digitalized psychopathology obtained from Internet. After cleaning up the corpus and applying the entropy-based pre-process (see Nakov Popova & Mateev, 2001 for a review), the semantic space is defined by 5,335 terms in 959 paragraphs. We used a dimension reduction criterion saving the 40% of the accumulated singular value (Wild, Stahl, Stermsek & Neumann, 2005), leaving us with 187 dimensions. The average cosine of similarity between terms was 0.018, and the standard deviation 0.074. Both LSA and the predication network are calculated using GALLITO, an LSA-based application implemented using .NET (C#, VB.NET) integrated with Matlab technology. The system used to extract the examples shown below is available at http://www.elsemantico.com and can be used to test the different ways of extracting semantic neighborhoods[17].

5.2 Simulation I: Structures of a single term

5.2.1 Parameters

One way to show the meaning or meanings of a word is by listing the semantic neighbors closest to the word, thereby bringing together all the terms that are distributed in its vicinity. In this way, we obtain both the dominant meaning of the word, although this may be in the form of a scale, as well as other less common meanings. To extract these semantic neighbors we need a procedure that calculates the cosine between this term and other terms in the

[17] Both the GALLITO application and the system for extracting semantic neighborhoods is available at the Latent Semantic Analysis Interest Group's website www.elsemantico.com

semantic space, and keeps a record of the *n* greatest values in a list. The end result will be a list of the *n* most similar terms to the selected term. At the same time, we can alternatively give priority in this list to the neighbors that are best represented in the semantic space. To do so, we propose using vector length to correct the formula that compares each pair of terms. Vector length is good measure to transform the cosine for several reasons. Its use may be very efficient if we wish to ensure that the neighbors extracted are not firmly tied to the term in question, but nonetheless maintain their relationship with the word. Although some authors have identified this measure with frequency or familiarity (Blackmon & Mandalia, 2004; Blackmon, Polson, Kitajima & Lewis, 2002) others consider vector length to be a richer, more complex measure than frequency itself, especially when working on a scientific corpus (Rehder et al, 1998). These authors draw our attention to the protocol of LSA in specific domains (such as our own) in order to understand what vector length in fact represents:

A) The analysis is composed solely of fragments that represent a specific domain. Thus words that are not used in this topic cannot affect the measures, vector length included.

B) Less common words from the texts (including technical terms) are weighted during the pre-process using Entropy or IDF. This gives them a higher weighting than the more common terms, the assumption being that these will be the words that differentiate one text from another. Weighted words increase the vector length.

C) Before the analysis, the high frequency words in the language such as the function words (stop words) are eliminated. Based on these observations the authors summarize as follows: Vector length is a strong positive function of the number of less common (technical) words in the domain, a moderate positive function of common words in the domain, and a function that is not related with words that do not belong to the topic in question. Therefore, in domain-specific corpora, weighting based on the vector length may sometimes be a way to select terms that adequately represent the content of the topic in question, as well as having a minimum frequency. This avoids them being tied only to the term in question and only involved in perfect positive associations. In other words, it avoids terms that co-occur only in the same documents as the term in question, and that never occur in its absence.

Based on these observations, two different ways of extracting semantic neighbors are proposed.

(1) COSINE: Similarity = Cos(A, I).
(2) CORRECTED COSINE or Confidence = Cos(A, I) * log (1 + VectorLength(I)),

where

A = The vector that represents the term whose list of neighbors we seek to extract.
I = The vectors that represent each of the terms in the semantic space.

Formula (1) is the simple comparison of vectors using the cosine. In formula (2), 'neighbor' vector length is introduced as a correcting factor[18]. Thus the second formula gives the most representative terms from the semantic space priority when it comes to consider neighbors, but does not totally exclude the rest.

5.2.2 Extraction of neighbors.

The neighbors extracted using the cosine measure without correction show that the dominant meaning of the term "phobia" is "social phobia". This is clear because the neighborhood extracted is related with social phobia: the Spanish terms for "social", "public", "shyness" and "blushing" (see figure 2-left), and because components of other types of phobia are absent. In contrast, with the correction based on vector length, the closest neighbors tend to designate content that is more common to all types of phobia (see figure 2-right). The first positions hold Spanish terms for concepts such as "avoidance", "exposure", "specific", "agoraphobia", "phobia" and "crisis", and terms such as "situations" and "fear" move to higher positions. In contrast, more concrete terms such as

[18] This second formula makes reference to the level of *confidence* that the neighbors extracted reach a minimum level of representativity in the semantic space. Or put another way, given a similarity between the term in question and a term from the semantic space, to what extent can we be sure that the similarity is not due to the chance appearance of the second.

"blushing", "shyness", "humiliating", "girls", "embarrassing", "shops" and "meetings" lose their status, all of them being terms more closely linked to the dominant meaning in this semantic space: social phobia. With the correction based on vector length a more representative neighborhood of topics related to the key term is obtained. Definitions using this type of extraction have a broader range in hierarchical terms than those using the cosine without correction. In this way we avoid what we called Low-level definition.

Figure 2. Neighbors of phobia with the cosine on the left and the corrected cosine on the right. The 21 semantic neighbors are ordered from greatest to lowest similarity in a clockwise direction (phobia is the most related term, then social, etc.). The blue area represents the vector length of each of the terms on a scale of 1 to 5. Greater areas represent terms with greater vector length and representativity in the semantic space.

In the other example, the neighbors of "storms" extracted with the cosine seem to be terms at the same level within the definition, in other words "cliffs", "bridges", "injections", "airplanes" and "snakes" (see figure 3-left). These can be looked on as types of situations that phobics fear. However, using the correction based on vector length, the neighbors obtained better represent the general topics of psychopathology and designate higher categories such as "fear", "sub-type" or "phobia" (figure 3-right). Note too how the term "storms" itself is not among the first positions, replaced instead by terms whose meaning is more general. If we only considered in mind the most local co-occurrences (such as the cosine), "storms" would be the neighbor most closely related with "storms", as they evidently coincide in each of the documents. However, as preference is given to other characteristics of the possible neighbors, terms with lower but more representative contingencies gain ground. Thus the Low-level definition effect is again avoided.

Figure 3. Neighbors of storms with the cosine on the left and the corrected cosine on the right. The 21 semantic neighbors are ordered from greatest to lowest similarity in a clockwise direction. The grey area represents the vector length of each of the terms on a scale of 1 to 5.

5.2.3 Conclusion

There are differences between neighbors extracted with the two methods. Extraction of neighbors using the cosine seems much more concrete and restricted to terms that are not very representative in the semantic space, whilst that obtained with the correction based on vector length is more generic and represents content more representative of the knowledge domain. Thus both meanings can be extracted to complement each other. Whilst the first method detects similar relationships in terms of levels of concretion, the second establishes relationships with those terms that provide more information about the complete domain. Another observation that may be drawn from this simulation is that the usage of correction based on vector length seems more beneficial for words whose vector length is small (“storms”), although it may also provide information on the most representative words in the corpus (“phobia”).

5.3 Simulation II: Two-term predicate structures (centroid and predication algorithm)

5.3.1 Theoretical framework of the predication algorithm.

The predication algorithm developed by Kintsch (2001) seeks to resolve the limitations of the centroid or vector sum method when extracting the meaning of predicate structures. As explained in section 2, this method is dependent on the vector lengths of predicates and arguments, and normally favors only the predominant content of the terms. What the predication algorithm does is to bias the vector length, adding an adequate context for the argument type we are predicating. This context comprises the semantic neighbors of the predicate, which are also related to the argument. This is what Kintsch (2001) refers to when he points out that in any predication the predicates become dependent on their arguments, and this approach should be adopted in some way in models that attempt to formalize clause processing. The procedure is both ingenious and simple, although its use is hard to implement owing to the difficulty in identifying this type of structures and their components. Returning to the example from section 2 ("the winger crossed"), the steps we need to follow might be as follows:

6) Identify predicate ("to cross") and argument ("winger") within a proposition. P(A).
7) Extract the *n* semantic neighbors closest to the predicate (P). Given a semantic space, the cosines of P should be calculated with each of the terms that make up the semantic space. Once this step is performed, the first *n* terms closest to P are selected (the choice of n is open to the type of model sought and to empirical observations).
8) The cosines between each of the *n* chosen neighbors of P and the argument (A) are calculated.
9) A connectionist network is implemented with the terms P, A and the *n* selected neighbors of P as nodes. Besides this, there are inhibitory connections between the *n* neighbors of P (which compete with one another for activation) and excitatory connections between each of the *n* neighbors with argument (A) and predicate (P). The strength of the connections is established according to the base value of the cosines calculated in steps 2 and 3. In short, the aim is to objectively locate those

semantic neighbors in the vicinity of the predicate (P) which are also pertinent to the argument (A), and a network is implemented to this end. This network needs no previous training, as the corpus that was processed with LSA is what creates it.

10) The network is run and left to settle into a stable state. For this we can use Kintsch's own (1998) CI model.

11) The final step is to calculate the vector P(A) with the vector sum of the Predicate (P), plus the Argument(A), plus the k terms that receive most activation in the network - in other words, those that receive more excitatory activation from Predicate and Argument and least lateral inhibition from the terms in their own layer. k again depends on the type of model in use and the empirical observations carried out *a posteriori*. Once this is done, we will finally obtain a vector P(A) which will have the meaning covered by the predicate and its accompanying argument, as the final sum will also incorporate vectors of terms that are pertinent neighbors of the predicate and hence of the argument too.

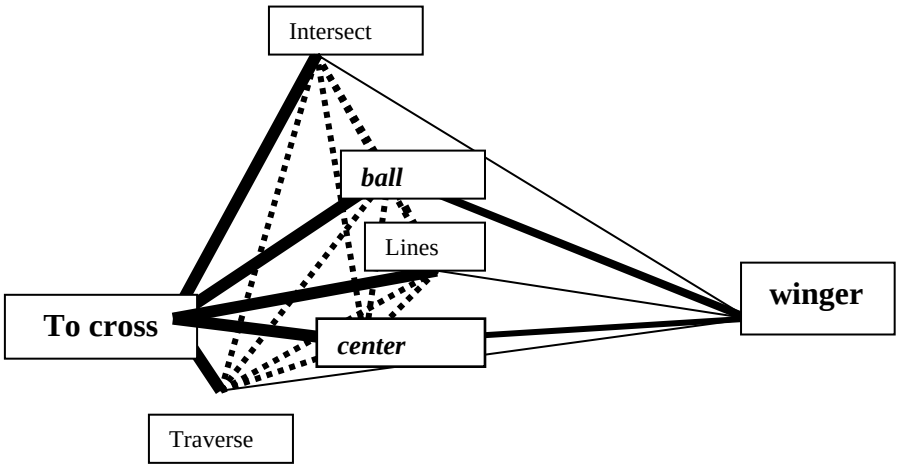


Figure 4. Graphical representation of the predication algorithm

Figure 4 shows how results might look using the predication algorithm for the proposition “the winger crossed”. The semantic neighbors extracted from the verb “to cross” might include “intersect”, “ball”, “lines”, “center” and “traverse”. Of these semantic neighbors those that would finally be most strongly activated are those that receive greater excitatory connections from both sides. In other words, given their connections with Predicate and Argument, the words that have high cosines on both sides will be those that are most strongly activated, and will send inhibitory connections to the rest. In this hypothetical case, the terms “center” and “ball” will be most strongly activated, since they are the terms most closely related semantically to the argument “winger”. Thus, “ball” and “center” will be the words that are added to the Predicate (P) “to cross” and the Argument (A) “winger”, to give the vector of the whole proposition. In this way, a bias is imposed on the standard centroid method such that it contemplates the linguistic phenomenon that the meaning of the predicate is dependent on the information provided by its arguments.

Kintsch (2000) shows the algorithm at work and checks the final meaning of predications such as "The bridge collapsed", "The plan collapsed" and "The runner collapsed", as well as an example better suited to taxonomic or hierarchical structures - "Pelican is a bird" and "The bird is a pelican". Besides this, Kintsch illustrates how the predication mechanism itself may be useful for modeling the understanding of metaphors Kintsch (2000), and even investigates the difference between metaphors that are simple and difficult to understand based on the predication parameters (Kintsch & Bowles, 2002).

5.3.2 Implementation

Our aim with this network study is not to reproduce the network proposed by Kintsch (2001). Our network contains some implementation differences, although the operations carried out are functionally similar. In our study we attempt to assign connection weights according to the cosines obtained and activation of nodes according to a rule that favors bilateral activation from both Predicate and Argument.

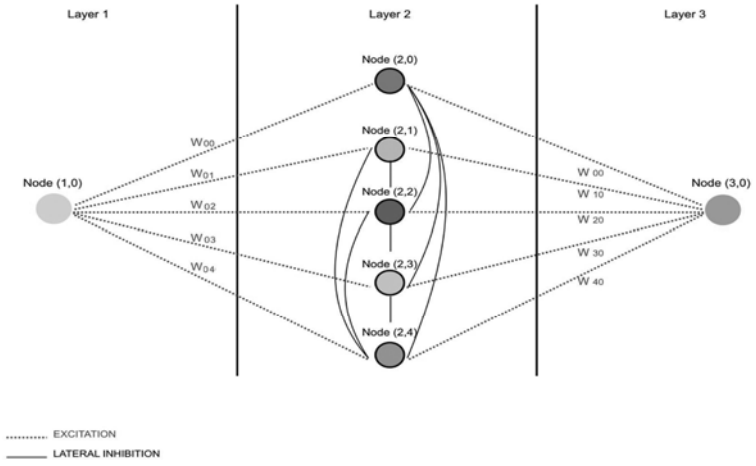


Figure 5: Network, layers and nodes.

The version of the predication network implemented here comprises 3 layers, although two of them (the first and third) have only one Node (see figure 5). These nodes in the first and third layers are those related to the Predicate (1,0) and the Argument (3,0). The central layer consists of as many Nodes as Predicate neighbors are to be contemplated in the algorithm, subject to empirical factors. Each Node in this second layer represents a term from the semantic neighborhood of the Predicate. Besides this, the Nodes in the central layer have two activation mechanisms. The first is the inter-layer activation mechanism, ensuring that each node is activated by both members of the predication, increasing its similarity index with each of them. Each of the connection weightings between a predicate and one of the central nodes is represented by the cosine between the predicate and the term each central node represents (W00, W01, W02, W04...W0N). Similarly, the connections between the argument and each node in the central layer (W00, W10, W20, W40...WN0) will be equivalent to the cosine between the terms of each of the central nodes and this argument. The second mechanism is that of lateral inhibition (intra-layer), whereby each node is inhibited by every one of its neighbors in the central layer. In this way, each node competes against the others.

Once the Inter- and Intra-layer activations are calculated, a global activation index will be obtained for each of the nodes in the central layer. Ordering these from highest to lowest, the first *n* nodes are chosen. With the terms that represent these nodes plus the predicate and argument terms, the centroid is calculated, in other words the sum of all vectors of these terms, thus obtaining the resultant predication vector P(A).

5.3.3 Parameters

There are some unresolved issues concerning the calculation of the predication algorithm, which are subject to empirical observations and possibly dependent on the type of semantic space being processed. The first issue is choosing the number of neighbors of the Predicate selected to configure the network - the first *n* neighbors are those that will participate in the network. Kintsch and Bowles (2002) acknowledge that this size may vary considerably depending on the relationship between predicate and argument, recommending that *n* should be around 20. This figure rises to 500 in the representation of predicative metaphors, given that the relationship between predicate and argument is looser and the crucial terms that are pertinent to both are not often found among the first 100 neighbors of the Predicate. In our case we have set *n* as 10% of the total number of terms in our space. Our decision to adopt a variable figure initially (it may later be reduced) is based on the observation that *n* is also linked to the size of the semantic space. A second issue is the number of activated term nodes (*k*) whose vector is taken to form the final representation of the predication P(A). Kintsch and Bowles (2002) suggest that the figure that gives best results is around 5. Considering a greater number introduces an unnecessary risk, as the resultant semantic representation (the meaning) would be clouded by the influence of spurious values. In contrast, taking only a very small number would mean the loss of crucial information. We follow this recommendation and make *k* equal to 5. Another significant issue open to debate is the node activation rule, particularly concerning the part of this activation that derives from inter-layer connections. The activation value of each node derived from the inter-layer connections may be calculated using

only the cosines of the predicate and argument with each of the nodes. It may also be corrected by manipulating values such as vector length of Predicate and Argument, standard deviation between both cosines or the vector length of the neighbors of the Predicate that are introduced into the network.

In this case we will use some of these parameters in our formula. **P** represents the predicate, **A** the argument and ***i*** each of the term nodes from the central layer. The weight of the inter-layer connections are ***Cos(P,i)*** and ***Cos(i,A***), ***Cos(P,i) being*** the cosine between the vector of the Predicate Term (P) and the Vector of each Term Node(i), and ***Cos(i,A)*** the cosine between the Vector of each Term Node(i) and the vector of the Argument Term(A).

The most basic form of excitatory activation of the nodes would be

Activation = ***Cos(P,i)+ Cos(i,A).*** In other words, each central node will be activated more or less depending on the activation received from both connections (based on the weights of both connections).

However, after exploring several possibilities with the parameters described above (vector length of Predicate and Argument, standard deviation), we chose to use the following formula:

Activaton=***Cos(P,i)***+***Cos(i,A)****(1+log(VectorLength(P))+(1/(SD(***Cos(P,i)***,***Cos(i,A)***)+0.5))

The justification for using this formula is as follows. The difference between the vector length of the predicate and of the argument may be excessive, favoring the former and preventing the predication algorithm from extracting the true meaning. For this reason the formula should be corrected, multiplying the cosine between each Node and the argument by a weighting based on the vector length of the Predicate. In this way, the argument plays a greater role in activation, its participation being directly proportional to the vector length of the predicate. At the same time, the correction based on standard

deviation is introduced in order to promote activation of term nodes whose two cosines (between Predicate and Argument) are similar - in other words, not to promote nodes that receive unilateral activation.

5.3.4 Procedure

We will use the last formula from the previous section to extract lists of neighbors for two-term structures in which the first will act as predicate and the second as argument. The structures to be used are “fobia a las tormentas” (storm phobia) and “personalidad de la pistola” (gun personality). In addition, we will extract neighbors of these same complex structures with the simple sum or centroid, using these conditions as a baseline. As in simulation 1**,** we will use correction based on vector length, for both forms of extraction including those that use predication[19] and those that use centroid. The lists of neighbors will be extracted in four ways, using the following combinations (see table 1):

Structure for analysis of examples

		Correction based on vector length	
		No (normal version)	Yes (corrected version)
Predication	No (Centroid)	Uncorrected centroid	Corrected Centroid
Algorithm	Yes (predication Alg.)	Uncorrected predication	Corrected predication Alg.

Table 1. Structure for analysis of examples.

5.3.5 Extraction of neighbors

To begin with, we extract the semantic neighbors of the proposition “fobia a las tormentas” (storm phobia). The left-hand graphic in figure 6 shows how using the vector sum of both terms (from now on centroid) we observe the

[19] Once we have obtained the vector that represents the predication P(A) - in other words, the vector sum of the predicate (P), argument (A) and five nodes with highest activation - we then extract the twenty-one first semantic neighbors of P(A). We apply the correction based on vector length during this extraction process. It should be borne in mind that the node activation formula takes into account the vector length of the predicate, but that this formula does not represent what we refer to in this section as the correction based on vector length. The correction based on vector length is applied after calculating the predication vector.

Predominant meaning inundation effect. The essence of this problem is that the neighbors extracted belong to the dominant subject matter of “phobia” (“social”, “shyness”, “public”, etc.) even when we specify that the phobia relates to storms. The predominant sense of “phobia” may be seen in Figure 2, where we show that neighbors of “phobia” alone are related to the domain of social phobia. Using the predication algorithm (right-hand part of figure 6), this predominant sense gives some ground to meanings more in line with specific phobias, (“spaces”, “cliffs”, “bridges”, “snakes”, “specific”) - more coherent with “storm phobia”.

6. Neighbors for “fobia a las tormentas” (storm phobia): Centroid without correction on the left and predication without correction on the right. The 21 semantic neighbors are ordered from highest to lowest similarity in a clockwise direction. The blue area represents the vector length of each of the terms on a scale of 1 to 5.

As for the correction based on vector length (figure 7-right, corrected predication), we can see that the neighbors have greater vector lengths, allowing the definition of the predication to contain more representative terms and thus avoiding the problem of Low-level definition. Nonetheless, this other form also produces a definition containing terms related to the predominant sense such as “social” (Predominant meaning inundation).

In view of the neighbors extracted with the more efficient predication methods, the idea that the bias introduced when representing this predication reveals its true meaning seems to lose weight. In the case of “storm phobia”, the term for “storms” introduces parameters that modulate the general, dominant meaning of “phobia”. In this case, the phobia must have connotations which differ from those of “social” phobia, but must conserve the general meaning common to all phobias, a meaning that might be defined using terms

such as “fear” or “situational”. This common general meaning seems more palpable when the representation of the predication is calculated, and its neighbors are extracted taking into account their vector length. Using this method, the meaning of the predication makes reference to more representative terms.

Figure 7. “Storm phobia” neighbors: Corrected centroid on the left and corrected predication on the right. The 21 semantic neighbors are ordered from greatest to lowest similarity in a clockwise direction. The blue area represents the vector length of each of the terms on a scale of 1 to 5.

From the further examples simulated we have chosen one which is closer to natural language and not simply taxonomic in nature.

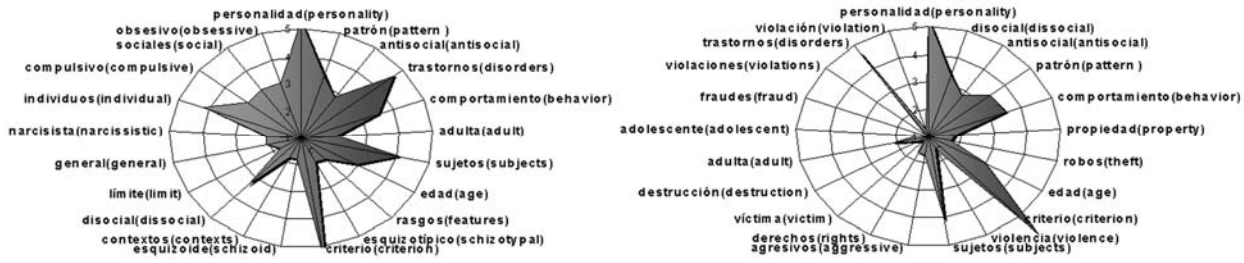


Figure 8. “Gun personality” neighbors: Centroid without correction and Predication without correction. The 21 semantic neighbors are ordered from greatest to lowest similarity in a clockwise direction. The blue area represents the vector length of each of the terms on a scale of 1 to 5. Greater areas represent terms with greater vector length and greater representativity in the semantic space.

In the field of psychological and psychiatric pathology, suppose that we wish to metaphorically designate a violent, maladjusted personality type a “gun

personality". If we told someone that an individual has a "gun personality", they might understand what we are referring to if they have a domain-specific mental model similar to the one LSA uses, even without having received any kind of explanation. This pseudo-metaphorical language can be captured with the same mechanisms that are used for predication. The mechanism for capturing the metaphorical meaning of the structures is still influenced by the introduction of contextual bias (one term exerts influence on another) (Kintsch, 2000; Kintsch & Bowles, 2002). Here the word "pistola" (gun) introduces a bias with respect to the broad sense of personality, provoking the activation of content referring to a specific type of personality - in this case an antisocial personality. Taking the neighbors extracted with the Centroid method as a baseline we can see that the structure "gun personality" takes on a meaning much closer to reality if we use the predication algorithm in its corrected or uncorrected version. In the uncorrected Centroid version (figure 8-left) the first positions contain terms belonging to other types of personality disorder such as "schizotypical", "schizoid", "anancastic", "eccentric" or "introverted" (in other words, we can observe the so-called Predominant meaning inundation). This seems to be partially rectified if we use the Corrected centroid (figure 9-left) although terms such as "schizotypical", "schizoid" or "narcissistic" still appear. However, with both versions of Predication (figure 8 and 9 right), it seems that the meaning of "gun personality" takes on connotations more in line with its potential meaning. In both versions, the terms that appear in the list of neighbors belong to the category "antisocial personality disorders", closer to the true meaning. In the uncorrected version of Predication it seems that the neighbors are less representative of the content. Although general personality terms such as "Personality" and "dissocial" are conserved, terms such as "fraud", "prohibition", "falsification", "possessions", "extortion" and "knife" appear (low-level definition effect). Therefore, the algorithm has been responsible for this mix of terms, with a bias introduced by the order (first "personalidad" then "pistola") and the role of its constituents (Predicate and Argument). This becomes clearer in the corrected predication version (figure 9-right). In this version terms such as "schizoid", "schizotypical", "boundaries", "narcissistic" and "avoidance" disappear, but general properties of personality remain such as "pattern", "behavior" and "personality". Other representative terms (with a certain vector

length) restricted to the field of the antisocial personality make their way into the list, such as “theft”, “violence”, “property” and “aggressive”. In addition, the following positions contain terms with lower vector length such as “possessions”, “fraud”, “damages” and “extortion”.

Figure 9. “Gun personality” neighbors: Corrected centroid on the left and corrected predication on the right. The 21 semantic neighbors are ordered from greatest to lowest similarity in a clockwise direction. The blue area represents the vector length of each of the terms on a scale of 1 to 5.

5.3.6 Conclusion

In comparison with our respective baselines (neighborhood of isolated terms or neighborhood of the two words using the centroid method), the two predication methods (corrected or uncorrected) seem to perform correctly in terms of avoiding the effect we termed Predominant meaning inundation. In addition, the corrected predication method seems to do so avoiding the Low-level definition effect, although the benefit is greater in the predication whose argument has a lower vector length (“gun personality”). In the case of this second structure, the predication algorithm seems to reveal a phenomenon common in natural language - that a term of a much lower hierarchical level metonymically identifies the content of higher-level structures.

6. Experiment: comparison with real definitions

6.1 Aims

Once the lists of semantic neighbors of the composite structures had been extracted (section 5.3.5 above), we proposed checking whether results from the different methods used match a sample of real psychopathological definitions. In this way we are able to analyze whether the meaning of each list resembles the content to which it refers: “storm phobia” as a specific phobia and “gun personality” as a possible means of designating an antisocial personality. This will also help to support the claims made in previous sections regarding the lists extracted with the predication method in terms of avoiding both of the effects that concern us here (Predominant meaning inundation and Low-level definition).

6.2 Materials

The definitions for checking the neighbors extracted from “storm phobia” will be associated with one of the following themes: general concept of phobia, social phobia, specific phobia and generalized anxiety. Eight definitions will be sought for each of these areas. Similarly, to check “gun personality” eight definitions will be sought for four themes: general concept of personality disorders, schizoid personality disorder, avoidant personality disorder and antisocial personality disorder. The definitions will be extracted from specialized texts in digital format based on the DSM-VI and ICE-10 published on the Internet. The average size of the definitions with which “fobia a las tormentas” will be compared is 89.62 words with a standard deviation of 41.77. The average size of the definitions with which “gun personality” will be compared is 72.63 and the standard deviation is 29.32. Of these words, only those contained in the semantic space will be taken into account in the comparison. In other words we consider only those that remain after the preprocessing carried out before instantiating the occurrence matrix. None of these definitions formed part of the corpus used to train LSA, although they do

belong to the same subject area, as they are also psychopathological diagnoses based on ICE-10 and DSM-IV.

6.3 Method

The method will be as follows: Each list of neighbors extracted from the LSA system (the lists from section 3.3.5 in figures 6, 7, 8 and 9) comprise the definitions that LSA has of a term. For example, in the experiment for “storm phobia” (section 3.3.5) we have four lists created with the four methods: Centroid, Centroid corrected using vector length, Predication and Predication corrected using vector length. With these four lists of neighbors we draw up the four documents from “storm phobia” (see figures 6 and 7 for the list of words that make up each document). These four documents are compared with each of the 8 real documents chosen to represent the general concept of phobia, the 8 for social phobia, the 8 for specific phobia and the 8 for generalized anxiety. This allows us to later calculate the averages of the eight scores (converting the texts and lists into pseudodocuments and using the cosine). From these averages we will extract the gradients that show the different meanings offered by each of the structures (“storm phobia” and “gun personality”) using each of the methods. The ideal aim would be for the “storm phobia” documents to be closer to the definitions of specific phobia, even though they conserve some similarity with the definitions of social phobia and phobia in general. Similarly, the ideal aim for the documents from “gun personality” is a greater similarity with definitions of antisocial personality disorder, in addition to a similarity with other kinds of personality disorders. To objectively check that the gradients show optimum discrimination, we will check that the differences between the similarities with each group of definitions are significant, using two ANOVAs. In each of the ANOVAs we will be testing two factors. On the one hand, the method of extraction of neighbors (with 4 levels: Centroid, Centroid corrected using vector length, Predication and Predication corrected using vector length). And on the other, the texts or groups of definitions referring to disorders (with 4 levels according to the group of definitions: phobia, social phobia, specific phobia and generalized phobia).

6.4. Results and discussion

6.4.1 "Fobia a las tormentas" (storm phobia)

To evaluate the results a two-factor ANOVA was carried out: (1) Method of extraction of neighbors (with 4 levels): Centroid, Centroid corrected using vector length, Predication and Predication corrected using vector length. (2) Standard psychopathology texts (4 definitions): General concept of Phobia, Social Phobia, Specific Phobia and Generalized Anxiety. There is a main effect for the type of text ($F(3,84) = 99.22$, $p < 0.05$) and for the method ($F(3,28) = 22.91$, $p < 0.05$). An interaction effect can be observed ($F(9,84) = 11.32$, $p < 0.05$) as shown in figure 10. In both centroid conditions the list extracted is most similar to the paragraphs on social phobia, which seems to be the predominant content of the term "phobia" in this corpus, even though the only significant difference is between the Generalized Anxiety texts and those of the other conditions ($p < 0.05$). This same profile can be found in both the corrected and the uncorrected versions - note that the correction hardly produces a differential effect. Nonetheless, applying the predication algorithm the gradients change, producing a more marked similarity with the paragraphs on specific phobia. The pair comparisons show that in the uncorrected predication algorithm condition (marked as uncorrected PRE), the similarities are greater with the definitions or texts on specific phobia compared with social phobia and generalized anxiety ($p < 0.05$). Besides, no significant differences were found between the similarities with texts on specific phobia and those on the general concept of phobia in this condition ($p = 0.16$). In the three remaining conditions we find two groups. The greatest similarities were observed for the texts on specific phobia, social phobia and the general concept of phobia, and the lowest for those on generalized anxiety ($p < 0.05$). In conclusion, the results show that the uncorrected predication method is the closest to the ideal gradient. In other words it is the most discriminative method, since the list produced is more similar to the texts that define the structure "Storm phobia" and less similar to those which do not. In addition, the lists using this same method also resemble the texts on the general concept of phobia. The corrected predication algorithm

method (marked as corrected PRE) together with the Centroid methods (marked uncorrected or corrected) produce documents that do not differentiate between texts on phobia, social phobia or specific phobia. They only discriminate for generalized anxiety texts, and therefore proved less sensitive. In general, this result shows that using certain predication algorithm methods satisfactory meanings emerge.

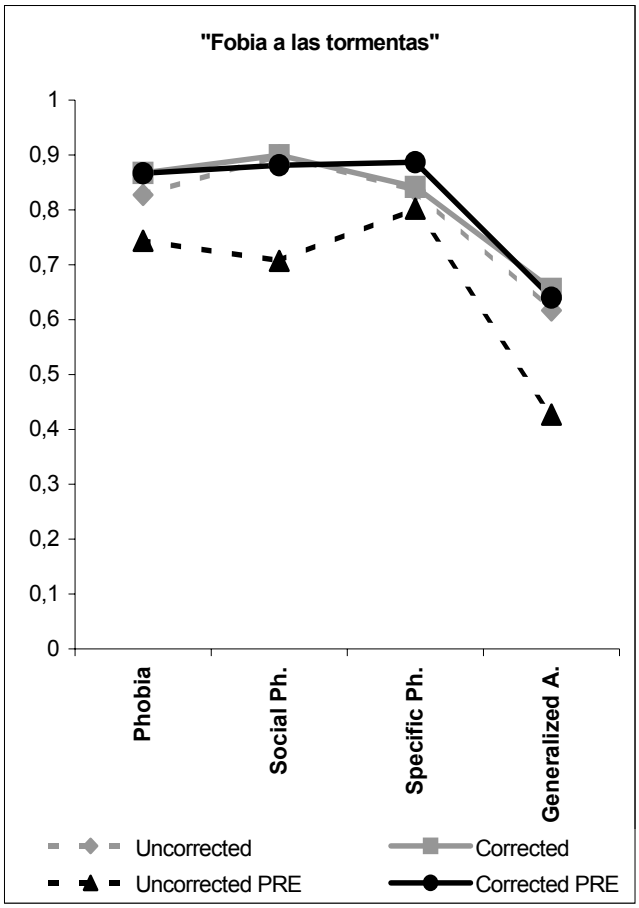


Figure 10. Gradients of meaning for “storm phobia”

6.4.2 “Personalidad de la pistola” (Gun personality)

Secondly, we applied another ANOVA, this time to the structure “gun personality” with the two factors described above (1) Method of extraction of neighbors (with 4 levels): Centroid, Centroid corrected using vector length,

Predication and Predication corrected using vector length. (2) In the second factor we have the four kinds of real texts: general concept of personality disorders, schizoid personality disorder, avoidant personality disorder and antisocial personality disorder. A main effect was found for the type of text ($F(3,84) = 645.45$, $p < 0.05$) and for the method of extraction ($F(1,28) = 1044.92$, $p < 0.05$). We also found an interaction effect ($F(9,84) = 10.78$, $p < 0.05$), represented in figure 11. The lists for “gun personality” reveal some interesting features. The results show the usefulness of applying the predication algorithm to structures of this type. Whilst the lists generated by the Centroid conditions show no significant differences in terms of similarities with each of the personality type paragraphs, both the corrected and uncorrected conditions of Predication algorithm (marked uncorrected PRE and corrected PRE) show significantly greater similarities with the paragraphs on antisocial personality. In the uncorrected predication algorithm condition (marked as uncorrected PRE) the similarities with the texts on antisocial personality were significantly greater than with the other three kinds of texts ($p < 0.05$). In the corrected predication algorithm condition (marked as corrected PRE) we also find greater similarities with the antisocial personality texts, and greater similarities too with the texts dealing with the general concept of personality disorders, although this last difference was only marginally significant ($p = 0.08$). Using this last condition we can also observe that on some occasions the predication method may become more effective when a correction based on the vector length is applied. Applying this correction we conserve the discriminatory capacity in favor of the antisocial personality disorders texts, but also increase the similarity with all of the texts in general. This condition is therefore not only as discriminative as the other predication condition, but rather is the one that covers most content related with personality disorders. In this way, the effect of the low vector length for the argument “pistola” can be mitigated using this correction method, thus obtaining more precise definitions. In conclusion, the results show that both predication conditions match the ideal gradient, but the corrected predication condition performs best. On the one hand, its representation is most similar to the antisocial personality disorder texts, followed by the texts that discuss the general concept of personality disorders. At the same time, it best covers the definition of “gun personality” within the range of personality disorders.

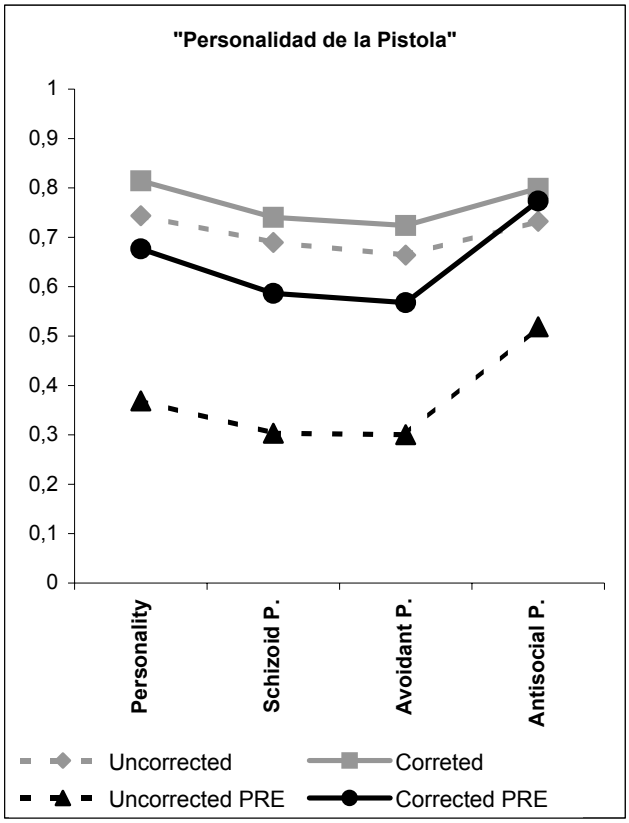


Figure 11: Gradients of meaning for “gun personality”

7. General conclusion

In this article we have sought to explore certain ways in which a system based on LSA and trained using diagnostic corpora may generate definitions. These definitions take the form of lists of semantic neighbors and have been extracted from examples of structures that might fit well with certain simulations previously performed by Kintsch (2001), such as complex structures like “storm phobia” or the terms “phobia” and “storms” separately. To extract the lists of terms separately we used the normal cosine and a correction of the cosine based on vector length. For the complex structures, the centroid and the predication algorithm were used combined with both of the above. These lists intuitively show how the meanings extracted vary in terms of the extent to which the constituent neighbors are restricted to a perfect positive association with the structure they are extracted from (appearance of the low-level definition effect).

Also in the case of complex structures, we see the extent to which content promoted by the arguments or by the predominant meanings take precedent (appearance of the predominant meaning inundation effect). The results show how certain definitions best fit the reality of each structure.

To check these claims in a more objective manner, we selected samples of actual definitions of some disorders related with the target structures, and compared them with each of the lists of neighbors obtained, taking the cosine. This procedure gave us gradients of content that match the actual definitions more or less closely.

In summary, the results show how the predication algorithm can be highly useful for structures of a diagnostic nature where specific characteristics such as “storm” are predicated to a general category such as “phobia”. The meaning can even be extracted when the argument does not coincide with the name of a sub-category, but rather is a simple, very well-defined term such as “gun” (in a hypothetical definition of “gun personality”). Besides this, we have observed that performing certain corrections based on vector length may lead to these definitions covering a wider range of content regarding the intended disorders, although it may occasionally cause an effect similar to what we have termed predominant meaning inundation. The exact conditions under which the latter collateral effect appears might be an area for future investigation. Nonetheless, a combination of the two forms of extracting neighbors may help to extract definitions that cover a greater spectrum of the definition.

A theoretical conclusion that could be drawn from the above is that phenomena found in ordinary language can also be simulated in scientific corpora, such as the predication of properties on some category. Another conclusion that may be made is that LSA models must be treated cautiously as a way of simulating semantic representation, and new algorithms such as that of predication must be found which mean the static matrix representing the semantics of the terms are used efficiently to simulate linguistic and cognitive processes. Broadening the horizons of LSA models means treating them as

more than just a theory of knowledge storage. They should more usefully be considered as a basis for representing information processing.

The practical conclusions revolve around the form of extracting definitions of terms from scientific domains. Although there are parallels between ontologies and models of scientific knowledge extracted from LSA (Burek, Vargas-Vera & Moreale, 2004; Cederberg & Widdows, 2003; Rung-Ching, Ya-Ching & Ren-Hao, 2006), only the former has the capacity to extract the meanings of terms based on previously specified relationships such as synonymy, partonymy, hyponymy, hypernymy and meronymy. However, models of scientific knowledge based on LSA have certain critical advantages: 1) the metric is clearly specified and 2) they are based on actual occurrences in language, which makes them plausible in their mimicry of human cognitive functioning (Dumais, 2003). Thus, the static knowledge represented in LSA may be used to create algorithms based on human bias. With the aid of parsers, it can also detect certain structures that allow the creation of technology to aid classification and management of large quantities of information, such as the indexing of information provided by medical diagnoses (Pakhomov et al., 2006) or in assistance with searches of medical texts (Lee et al., 2006). One such form of assistance is the creation or search for questions drawn from a query of structures similar to those used in this article - for example “eating disorder” or “diabetic retinopathy”. Extracting precise definitions in the form of terms, we would be able to search, or form menus or questions that facilitate searches, and even present alternatives in the form of graphical networks (Jorge-Botana et al., in press) or VIRIs (visual information retrieval interfaces). We believe that a large proportion of future research - both basic and applied - will work in this direction.

Capítulo 9

Visualizing polysemy using LSA and the predication algorithm

(Artículo publicado en ***Journal of the American Society for Information Science and Technology***
Vol 61, Issue 8, pp. 1706–1724.
)
)

Journal of the American Society for Information Science and Technology
Vol 61, Issue 8, pp. 1706–1724.

Visualizing polysemy using LSA and the predication algorithm

Guillermo Jorge-Botana, José A. León, Ricardo Olmos
Universidad Autónoma de Madrid
Yusef Hassan-Montero
SCImago Research Group (CSIC)

Abstract

Context is a determining factor in language, and plays a decisive role in polysemic words. Several psycholinguistically-motivated algorithms have been proposed to emulate human management of context, under the assumption that the value of a word is evanescent and takes on meaning only in interaction with other structures. The predication algorithm (Kintsch, 2001), for example, uses a vector representation of the words produced by LSA (Latent Semantic Analysis) to dynamically simulate the comprehension of predications and even of predicative metaphors. The objective of this study is to predict some unwanted effects that could be present in vector-space models when extracting different meanings of a polysemic word (Predominant meaning inundation, Lack of precision and Low-level definition), and propose ideas based on the predication algorithm for avoiding them. Our first step was to visualize such unwanted phenomena and also the effect of solutions. We use different methods to extract the meanings for a polysemic word (without context, Vector Sum and Predication Algorithm). Our second step was to conduct an ANOVA to compare such methods and measure the impact of potential solutions. Results support the idea that a human-based computational algorithm like the Predication algorithm can take into account features that ensure more accurate representations of the structures we seek to extract. Theoretical assumptions and their repercussions are discussed.

1. Introduction

For decades now, there have been a significant number of psychological models proposing and explaining the processes by which humans understand the meanings of a word, sentence or text from different perspectives. During the seventies, some authors of memory studies took an interest in how a mental representation is stored in LTM (Long-Term Memory[20]) forming semantic networks (e.g., Collins & Loftus, 1975; Collins & Quillian, 1969). Also, discourse has given rise to a number of theoretical models of text comprehension over the last two decades (Glenberg, 1997; Goldman, Varma & Cote, 1996; Graesser, Singer & Trabasso, 1994; Kintsch, 1998; Zwaan & Radvansky, 1998). Each of these models assigns a different weight to the role of context in driving comprehension and explaining how word or sentence disambiguation is formalized.

Whilst some of these models have been implemented in some form, such implementations were not powerful enough to cover a wide range of real situations. However, the recent application of several statistical techniques, powerful programming methods (such as Object-Oriented Programming), new standardized ways of representing entities (such as XML) and new ways of instantiating mathematical objects (such as sparse matrices) have improved our ability to capture large bodies of information in a form that attempts to mimic such mental representations.

This is the case of Latent Semantic Analysis (LSA), a linear algebra and corpus-based technique that has been proposed by some authors as a very effective tool for simulating human language acquisition and representation (Landauer & Dumais, 1997). LSA analyzes a corpus and constructs a dimensional matrix (usually sparse) where each row represents a unique digitalized word (term) and each column represents one document, paragraph or sentence. After some linguistic calculations on this matrix (local and global

[20] Long-Term Memory is a static representation of permanent knowledge. It differs structurally and functionally from Working Memory, which stores items for only a short time and involves temporary activation of meaning.

weighting of each term), the original matrix is reduced via Singular Value Decomposition (SVD). In the resulting matrices, known as a Semantic Space, a word or a combination of words is represented by a vector. To establish the semantic relationship between two words or documents, LSA uses the cosine of the angle between them. A cosine close to 1 reveals a strong semantic relationship, whereas a cosine close to 0 (or even negative) reveals no semantic relationship between the two words. The same principle can be applied to identify the semantic relationship between two documents, or between a document and a term. In addition, the LSA model uses vector length (calculated with Euclidian norm) of the term, which shows how well represented the word is in the semantic vector space.

But LSA, and other models that capture co-occurrences such as the Topic model (Griffiths & Steyvers, 2004; Steyvers & Griffiths, 2007), does not itself distinguish between the different meanings of terms. It is merely a way of representing all the meanings a word can have, statically (Burgess, 2000), with biases and free from context. This, however, is not a disadvantage. The representation should be static since it emulates permanent knowledge, as in LTM. It should be biased since the different meanings represented in the semantic space are weighted according to their occurrences in the real world; this kind of asymmetry between the meanings of a word has been revealed using priming experiments in humans (Williams,1992). It should be context-free since such representations do not need separate entries for each meaning, as for example with lexicography (Kilgarriff, 1997). With this lexical base provided by LSA, we need some mechanisms to be implemented in order to retrieve relevant meanings for a particular context. This has been shown to mimic some parts of human processing, and has produced good results in machines (Kintsch, 2000; Quesada, Kintsch & Gómez-Milán, 2001; Kintsch & Bowles, 2002; Denhière, Lemaire, Bellissens & Jhean-Larose, 2007; Kintsch, 2008)

In this paper we focus our attention on simulating and visualizing how humans understand ambiguous words, using LSA as a static base of word representations, and the predication algorithm (Kintsch, 2001) as a mechanism to filter meanings (although we use other methods to set a baseline). All

processes result in a network diagram that helps us better understand language comprehension, and offer a plausible representation of polysemy in the mind which is also helpful to NPL (Natural Language processing) application designers.

2. Polysemy in context

Language is a complex system that implies deep relations between its components (visual and speech features, letters, words, sentences, paragraphs, texts and concepts). Nonetheless it strikes speaker or reader as an easy, automatic process, perhaps because humans can exploit and manage the context of what they perceive. The properties of polysemy highlight the context effects that operate when we extract word meaning, since meaning depends solely on contingent information.

But in contrast with humans, generally speaking computers and most AI programs do not yet work with semantic context efficiently, because language is difficult material. For example, lexical matching or Boolean searches are hampered by the following limitation (Dumais, 2003): a word can have more than one meaning (e.g. the word “bug”) - this issue can be referred to as the polysemy problem - and a meaning can be expressed in more that one way (e.g. “tumor” and “neoplasm”) - this can be referred to as synonymy. In the context of information retrieval from documents in huge databases, polysemy hinders *precision*, which is the proportion of relevant documents retrieved compared to all those retrieved (many items are irrelevant because they belong to other meanings of the query). Knowledge about semantics and cognition can be used to address this problem. The challenge is to establish mechanisms that filter the information like humans do, exploiting the contexts in which the word, the paragraphs or the texts appear.

Some authors refer to this challenge as “the evolution from key-word to concept” (Kiefer, 2005), based on abandoning the idea of the word as our unit of retrieval, and focusing instead on concept retrieval. This involves some techniques that can be implemented using human-like reasoning. On the one

hand, there are those techniques that are based on categorization, such as controlled vocabularies and ontologies. Examples might be the word-net project, a human-based lexical database, or standards for constructing domain-specific ontologies and knowledge-based applications with OWL or XML-like editors such as PROTÉGÉ. And on the other hand there are statistical techniques using patterns of word co-occurrence. Here we will focus our work on some techniques that use statistical methods, such as LSA, and manage its representation in order to determine which meaning of a word is intended using context. This is often referred as disambiguation.

3. LSA as a basis for semantic processing.

LSA was first described as an information retrieval method (Deerwester et al., 1990) but Landauer et al (1997) suggested that LSA could be a step towards resolving the kind of human advantages that concern the capturing of deep relationships between words. For instance, the problem called “poverty of the stimulus” or “Plato’s problem” asks how people have more knowledge that they could reasonably extract from the information they are exposed to. The solution is that a functional architecture such as LSA allows us to make inductions from the environment – i.e. the reduced vectorial space representation of LSA (explained in the introduction) allows us to infer that some words are connected with one another even if they have not been found together in any sentence, paragraph, conversation, etc. Furthermore, Landauer & Dumais use a simulation to show that for the acquisition of knowledge about a word, the texts in which that word does not appear are also important. In other words, we will acquire knowledge about lions by reading texts about tigers and even about cars; the higher the frequency of a word, the more benefit obtained from texts where it does not appear. These observations are in line with studies that measures the capacity of n^{th} order relations (second order and above) to induce knowledge (Kontostathis & Pottenger, 2006; Lemaire & Denhière, 2006). 1^{st} order relations indicate a type of relationship where the two words occur in the same document of a corpus. With 2^{nd} order relations the two words do not occur together in a single document, but both occur together with a common

word. Higher relations indicate that words don't occur together or linked by a common term. The links lie at deeper levels. These studies conclude that while first order relations can overestimate the eventual similarity between terms, high-order co-occurrences - especially second order relations - play a significant role.

Given such a "human-like" representation, it is not surprising perhaps that spaces formed with LSA have proven adept at simulating human synonymy tasks - managing even to capture the features of humans errors (Landauer & Dumais, 1997; Turney, 2001), and also at simulating human graders' assessments of a summary (Foltz, 1996; Landauer, 1998; Landauer & Dumais, 1997, Landauer, Foltz & Laham, 1998; León, Olmos, Escudero, Cañas & Salmerón, 2006).

The way in which LSA represents terms and texts is functionally quite similar to others stochastic word representation methods. Each term is represented using a single vector that contains all information pertaining to the contexts where it appears. Each vector can be thought of as a box containing meanings. The most salient meaning is the most frequent in the reference corpus (the corpus used to train LSA) followed by other less frequent meanings. Supporting the idea that LSA vectors make massive use of context information, some authors forced LSA to process material that had been tagged, marking explicit differences in usage of the term in each context. The word "plane", for example, was given as many tags as meanings found (plane_noun, plane_verb). Such manipulation results in worse performance (Serafín & DiEugenio, 2003; Wiemer-Hastings, 2000; Wiemer-Hastings & Zipitria, 2001).

Nevertheless, Deerwester et al. (1990) draw attention to some limitations in representing the phenomena of homonymy and polysemy. As explained above, a term is represented in a single vector. This vector has certain coordinates. Since it has several meanings, these are represented as an average of its meanings, weighted according to the frequency of the contexts where it is found. If none of its actual meanings is close to that mean, this could

create a definition problem. This recalls the criticisms leveled at older prototype-based models, which proposed the existence of a prototypical form - the sum of the typical features of the members of that category. The criticism argues that if the prototype is a cluster of features of a category, and bearing in mind the variability of the typical elements, paradoxically the resulting prototype used to establish similarity is in fact a very atypical member, perhaps even aberrant (Rosch & Mervis, 1975).

Theoretically the observations of Deerwester et al. (1990) are true, but less so if we enhance LSA's static representation by introducing some kind of mechanism that activates meaning according to the active context at the moment of retrieval. As Burgess (2000) claims in response to a criticism by Glenberg & Robertson (2000), LSA is not a process theory, but rather a static representation of one kind of knowledge (the knowledge drawn from the training corpus). To simulate the retrieval process we have to manage the net of information supplied by LSA and exploit the representations of terms as well as those of context. Using this mechanism, LSA has been attempted to simulate working memory (Kintsch, 1998), paragraph reading (Denhière et al., 2007), processing whole sentences (Kintsch, 2008) and even approaches to reasoning (Quesada et al., 2001) and understanding predicative metaphors (Kintsch, 2000; Kintsch & Bowles, 2002).

All this simulations have to do with word disambiguation within a language comprehension framework, with the assumption that the value of a word is evanescent and takes on meaning only in interaction with other structures (Kintsch, 1998). There is no way to disambiguate the word ***planta***[21] (plant) if we have no further data. Perhaps some general meanings are automatically triggered in people's minds, such as the plant kingdom, but we can draw no definite conclusion in this respect. If someone asks us "*¿Qué planta*?" in an elevator, an evanescent representation is generated which responds to a task demand, in a context that had previously activated both

[21] *Planta* in Spanish has several meanings in common with the word "plant" in English – for example it can be used to refer to the plant kingdom of living things, and to industrial or power installations. However, other senses are not shared: in Spanish *planta* also refers to the floors of an apartment block, for example.

linguistic and non-linguistic content. According to Kintsch (2008), an ideal implementation of a mechanism that generates such an evanescent representation requires only two components: flexible representation of words, and a mechanism that can choose the correct form in a given context, taking into account representation biases. LSA can provide a good basis for representing words and texts in terms of discrete values. It is a statistical data-driven technique offering flexible vector representation, thus claiming some advantages over ontologies, for instance (Dumais, 2003). Its clear metric (operations with vectors) allows us to implement efficient algorithms such as the predication algorithm (Kintsch, 2001).

In summary, storage and retrieval are not independent processes. The form in which a term is stored, even if this seems very atypical or even aberrant, is not critical. What is important is that, as in real life, linguistic structures take on the correct form, in line with the context, at the time of retrieval. The goal is to manage the flexible representation well. As Kintsch (1998) stated, knowledge is relatively permanent (the representation that supplies LSA) but the meaning - the portion of the network that is activated - is flexible, changeable and temporary.

4. The Predication Algorithm operating on LSA.

As we saw above, resolving polysemy involves retrieving the right meaning for a word in its context, and ignoring irrelevant meaning. In order to do this, we need to implement a mechanism that activates only that meaning of the word pertinent to the retrieval context.

Suppose this structure (the word "***planta"*** in the context of the word ***rosal)***:

"El rosal es una ***planta***" (The rosebush is a plant)

One way to show the meaning of such a structure is by listing the semantic neighbors closest to that structure (the vector that represents it). To

extract these semantic neighbors we need a procedure that calculates the cosine between this vector and all vector-terms contained in the semantic space, and keeps a record of the *n* greatest values in a list - the *n* most similar terms to the selected structure. Using this procedure we should obtain neighbors related to the plant kingdom.

But the word "***planta***" has more meanings:

> "La electricidad proviene de la ***planta***" (The electricity comes from the plant)
> "El ascensor viene de la ***planta*** 3" (The elevator came from the 3rd floor)

All these structures have the term "***planta***" as a common denominator, while this same word takes on different meanings. We all know that in "El rosal es una ***planta***" the word "***planta***" does not have the same meaning as in "la electricidad proviene de la ***planta***". The same term acquires one or other set of properties according to the contexts that accompany it. In other words, the properties that give meaning to this term are dependent on the context formed by the other words.

Let us take the proposition TERM [CONTEXT], assuming that the TERM takes on some set of values depending on the CONTEXT. Both TERM and CONTEXT would be represented by their own vectors in an LSA space.

To calculate the vector that represents the whole proposition, the common form of LSA and other vector space techniques would simply calculate a new vector as the sum or the "centroid" of the TERM vector and the CONTEXT vector.

Thus if the representation of the vectors according to their coordinates in the LSA space were:

TERM vector= {t1,t2,t3,t4,t5,…,tn}
CONTEXT vector= {c1,c2,c3,c4,c5,…,cn}

Then the representation of the whole proposition would be:

PROPOSITION vector = {t1+c1, t2+c2, t3+c3, t4+c4, t5+c5,…, tn+cn}

Due to the limitations of representation of meanings in LSA (Deerwester et al, 1990) explained earlier, this is not the best way to represent propositions, as it does not take into account the term's dependence on the context. In other words, to compute the vector of the entire proposition we do not need all the properties of the TERM ("*planta*"), only those that relate to the meaning of the subject area (plant kingdom). What the centroid or Vector sum does using the LSA method is to take all the properties - without discriminating according to CONTEXT - and add them to those of the TERM.

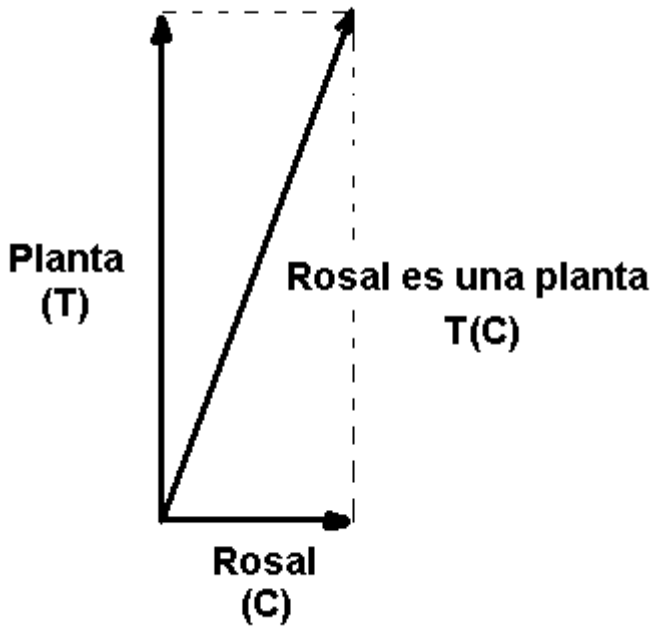


Figure 1. Bias of the Centroid method between the words "Planta" and "Rosal". Due to the vector length of the terms, the vector of Planta(Rosal) is close to the vector of Planta.

All the properties of the term "*planta*" are taken into account when calculating the new vector. If, as in the above example (figure 1), the CONTEXT has a much lower vector length than the TERM, and other meanings are better represented in the TERM, the vector that represents the predication will not capture the actual intended meaning. The meaning will be closer to the meaning of the context most represented in the LSA space. With this simple vector sum method, the length of the term-vectors involved dictates which semantic properties the vector representing the predication will take on. We can therefore assume that the Vector Sum method fails to account for the true meaning of certain structures, and tends to extract definitions of a given structure that are subordinate to the predominant meaning.

The Predication Algorithm (Kintsch, 2001), as its name suggests, was first used on Predicate (Argument) propositions such as "The bridge collapsed", "The plan collapsed", "The runner collapsed". Nonetheless it may be used for general Term(Context) structures. It aims to avoid this unwanted effect by following some simple principles based on previous models of discourse comprehension such as Construction-Integration nets (Kintsch, 1998). These principles are founded on node activation rules that show how the final vector representing Term(Context) must be formed with the vectors of the most highly activated words in the net. This activation originates from the two terms (*Term* and *Context-term*), spreading to all words in the semantic space (see figure 2). It is assumed that the more activated nodes, the more words are pertinent to both Term (T) and Context (C) where T is constrained by C.

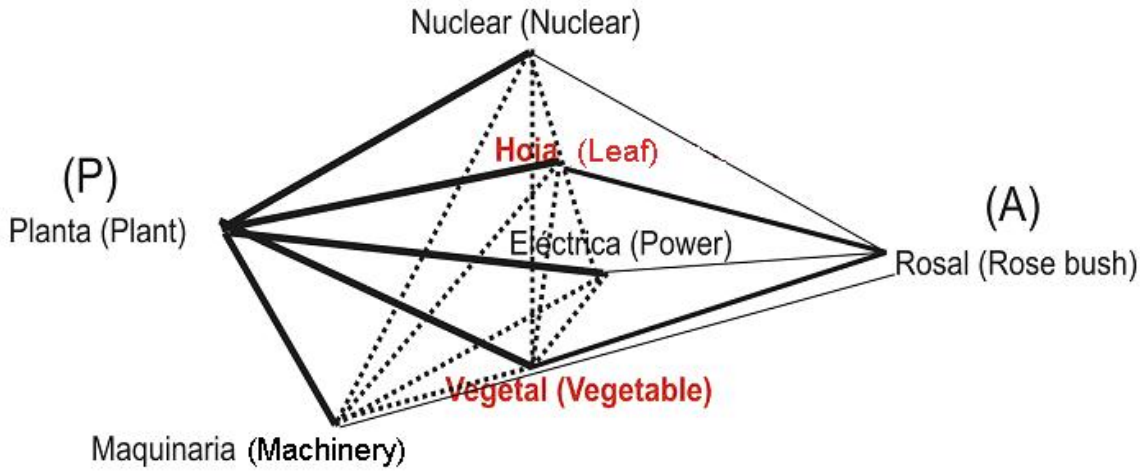


Figure 2. Predication net between the words "Planta" (Plant) and "Rosal" (Rosebush). Only neighbors of the Predicate relevant to the Argument will receive high activation: "vegetal" (vegetable) and "hoja" (leaf).

To apply this algorithm it is necessary to have an LSA semantic space or some other vector space model as a starting point (see Kintsch, 2001 for an in-depth presentation of the procedure). The steps are as follows: 1) find the *n* first terms with greatest similarity to the Term (T), *n* being an empirical parameter. 2) Construct a net with excitatory connections between each of those *n* terms and the Term (T), and between each of those *n* terms and the Context (C). In each case the cosine is used as the connection weighting value, i.e. a measure of similarity between the words. 3) Some inhibitory connection between terms in the same layer can be implemented (terms in the *n* first terms layer compete with one another for activation while they are activated by Term (T) and Context (C)) 4) Run the net until all nodes are recalculated using a function that uses excitatory and inhibitory connections and promotes bilateral excitation of each node (the net does not need a practice trial because the definitive weights are those imposed by the LSA matrix). 5) The final vector for the predication is calculated using the sum of the Term-vector (T), Context-Vector (C) and the *p* most activated vector-nodes (again, *p* is an empirical value and $p < n$). The idea is that these *p* terms are semantically related to both predicate and argument. The Context filters out the *n* terms most closely related to the Term,

and of these retain only the *p* terms most pertinent to both predicate and argument.

Once the vector T(C) is established it can be compared with the terms in the semantic space (using cosine or another similarity measure) to extract a list of semantic neighbors - the meaning that is most closely related to the structure, and which would contain a definition of it. This is the way that we use the predication algorithm to resolve polysemy issues, where meaning is constrained by a particular context.

5. Objectives

The main aim of this article is to analyze the disambiguation of a polysemic word in a retrieval context and reveal the biases that affect it.

Whilst several studies have used larger contextual units such as sentences or paragraphs (Lemaire, Denhière, Bellisens & Jhean-Larose, 2006; Olmos, León, Jorge-Botana & Escudero, 2009), we have used a single context word to modulate the meaning of a polysemic word. Such a small window of context has been referred to in other studies as a micro-context (Ide & Veronis, 1998). The word "plant" takes on one meaning or another depending on the context word ("rosebush" or "energy"). Our feeling is that combinations such as Word/Context-word[22] are a more parsimonious way to visualize the concepts, whilst preserving sensitivity to minor changes in context word, and this is therefore the framework we use to investigate the behavior of vectorial space models such as LSA.

We consider three problems concerning the system's approach to processing polysemic words (see Jorge-Botana, Olmos, León, 2009, for a previous study with a specific domain corpus and different meanings of diagnostic terms). Problems emerge during the extraction of a polysemic

[22] Following the notation adopted in section 4, we refer to Word/Context-Word structures as [T(C)].

word's meaning (as defined by its neighbors), both with and without an explicit context.

I) Potential problems extracting the meaning of polysemic words with no explicit context.

(I.a) Predominant meaning inundation: It is possible that only the predominant meaning of a word arises, if other meaning are not sufficiently well-represented in the semantic space.

(I.b) Low-level definition: It is possible that not only predominant meanings are generated, but terms are restricted to local relationships with the polysemic word. In a previous study using a scientific corpus, some neighbors extracted were excessively ascribed to local relationships (Jorge-Botana et al., 2009). For example, when extracting the list of neighbors for "*fobia*" (phobia), we obtained a list with many words such as "shy", "humiliating", "girls", "embarrassing" and "shops", whereas an ideal list would also contain neighbors which best represent the general topics of psychopathology and designate higher categories such as "fear", "sub-type", "exposure" or "anxiety". The lack of this kind of terms was probably due to the simple cosine frequently promoting highly local relationships in the comparisons. It seemed that most of the neighbors rarely appear without the polysemic word.

II) Potential problems extracting the meaning of polysemic words with explicit context.

(II.a) Predominant meaning inundation: As in case I.a above, if the context-word is represented weakly in the semantic space, it is possible that only the dominant meaning of the polysemic word is generated.

(II.b) Imprecise definition: Even when the retrieval context is the dominant meaning of the polysemic word (in the semantic space), the

meaning extracted may be very general if the vector length of the word is greater than that of the context-word. The result is that the meaning extracted from the polysemic word is related to the context word but is not sufficiently precise.

(II.c) Low-level definition: As in case I.b, the meaning facilitated by the context-word may be generated, but the terms that represent this meaning are related in a very local way with the polysemic word. These terms usually co-occur in documents containing the polysemic word but never without it.

Problem I.a cannot be solved by computational systems (nor by humans), since there is no explicit context to guide the correct meaning of a polysemic word. The meaning extracted depends on the representation of each context in the semantic space. When context is present, problems II.a and II.b, we propose, can be resolved by applying Kintsch's algorithm, as explained in section 4. In the case of problems I.b and II.c, we propose to extract the neighbors that represent each meaning, adjusting the simple cosine with the vector length of each term in the semantic space. This should produce a more "high-level" neighbor list, with words from local relationships as well as words that are less constrained by this kind of relationship. The function simply weights the cosine measure according to the vector length of the terms (see section 7 below), thus ensuring a semantic network with some well-represented terms. This method showed good results in a previous study using a domain-specific corpus (Jorge-Botana et al., 2009), and our aim now is to apply it to a general domain corpus like LEXESP.

We will use the following protocol:

In the first step, visualization, we visualize two meanings of two words in a semantic network as an example. We will extract semantic neighbors using the two methods outlined in section 4: Vector Sum and the Predication Algorithm. A base line condition is also used extracting the neighbors for each word without context. This procedure will allow us to visualize the probable main

biases in the disambiguation of a word using LSA or another vector-space method, (explained as problems II.a and II.b).

In the second step, to test the efficiency of the predication algorithm, we examine a sample of polysemous items conjoined to their contexts. To compare its efficiency, we conduct an ANOVA comparing the three conditions from the visualization step, and numerically demonstrate the biases of these steps.

6. Visualizing the networks

6.1 Procedure

Following on from the work of authors who extracted and ordered all meanings of some polysemic terms – for instance "apple" as a software company or a fruit (Widdows & Dorow, 2002) – we represent polysemic meanings using a term-by-term matrix (N×N) in which each cell represents the similarity of two terms from the list of the *n* most similar terms (*n* first semantic neighbors) to the selected structure. For example, in the case of "***planta***", the term-by-term matrix would comprise the first *n* semantic neighbors extracted. The resulting matrix was the input to Pathfinder[23].

Our aim was to calculate the vector that represents the structure, its neighbors and the similarity between them, using LSA as our static basis for word representation, combined with some derived methods to solve the problems outlined in section 5. The main procedure is as follows:

First, we drew the net of the polysemic word alone without any context word (e.g. "*plant*"), in order to see the natural predominant meaning. This method involves extracting a list of the term's semantic neighbors[24] and compiling a graph with them (Pathfinder input). This diagram will highlight the problems that

[23] We explain Pathfinder Network Analysis in section 7 (method)
[24] Semantic neighbors of a term (or of a structure) are extracted comparing the vector of the term with each of the terms in the LSA semantic space.

arise from (I.a Predominant meaning inundation). The examples that we use are *planta* and *partido*[25], polysemic words without any context.

Second, we drew the net for the polysemic word T accompanied by two context words (C1 and C2), (e.g. [plant (energy)] joined to [plant (rosebush)]). We calculate the values of each structure, T(C1) and T(C2), with the simple vector sum of their component vectors ($V_{plant} + V_{energy}$) *and* ($V_{plant} + V_{rosebush}$), and again extract the semantically related neighbors for each structure. Finally, we compile a graph with all such neighbors. This will reveal problems II.a (Predominant meaning inundation) and II.b (Imprecise definition). The actual examples we use are *partido (fútbol, nacionalista)* [match/party (football, nationalist)] and *planta (rosal, piso)* [plant/floor (rosebush, apartment)].

Third, we also drew the net for the polysemic word T accompanied by each of two contexts C1 and C2, but this time calculating the values of the two structures, T(C1) and T(C2), using Kintsch's predication algorithm (Kintsch, 2001). Again, we extract the neighbors of each structure (both vectors calculated with Kintsch's predication algorithm) and join them to make a graph. This will show the proposed solution to the problems (II.a Predominant meaning inundation) and (II.b Imprecise). In other words, we sought to verify whether Kintsch's predication algorithm is an effective method to visualize the meanings of the two meanings together (compared to the first and second conditions). The examples we use are again *partido (fútbol, nacionalista)* [match/party (football, nationalist)] and planta (rosal, piso) [plant/floor (rosebush, apartment)].

Additionally, to search for a solution to problems I.b and II.c (Low-level definition), we compose the extracted list of neighbors from the vector representation of each structure in each condition (isolated word, vector sum and predication algorithm) using two methods: the simple cosine measure to find the similarity between vectors, and the cosine corrected using vector length (Jorge-Botana et al., 2009). The assumption behind this latter method is that it

[25] *Partido* in Spanish has several meanings, the commonest being "political party" and "game/match" (only in the sense of playing competitively).

more carefully avoids local relationships. We will explain this method in the following paragraph.

6.2 Method

- Conditions. In this study we propose three different conditions for extracting a list of neighbors:

A) *Isolated Word*. To obtain a reliable benchmark value, we extract neighbors for isolated words (T) (such as "*planta"*). This shows us the actual representation of a word, independent of the retrieval context. We extract two lists of 30 neighbors, one obtained using the simple cosine method, and the other with simple cosine adjusted for vector length (explained below).

B) *Word/Context-word vector sum*. We need to contrast the predication algorithm results with a baseline value, so we extracted neighbors of the vector of each *Word/Context-word,* T(C1) and T(C2), using the classical method to represent complex structures – the simple vector sum. We extracted 30 neighbors for each of the two vectors and using each of the two methods (30 with simple cosine, 30 adjusting the cosine according to vector length) and so obtained four lists with a total of 120 terms. Repeated terms were deleted and reduced to a single representation, and the four lists were then merged.

C) *Word/Context-word with predication algorithm*. In this condition we extract semantic neighbors of each *Word/Context-word,* T(C1) and T(C2), using the predication algorithm (Kintsch, 2001). We extract 30 neighbors of each of two predication vectors with each of the two methods (30 with simple cosine, 30 adjusting the cosine according to vector length), again obtaining four lists with a total of 120 terms. As before, repeated terms were deleted and reduced to a single representation, and the four lists were merged to produce the input to Pathfinder (the square matrix of terms).

- LSA, corpus and pre-processing. LSA was trained[26] with the Spanish corpus LEXESP[27] (Sebastián, Cuetos, Carreiras & Martí, 2000) in a "by hand" lemmatized version (plurals are transformed into their singular form and feminines are transformed into their masculine form; all verbs are standardized into their infinitive form). The LEXESP corpus contains texts of different styles and about different topics (newspaper articles about politics, sports, narratives about specific topics, fragments from novels, etc.). We chose sentences as units of processing: each sentence constituted a document in the analysis. We deleted words that appear in less than seven documents to ensure sufficiently reliable representations of the terms analyzed. The result was a term-document matrix with 18,174 terms in 107,622 documents, to which we applied a weighting function. This function attempts to estimate the importance of a term in predicting the topic of documents in which it appears. The weighting functions transform each raw frequency cell of the term-document matrix, using the product of a local term weight and a global term weight. We applied the logarithm of raw frequency as local weight and the formula of entropy as global term weight. We applied the SVD algorithm to the final matrix, and reduced the three resulting matrices to 270 dimensions. The mean of the cosines between each term in the resultant semantic space is 0.044, and the standard deviation is 0.07.

- Parameters for the predication algorithm. We have set *n (n* first terms with greatest similarity to the Term T) at 5% of the total number of terms in our space, and *k* (number of activated term nodes whose vector is taken to form the final representation of the structure T(C)) equal to 5.

- Methods for extracting the list of neighbors: Simple cosine and simple cosine adjusted for vector length. Once we have a vectorial representation of the structure (Word/context-word or Isolated word), this vector

[26] For calculations we used Gallito ®, an LSA tool implemented in our research group and developed in Microsoft® .Net (languages: VB.net and C#) integrated with Matlab ®. We also use this tool to implement the predication algorithm with net activation calculations (available at www.elsemantico.com).

[27] In a study of Duñabeitia, J.A., Avilés, A., Afonso, O., Scheepers, C. & Carreiras, M.(2009), semantic pair similarities of the vector-words from this space have displayed good correlation with their analogous translations to English using spaces from an LSA model by http://lsa.colorado.edu/ and from HAL (Hyperespace Analogue to Lenguage) hosted at http://hal.ucr.edu/ , and also with judgment of Spanish natives speakers.

must be compared with each term in the semantic space in order to extract the list of neighbors. Two different methods were used:

a) The simple cosine with each vector-term in the semantic space, this being the traditional method.

[Similarity = Cosine (*A, I*)] where *A* is the vector that represents the structure from which we want to extract the neighbors (word/context word, Isolated word) and *I* is each of the terms in the semantic space.

b) The simple cosine adjusted for vector length.

[Similarity = Cosine (*A, I*) × log (1 + Vector length (I))], where *A* is the structure from which we want to extract the neighbors (word/context word, Isolated word) and *I* is each of the terms in the semantic space.

Based on previously successful results (Jorge-Botana et al., 2009), we decided to use both methods in order to obtain more authentic results, and both lists were used to construct the graphs.

		Method for extracting neighbors	
	Structure and concrete examples used	Cosine	Cosine adjusted w/ vector length
Isolated word	W *Planta* (Plant/Floor)	30	30
Word/Context word Vector Sum	W(WC1) *Planta (Rosal)* [Plant/Floor (Rosebush)]	30	30
	W(WC2) *Planta (Piso)* [Plant/Floor (Apartment)]	30	30
Word/Context word with predication algorithm	W(WC1) *Planta (Rosal)* [Plant/Floor (Rosebush)]	30	30
	W(WC2) *Planta (Piso)* [Plant/Floor (Apartment)]	30	30

Table 1. First, two lists of 30 neighbors were obtained for the isolated condition (W) - one with cosines and the other applying the correction to the cosine. Second, four lists are extracted from the Word/Context word structures formed with the Vector Sum condition - two for the first predication W (WC1) and two for the second predication W (WC2), one for each approach to extracting neighbors, cosines and corrected cosine. Thirdly, lists of the Word/Context word structures formed with the predication algorithm were extracted in the same way.

- Similarity Matrix. In order to understand and visually compare the semantic structure of the networks, visualization techniques were applied. As input data, we used a symmetrical N×N matrix that uses cosines to represent the semantic similarity of each term with all others. These terms are all the words merged in the list of neighbors from conditions in the previous steps (*Isolated Word, Word/Context-word vector sum, Word/Context-word with predication algorithm)* . In other words, N is the total number of terms in these three lists. For instance, in the case of *Word/Context-word with predication algorithm* N counts the two lists for the first structure T(C1) plus the two lists for the second structure T(C2) (Repeated terms of the merged list were deleted)

Following the methodology proposed by Börner, Chen and Boyack (2003), we first need to reduce this n-dimensional space as an effective way to summarize the most meaningful information represented in the network matrix. Analyzing scientific literature on visualization reveals that the most useful dimensionality reduction techniques are multidimensional scaling (MDS), Factor Analysis (FA), Self-Organizing Maps (SOM), and Pathfinder Network Analysis (PFNets).

- Dimensionality Reduction. For this study we chose Pathfinder Network Analysis (Schvaneveldt, 1990; Guerrero-Bote et al., 2006; Quirin et al., 2008), a robust method widely used in computer science, information science and cognitive science research, originally conceived to extract what human subjects judged to be the most pertinent concept-concept relations. Networks pruned with this algorithm are known as PFNets. Our aim using the Pathfinder algorithm is to obtain a clearer, more easily comprehensible network by pruning less significant links between terms – those which violate the 'triangle inequality' since they do not represent the shortest path between two terms. In contrast with other dimensionality reduction techniques, Pathfinder preserves the stronger links between terms instead of dissipating them among multiple spatial relationships.

The Pathfinder algorithm makes use of two parameters: r and q. The first determines how to calculate the distance between two terms that are not

directly linked. Possible values for parameter r are 1, 2 and ∞. For r = 1 the path weight is the sum of the weights of links along the path; for r = 2 the path weight is the Euclidean distance between the two terms; and for r = ∞ the path weight is the maximum link weight found on the path. Parameter q indicates the maximum number of links along the path in which the 'triangle inequality' must be satisfied. The q value must be in the range 0 < q < N, where N is the number of terms. Modifying q and r values, we obtain different PFNets, with different topological structures. For this study we used q = N-1 and r = ∞ as pruning values, because our trials have shown that the resultant networks are visually clearer and the links preserved are the most pertinent.

- Spatial layout. Once we have pruned the networks, in order to visually represent them in a 2D space we need to apply a graph layout algorithm. The layout algorithms aim to place all graph nodes (in our case terms) in positions that satisfy aesthetic criteria: nodes should not overlap, links should not cross, edge length and node distances should be uniform, etc. (Börner, Sanyal & Vespignani, 2007). The most widely-used layout algorithms are those known as Force-Directed Layout algorithms, specifically those developed by Kamada & Kawai (1989) and Fruchterman & Reingold (1991).

Whilst Fruchterman and Reingold's algorithm is more suitable for representing fragmented networks (i.e. a network comprising many components or sub graphs with no connections between them), Kamada and Kawai's algorithm has proved more suitable for drawing non-fragmented PFNets (r=∞; q=N-1) (Moya-Anegón, Vargas, Chinchilla, Corera, Gonzalez, Munoz et al., 2007). For our study Kamada and Kawai's algorithm might constitute a good option, but instead we chose a novel combination of techniques to spatially distribute PFNets (r=∞; q=N-1) that demonstrate even better results. We use Fruchterman and Reingold's algorithm, but with a previous radial layout. Radial layout is a well-known low-cost layout technique, where a focus node was positioned in the centre of the visual space, and all other nodes were arranged in concentric rings around it. Once radial layout is applied, then Fruchterman and Reingold's algorithm optimizes the spatial position of the nodes. These algorithms have been implemented in a network viewer application, developed

by the SCImago research group, which is also being used for visual representation of scientific co-citation networks (SCImago; 2007).

- Visual Attributes. In addition to the information represented by the term's spatial position and its connection with other terms, the graph contains other useful information, encoded using two visual attributes: a) the node's ellipse size indicates the term's vector length; b) the width of the link indicates the weight of the semantic relationship.

- Related visualization studies for LSA with Pathfinder. It is difficult to locate previous research that combined LSA with Pathfinder Network Analysis and spatial layout algorithms. One pioneering example is a 1997 study by Chen (Chen & Czerwinski, 1998), in which document-to-document semantic relationships (but not term-to-term networks) are visually represented using PFNets and LSA. Kiekel, Cooke, Foltz & Shope, (2001) applied PFNets and LSA to a log of verbal communication among members of a work team – very different source data to the type used in this study. Recently, Zhu and Chen (2007) have represented the terms of a collection of documents using graphs, using LSA to represent the semantics of the corpus, but without the Pathfinder pruning method.

6.3 Results and discussion

6.3.1 Results and discussion regarding PARTIDO (FÚTBOL, NACIONALISTA) [match/party (football, nationalist)]

- *Isolated word*. If we consider "*partido*" in isolation[28], only the general meanings referring to a "political party" are extracted (see figure 3). Only three or four nodes were assigned terms that corresponds to football topics (zone separated by line a in figure 3); all others contained political terms, an effect we have referred to as Predominant meaning inundation – the predominant meaning of "*partido*" is political.

[28] Note that the word "*partido*" does not need to be located in the center of the net, because owing to the spatial distribution of the visualization algorithm, it will be located around the words with links to it that were not pruned, i.e. around the words with strongest semantic relationships with "*partido*". This visual effect will be valid for all of the nets. In the cases where the term (T) is tested with two contexts (C1) and (C2), although all terms in the net are related with (T), the term is normally located near one of the meanings because not all the senses of the polysemous words are equally probable, and there is usually a dominant sense.

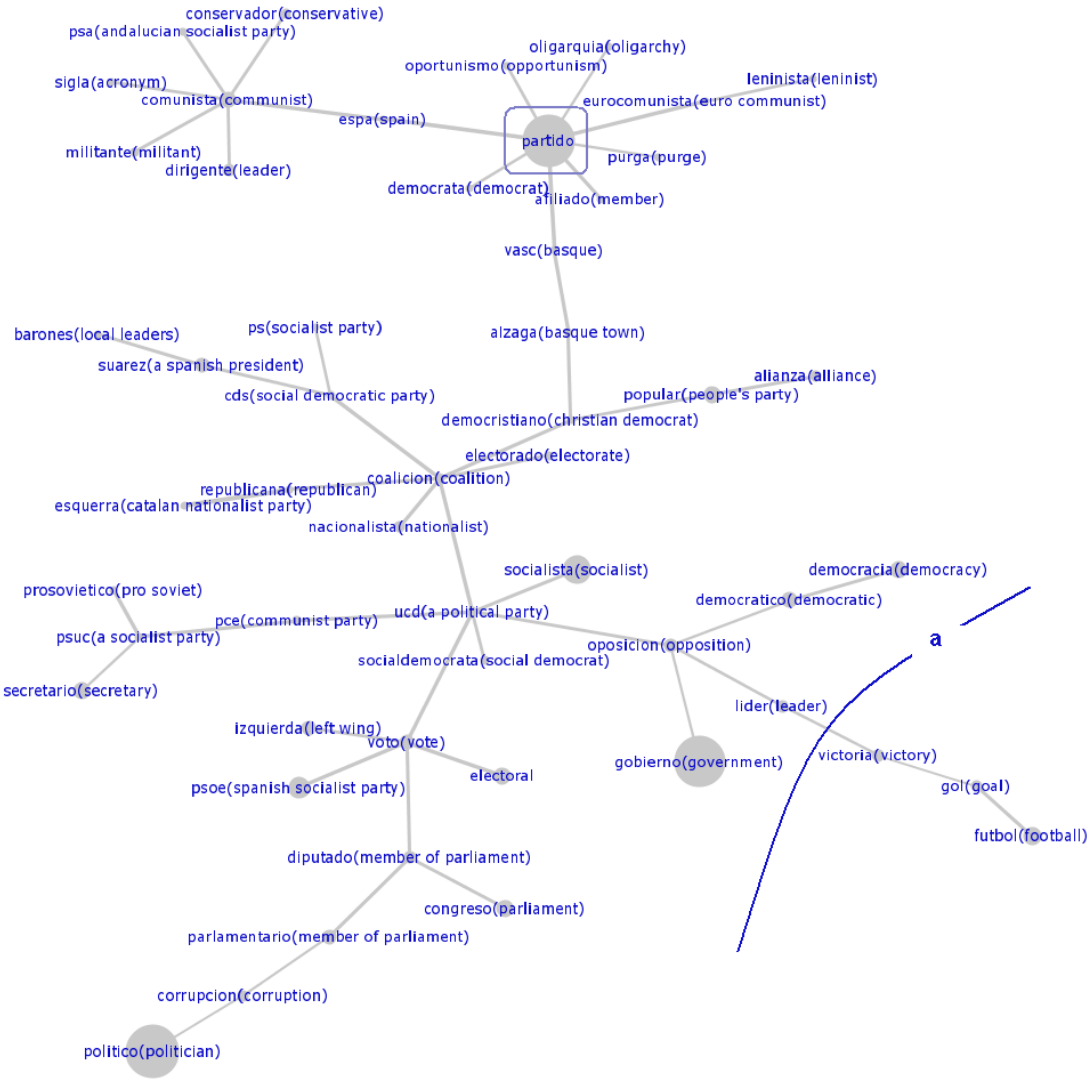


Figure 3. Visual representation of the word "partido" in isolation, without any retrieval context word. Note that the word "partido" don't need to be located in the center of the net because, due to the spatially distribution of the visualization algorithm, it will be located around the words with the most similarity with it, although all terms have enough similarity with it to be a neighbor. This visual effect will be valid for all the nets. In the cases in which the term (T) is tested with two contexts (C1) and (C2), although all terms of the net are related with (T), the term use to be located near to one of the sense because not all the polysemous words are equiprobables and usually have a dominant sense.

- *Word/Context word - vector sum condition*. To test contexts we formed two structures, *Partido (Nacionalista)* [nationalist party] and *partido (fútbol)* [football match]. Using the simple vector sum, the sports-related topics make

gains against political topics (zone separated by line in figure 4). Introducing the football context reduces the predominant meaning inundation effect but fails to eliminate it completely. When we introduce the context word *nacionalista* it imposes a purer definition in the shape of growing number of terms such as "republican", "nationalist" and "Basque", although most political terms are still related to politics in a more general meaning – something we referred to previously as an imprecise definition.

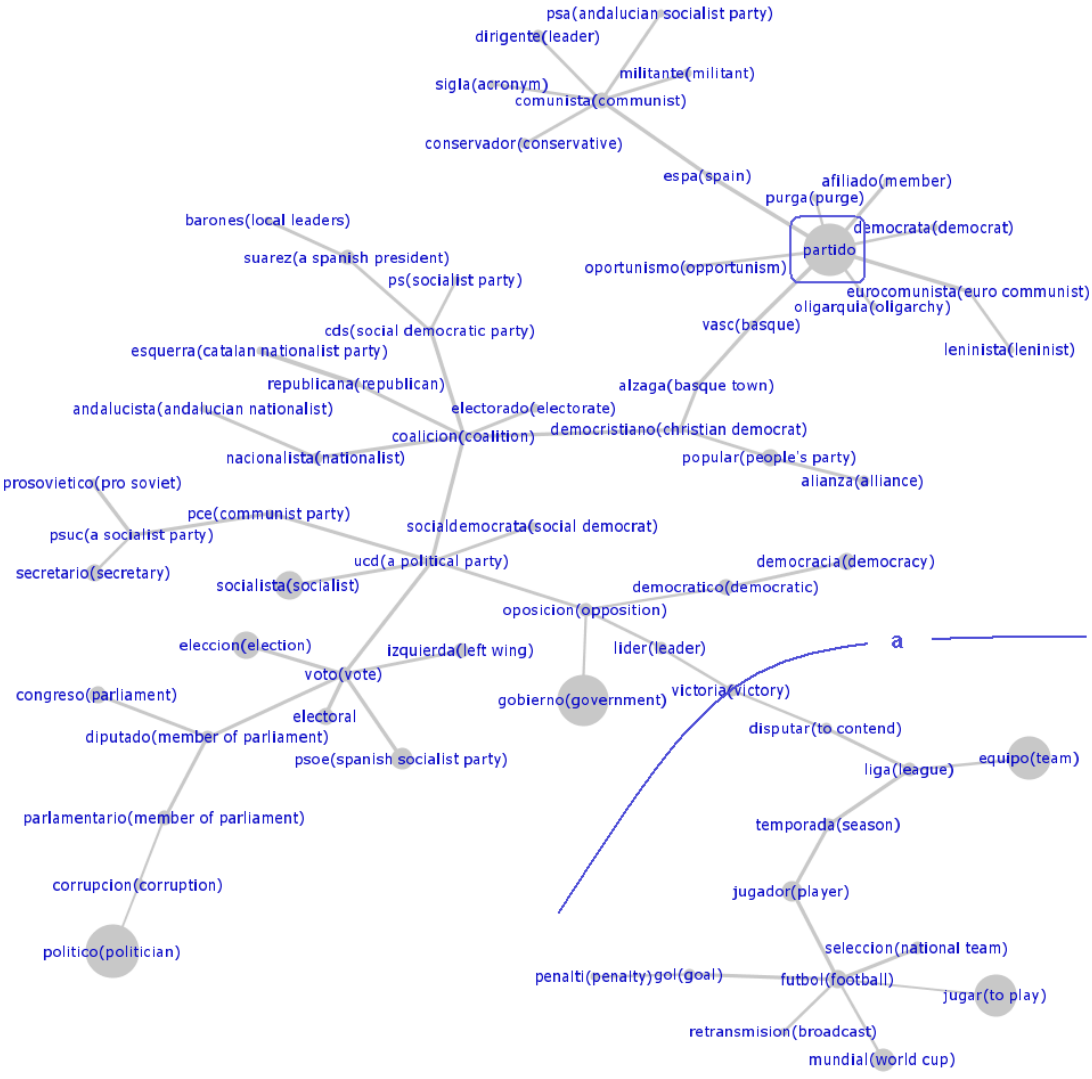


Figure 4. Visual representation of the word "*partido*" presented with two context words: "*nacionalista*" (nationalist) and "*fútbol*" (football) forming two structures: *Partido(nacionalista)* and *Partido(football)*. The predication algorithm is not applied on this occasion.

- *Word/Context word with predication algorithm condition*. As we can see from figure 5, sports-related meaning has enjoyed a surge relative to terms related to the argument "*nacionalista*". There is now no sign of the Predominant meaning inundation effect mentioned earlier (see the two zones separated by the line a), and the vast majority of political terms were concerned with nationalism, such as regions with nationalist parties (e.g., "*Cataluña*", "*Euskadi*" or "*Pais Vasco*", "*Galicia*", "*Andalucía*"), languages spoken there (e.g. "*castellano*", "*catalán*", "*vasco*"), nationalist political parties (e.g., "PNV", "*Esquerra*"), and even names of other political parties that operate in those regions (e.g. "PSOE", "PP", "UCD"). The predication algorithm has managed not only to retrieve the elements of "*partido*" with a political meaning, but also to retrieve some terms referring to a specific political sub-sector. This time the definitions of items more closely matched to the topic, avoiding the imprecise definition effect**.** In comparison with the *vector sum* condition (see figure 4), political meaning is expressed using different terms. An additional anecdotal phenomenon is that link between the two contexts is a football coach "Clemente", a native of the Basque country. This is a more orderly manner in which to link the contexts than the structural links generated in the two other conditions.

Since we wanted to avoid the meaning of our nets being flooded with very extreme local relationships, we had applied the correction based on vector length mentioned above[29], as used in a previous study (Jorge-Botana et al., 2009). Some long term vectors were therefore obtained, e.g. for "*jugar*" (to play), "*juego*"(game), "*lengua*" (language), "*equipo*" (team), "*gobierno*" (goverment), "*vasco*" (Basque), "*socialista*" (socialist), "*español*" (Spanish), "*campo*" (stadium or countryside), "*partido*" (political party or game/match), "*mundial*" (world cup or worldwide), "Barcelona" and "*elección*" (election). These terms with long vectors (represented by a large circle) do not necessarily coincide with the terms with more or stronger links (in bold type). As we can see from the network, just the opposite is true: nodes with more than one

[29] We did so to extract the neighbors for all three nets, but we explain the results in the Word/Context Word with predication algorithm condition.

branch are usually occupied by terms with low vector length. It could be claimed that such concrete terms are key examples of these concepts' definitions. This is probably due to the fact that words with short vectors are present only in those contexts that represent the single topic, and are attached to the topic that promotes the context. This makes these terms unmistakable features of the any definition – for *partido (fútbol)* they are words such as "UEFA", "*disputar*" (to contend), "*futbolístico*" (relating to football), "*liga*" (league), "*seleccionador*" (coach), "*selección*" (national team), "*centrocampista*" (midfielder), "*gol*"(goal); for *partido (nacionalista)* they are words such as "*nacionalista*" (nationalist), "*voto*" (vote), "PSE" (Basque socialist party), "PNV" (Basque nationalist party), "CIU" (Catalan nationalist party), "PSC" (Catalan socialist party). In contrast, terms with longer vectors are often found in a variety of contexts, and are related to a wider range of topics than those facilitated by the context, blurring the relationship with the key concrete terms. These longer vector terms are not unmistakable examples of a topic, but they are often contained in high-level definitions of the topic facilitated by the context. Both kinds of terms are important to provide the concept definition without what we refer to above as the low-level definition effect.

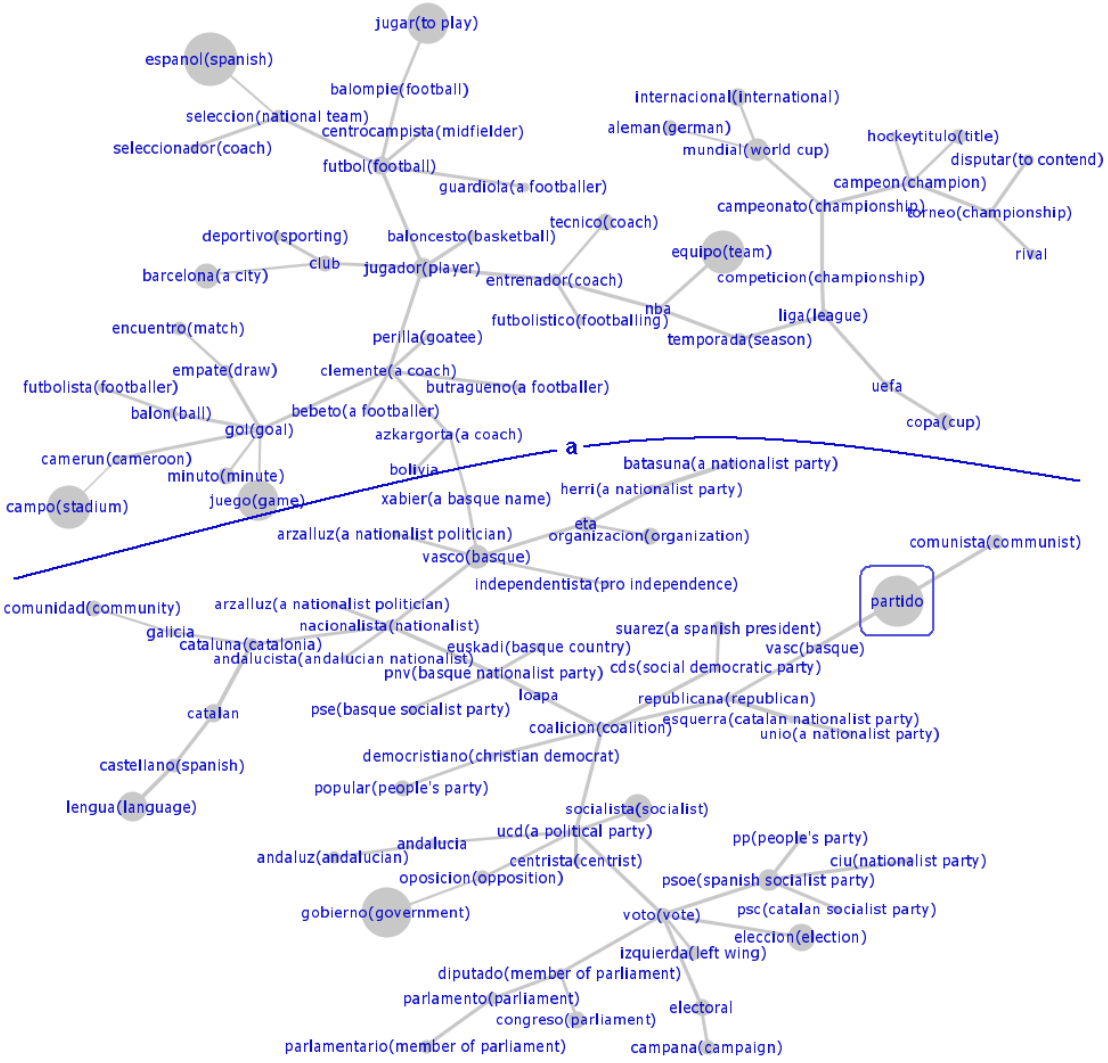


Figure 5. Visual representation of the word “partido” in the domain of two contexts “nacionalista” and “fútbol”. They forming two structures: Partido(nacionalista)” and Partido(fútbol)”. Now, predication algorithm is applied.

6.3.2 Results and discussion regarding to PLANTA (ROSAL, PISO) [plant/floor (rosebush, apartment)]

- *Isolated word*. If we wish to retrieve the meanings of the word “*planta*” in isolation (figure 6), we obtain some of their more important related meanings. Some have to do with power plants (zone separated by line a), some with the plant kingdom (zone separated by line b) and very few terms relate to buildings (zone separated by line c). This time there is no single predominant meaning inundation effect as there is more balance between meanings.

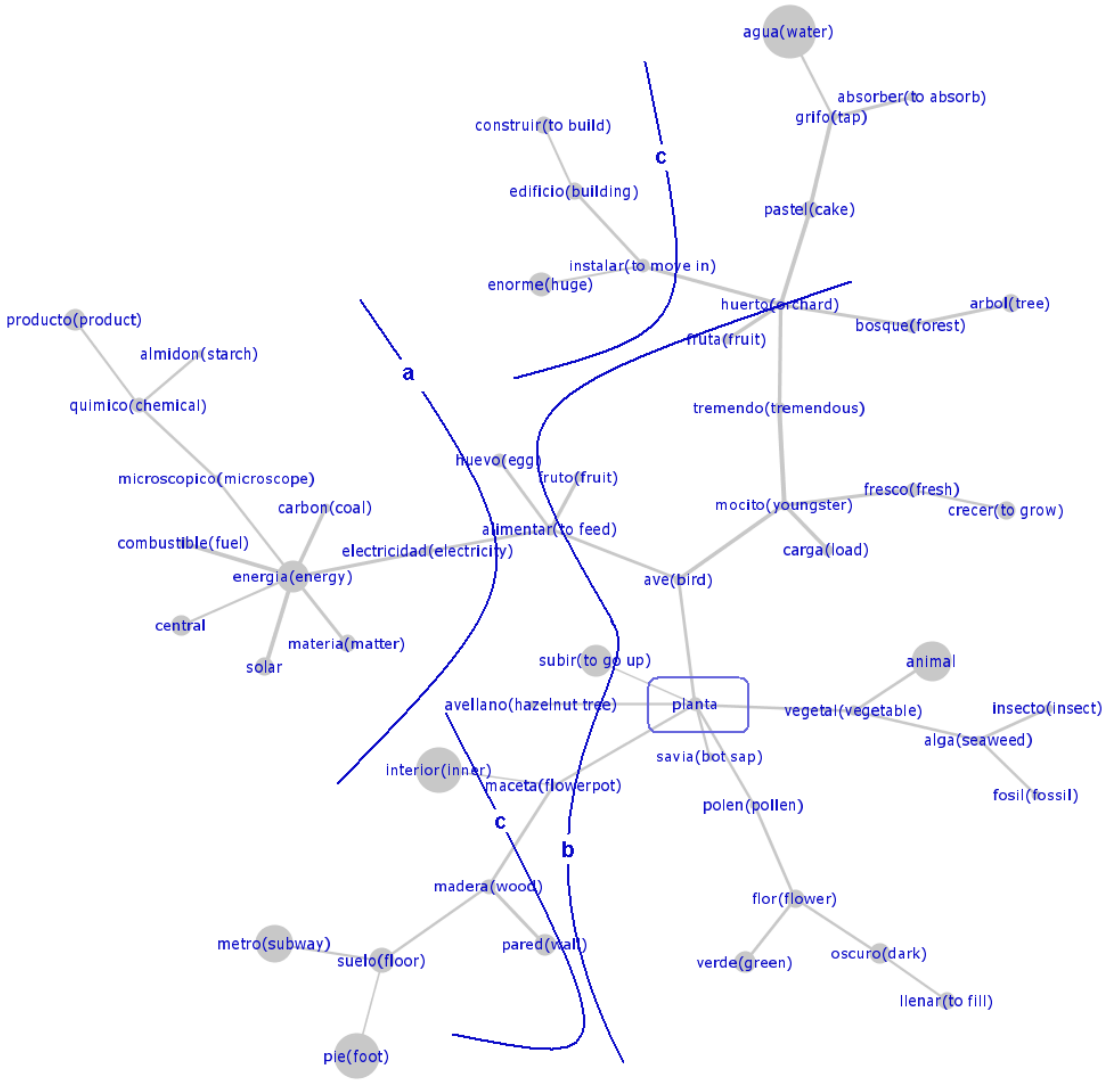


Figure 6. Visual representation of the word "*planta*" with no retrieval context word.

- *Word/Context word vector sum condition*. As shown in figure 7, "*planta*" plus its contexts, but using the vector sum method, changes the visual network representation of the word *planta*. Building-related meaning gains ground around the network, but energy-related meaning conserves some representative terms (zone separated by line a). Content related to the plant kingdom, those imposed by the context word "*rosal*" (rosebush) still occupy a strong position, albeit in a very general meaning – again we see an imprecise definition effect.

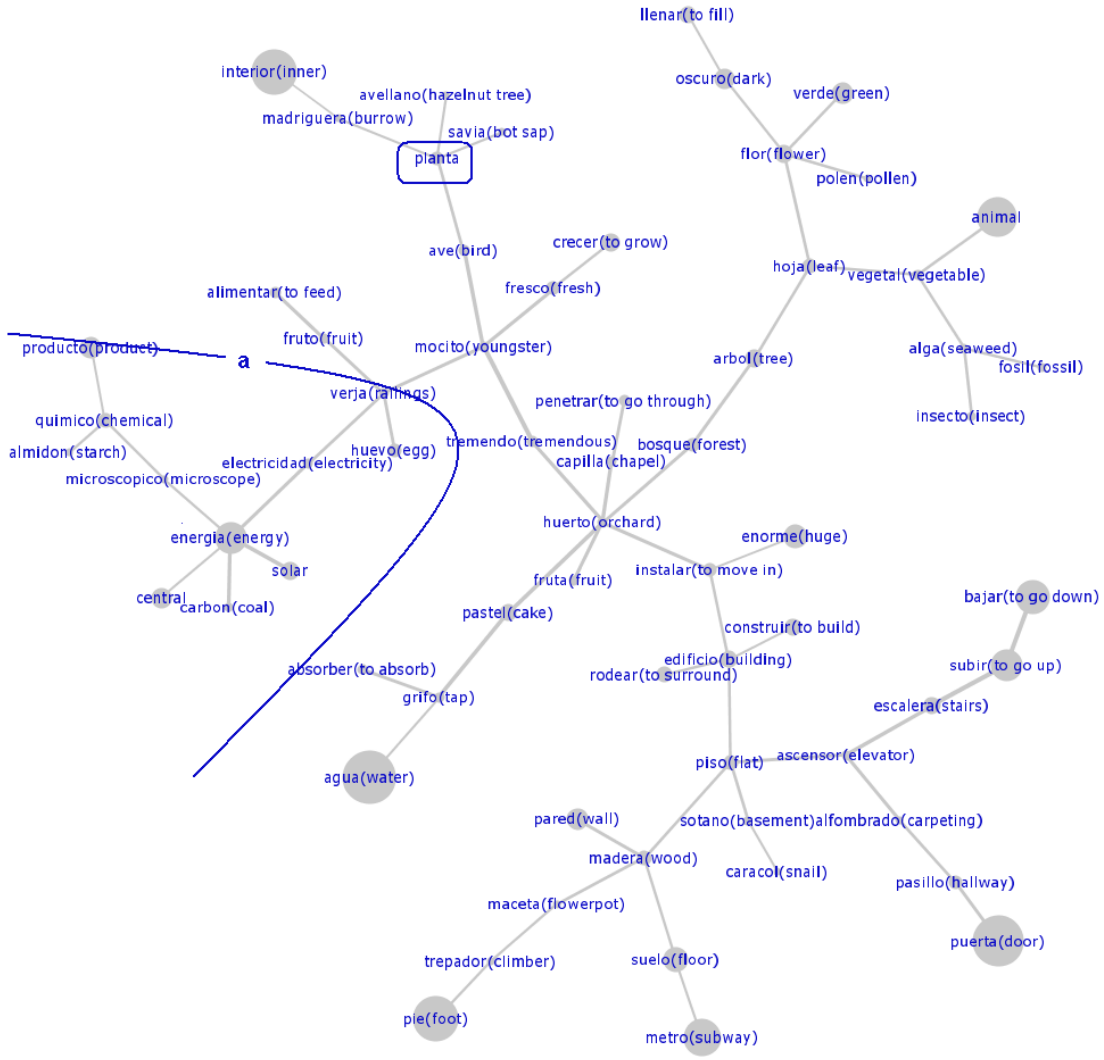


Figure 7. Visual representation of the word "*planta*" for two contexts, "*rosal*" and "*piso*" forming two structures: *Planta(rosal)* and *Planta(piso)*". The predication algorithm is not applied in this case.

- *Word/Context word with predication algorithm condition*. When we use the predication algorithm (see figure 8) energy-related meaning disappears, and building-related meaning proliferates and becomes more closely matched to the argument "*piso*" (apartment) with terms such as "*ascensor*" (elevator), "*escaleras*" (stairs), "*vestíbulo*" (hall), "*alfombrado*" (carpeting), "*edificio*" (building). Furthermore, the plant kingdom meaning was more closely related to the argument "*rosal*" (rosebush), with flower-related terms such as "*pétalo*"

(petal), “*olor*” (smell), *camelia* (camellia), “*flor*” (flower), “*perfume*” (scent) and “*aroma*” added (see the different zones separated by line a).

Examining the effect of adjusting vector length of the cosines when extracting neighbors, we again managed to avoid meaning of the nets comprising only terms from extreme local relationships (low-level definition). Terms such as “*calle*” (street), “*casa*” (house) or “*sol*” (sun), “*rojo*” (red), “*color*” (colour), “*metro*” (underground), “*subir*” (to go up), “*coche*” (car) demonstrate that high-level terms are also represented. As with the network for “partido”, terms with long vectors do not necessarily coincide with terms that have most links. Again, words with short vectors such as “*ascensor*” (elevator), “*pétalo*” (petal), “*rama*” (branch), “*flor*” (flower) or “*madera*” (wood) occur in few contexts and only in contexts relating to a single topic. This converts these terms into unmistakable features of a topic. Terms with longer vectors, on the other hand, habitually occur in a wider variety of contexts, blurring their relationship with the central concrete terms.

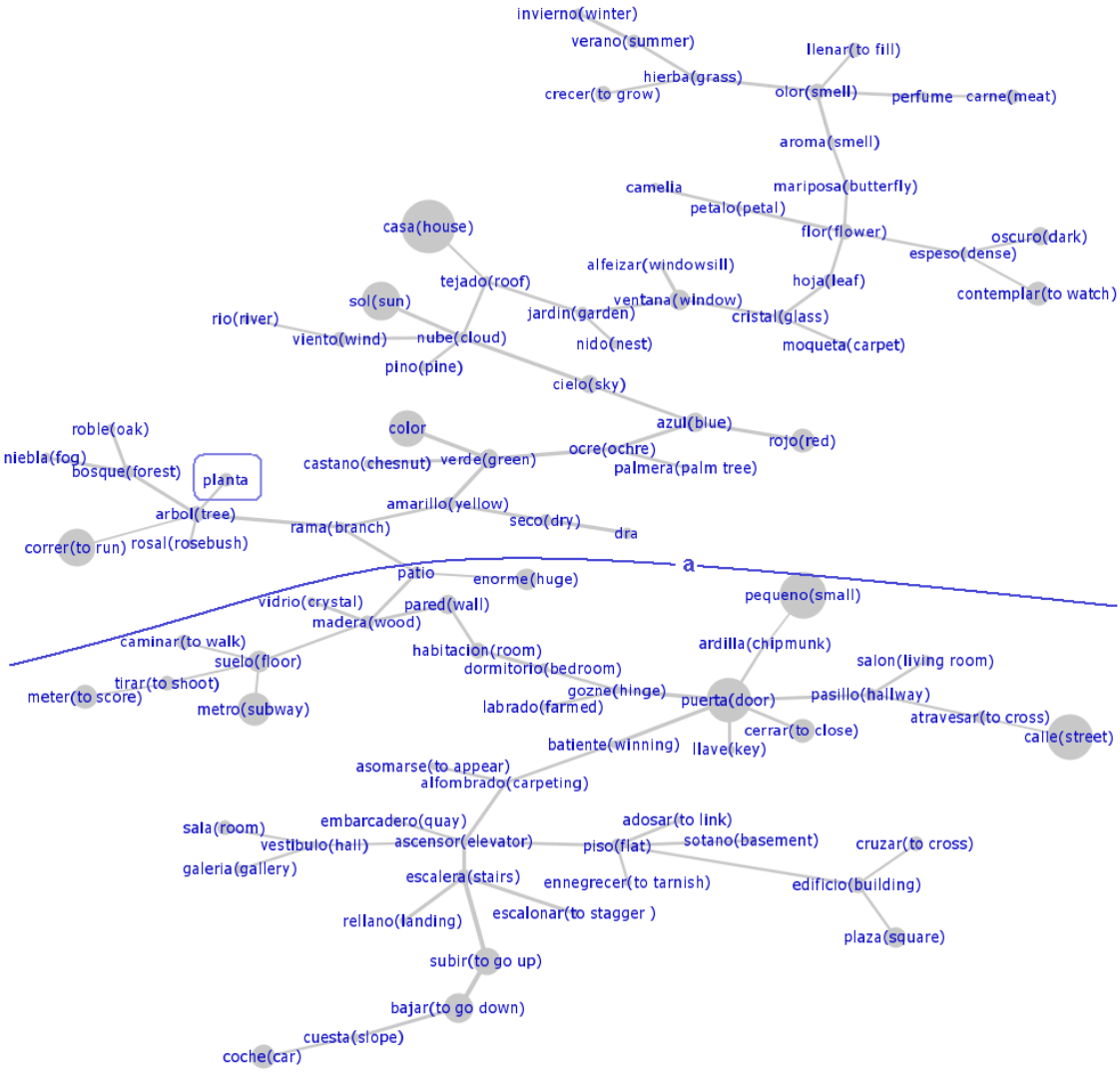


Figure 8. Visual representation of the word "*planta*" in two context domains: "*rosal*" and "*piso*", forming two structures: *Planta(rosal)* and *Planta(piso)*. This time the predication algorithm is applied.

7. Testing the networks

In order to test the suitability of using the predication algorithm in a survey of structures T(C1) and T(C2), we took a list of polysemic words and examined their similarities with possible context-words C1 and C2 in each of their possible structures. The procedure is based on a procedure explained in Kintsch (2008). The key is that every structure T(C1) must be related to its context C1 but not related with C2. The same effect must be true of T(C2), C2 and C1 respectively. With this idea in mind, we test the three conditions that

correspond with the diagrams above, taking the isolated condition as a base line.

7.1 Procedure

Our starting point is a table showing the similarities between T and the two contexts (C1 and C2) in the first and second column respectively. These two columns show the predominant meaning of the term in question T.

The next four columns show the similarity (cosines) between T(C1) and C1, T(C1) and C2, T(C2) and C2 and T(C2) and C1 respectively, where T(C1) and T(C2) are calculated using the vector sum. The last four columns show the similarity between T(C1) and C1, T(C1) and C2, T(C2) and C2 and T(C2) and C1 respectively, where T(C1) and T(C2) are calculated using the predication algorithm. These eight columns show the relationship between each structure and its possible meanings. For example, the fact that T(C1) has the same similarities with C1 and C2 would indicate that the method used to construct structure T(C1) is not powerful enough to filter out the dominant meaning of T (in this case, the meaning of C2). With this in mind we prepared such a table (Table 2), where the vector length of the two context-words C1 and C2 are smaller than the Term T.

	Isolated		Vector Sum				P. Algorithm			
	T		T(C1)		T(C2)		T(C1)		T(C2)	
	C1	C2	C1	C2	C2	C1	C1	C2	C2	C1
hoja (cuaderno/roble) leaf (book/oak)	.22	.37	.42	.37	.53	.23	.52	.25	.71	.14
copa (vino/fútbol) cup (wine/football)	.33	.47	.73	.36	.87	.2	.79	.06	.92	.09
banco (jardín/préstamo) bank[30] (garden, loan)	.11	.4	.49	.35	.46	.11	.58	.12	.69	.05
rosa (clavel/matiz) rose (carnation/hue)	.43	.16	.58	.16	.43	.41	.46	.28	.41	.38
lila (lirio/matiz) lilac (iris/hue)	.36	.25	.8	.25	.87	.28	.52	.29	.5	.46
diente (ajo/colmillo) tooth[31] (garlic/canine)	.29	.46	.42	.48	.55	.3	.61	.35	.57	.27
planta (rosal/ascensor) plant (rosebush/elevator)	.24	.27	.33	.27	.54	.23	.46	.24	.78	.14
programa (software/tv) program (software/TV)	.31	.36	.35	.36	.47	.31	.68	.3	.91	.1
caja (préstamo/envoltorio) box[32] (loan, packaging)	.15	.24	.28	.23	.33	.14	.63	.06	.63	.05
papel (actriz/cuaderno) paper[33] (actress/notebook)	.28	.26	.33	.26	.29	.28	.81	.05	.75	.22
cadena (tienda/tv) chain[34] (shop/TV)	.11	.81	.44	.75	.87	.11	.61	.18	.91	.08
partido (fútbol/nacionalista) party (football/Nationalist)	.33	.43	.5	.39	.49	.32	.96	.06	.91	.07
bomba (misil/vapor) bomb[35] (missile/steam)	.37	.35	.82	.31	.74	.35	.85	.31	.77	.32

Table 2. Similarities between structures and each context

With the table of similarities in place, we set out to calculate indices of the phenomena explained above - that is, to find whether the method used to construct the structures [T(C)] is powerful enough to filter out the dominant meanings and conserve the correct meaning. To do so, we calculate differences between cosines using the following procedure:

1) Taking an absolute value of the difference between the cosine between T and C1 and the cosine between T and C2 (column one and two of Table 2).

[30] Meanings of the Spanish noun *banco* include bank and bench
[31] Meanings of the Spanish noun *diente* include tooth and clove
[32] Meanings of the Spanish noun *caja* include box, safe-deposit box and bank teller's window
[33] Meanings of the Spanish noun *papel* include role, paper
[34] Meanings of the Spanish noun *cadena* include chain, TV/radio channel
[35] Meanings of the Spanish noun *bomba* include bomb, pump

This gives us a snapshot of the extent to which one meaning is predominant over another. The first column of Table 3 represents this calculation.

For instance, in the case of the term *Partido,* we calculate *Cosine(Vpartido , Vfútbol) = 0.33* and *Cosine(Vpartido , Vnacionalista) = 0.43.* This means that *Vpartido* has to do more with the meaning 'Nationalist' than with the meaning 'football'. The absolute value of the difference between the two cosines (0.1) shows the difference in dominance of the two context-words, one meaning being more dominant than the other.

2) For Table 2, we calculated the vector of the two structures T(C1) and T(C2) using the vector sum method, then calculated the cosine between the two structures and every context-word vector C1 and C2 (columns 3,4,5 and 6). In contrast, in this step we subtract the first cosine from the second in every structure to indicate the strength of every structure's correct meaning and filter out other meanings. In other words, *Cosine(T(C1), C1)* minus *Cosine(T(C1), C2)* and *Cosine(T(C2), C2)* minus *Cosine(T(C2), C1).* A negative value means that not only is the correct meaning not represented strongly enough but also that other meanings are better represented. The resulting values are in the second and third columns of Table 3.

For instance, the result of subtracting *Cosine(Vpartido fútbol, Vfútbol)* from *Cosine(Vpartido fútbol, Vnacionalista)* is 0.11. This value will indicate the extent to which Vpartido fútbol has the correct meaning and is not affected by the other meaning. On the other hand, subtracting the first cosine (Vpartido nacionalista, Vnacionalista) from the second (Vpartido nacionalista, Vfútbol) gives 0.17. This value will indicate the extent to which Vpartido nacionalista has the correct meaning and is not affected by the other meaning.

3) The same procedure as 2) but this time with the columns where T(C1) and T(C2) has been calculated the predication algorithm was used – that is, *Cosine(T(C1), C1)* minus *Cosine(T(C1), C2)* and *Cosine(T(C2), C2)* minus *Cosine(T(C2), C1)* in columns 7,8,9 and 10 of Table 2. Again, a negative value

means that not only is the correct meaning not represented strongly enough but also that other meanings are better represented. The resulting values are in the last two columns of Table 3.

To summarise, the last four columns in Table 3 indicate the bias toward the correct meaning represented by the vector of each structure. For example, .37 in the T(C1) column indicates that the vector of T(C1) is biased toward the meaning of C1. On the other hand, a value of .05 indicates that the vector of T(C1) is not biased toward the meaning of C1 (because similarity with the other meaning is still strong). The Isolated condition (the base line) is usually dramatically affected by the predominant meaning effect, and for this reason the differences are irregular. If any of the other conditions were affected by the predominant meaning effect, the differences will be also variable, so the mean will be similar to the base line condition.

	Isolated	**Vector Sum**		**P. Algorithm**	
		T(C1)	**T(C2)**	**T(C1)**	**T(C2)**
hoja (cuaderno/roble) leaf (book/oak)	.15	.05	.3	.27	.57
copa (vino/fútbol) cup (wine/football)	.14	.37	.67	.73	.83
banco (jardín/préstamo) bank[30] (garden, loan)	.29	.14	.35	.46	.64
rosa (clavel/matiz) rose (carnation/hue)	.27	.42	.02	.18	.03
lila (lirio/matiz) lilac (iris/hue)	.11	.55	.59	.23	.04
diente (ajo/colmillo) tooth[31] (garlic/canine)	.17	-.06	.25	.26	.3
planta (rosal/ascensor) plant (rosebush/elevator)	.03	.06	.31	.22	.64
programa (software/tv) program (software/TV)	.05	-.01	.16	.38	.81
caja (préstamo/envoltorio) box[32] (loan, packaging)	.09	.05	.19	.57	.58
papel (actriz/cuaderno) paper[33] (actress/notebook)	.02	.07	.01	.76	.53
cadena (tienda/tv) chain[34] (shop/TV)	.7	-.31	.76	.43	.83
partido (fútbol/nacionalista) party (football/Nationalist)	.1	.11	.17	.9	.84
bomba (misil/vapor) bomb[35] (missile/steam)	.02	.51	.39	.54	.45

Table 3. Differences between the similarities, with measures of bias towards a meaning.

With the table of the differences (Table 3), we conducted an ANOVA to see if the two methods are significantly different in efficiency. Using the Isolated column as our base line, the ANOVA had only one independent variable with three conditions: Isolated, Vector Sum and Predication Algorithm. The dependent variable is the difference represented in each cell of Table 3. We found homoscedasticity ($F_{Levene}(2,61) = 2.45$, $p = .097$), but we did not found normality in one of the conditions (Isolate). The F-test is generally robust against violations of normality.

7.2 Results and discussion

The ANOVA shows a main effect between the three conditions [$F(2,61) = 11.31$, $MSE = .059$, $p < .05$]. The means of Isolated, Vector Sum and Predication algorithm are .16, .23 and .50 respectively. The Bonferroni correction shows that there are no significant differences between Isolated and Vector Sum, while there is a significant difference between Isolated and Predication Algorithm ($P<.05$) and between Vector Sum and Predication Algorithm ($P<.05$). Although it displays more variability, the fact that the Vector Sum condition shows no significant differences from the Isolated condition indicates that the Vector Sum method is not usually powerful enough to cope with the predominant meaning effect of T. In other words, when we calculate the structures T(C1) and T(C2) with Vector Sum, it may be that one of them is still dependent on the dominant meaning of T - the alternative meaning. The Predication Algorithm appears to be less biased by the dominant meaning of the terms, and the structures T(C1) and T(C2) are better represented.

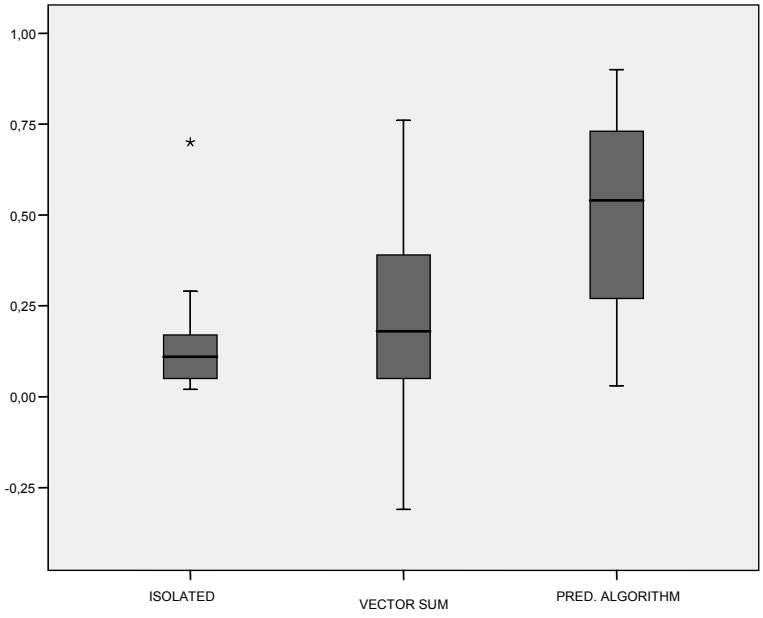


Figure 9. ANOVA of the three conditions

8. General discussion

In this study, we focus on polysemic words using a computational method (LSA) as a static base for word representation, and the predication algorithm as a filtering mechanism to disambiguate words with several meanings. We have shown some methods to visually check the role of context during retrieval of the meanings of some terms. Although previous research has used larger units such as sentences or paragraphs to implement context (Lemaire et al., 2006; Olmos et al., 2009), we have chosen the context provided by another word. This choice is a more parsimonious way of drawing on concepts without dispensing with the details of subtle changes. In addition, we have predicted unwanted effects that may be observed and some proposals for avoiding them. In order to clarify the impact of each effect, we first drew a condition in which a word is used without any context words, to visualize the baseline for each polysemic word and its definition bias. We then tested each word followed by each of its context words, using two methods – first the simple vector sum, which did not eliminate definition biases, and second the

predication algorithm, which proved an efficient method to avoid some biases and to simulate comprehension of some predicative structures and term-context structures (Kintsch, 2008).

Some interesting conclusions can be drawn from this study. One is related to the representation of words independent of context (isolated condition) which may vary in terms of how closely it matches reality. It is occasionally biased by how representative the different meanings are within the semantic space. Sometimes, we find that only the most representative meaning for this corpus are present (predominant meaning inundation). Sometimes there are several meanings surrounding the word but lacking detail (imprecise definition) – we see no more than general meanings of words, never sub-domains. In other cases, the retrieved list of terms actually contains terms unconnected with any of the meanings e.g. "*mocito*" (youngster) in figure 5, perhaps due to the features of the vector in isolation from context. As explained in the introduction, the dimensions of an LSA vector-term (especially the vectors of polysemic words) are context-free and biased by the frequency with which a term occurs in the document, resulting in a vector-prototype that is a cluster of features of all meanings. This resulting vector-prototype used to establish similarity is in fact a very atypical member and can sometimes promote spurious relations.

Another conclusion is related to the representation of words dependent of context. In the case of *Word/Context word*, when we compound the vector with the simple sum of the word and the context word, we find that sometimes the dominant context has accounted for all the nodes (predominant meaning inundation). This was probably caused by insufficient representation of one of the context words in the semantic space. For instance, if the predominant meaning of a word is *X* and we add another context *Y* which does not have sufficient vector length to compete with the predominant meaning, then meaning *Y* will only result in a few terms and the representation of X will be strengthened. This is why the meaning for the isolated word condition and the meaning for a word followed by two context words might not vary – as we observed for the word "*partido"*. If we use the simple sum of vectors, we can

see that the context "*fútbol*" (football) does not have sufficient vector length to retrieve more than a few examples. In others cases (again with simple sum of vectors), the visual representation of two predications conserves other meaning that does not correspond to these contexts. For example when we extract the graph for *Planta(rosal)* and *Planta(piso)*, using simple sum of vectors the energy-related meaning of "*planta*" is conserved in the shape of terms such as "*químico*" (chemical), "*energía*" (energy), "*central*" (power plant) and "*electricidad*" (electricity). We can briefly summarize the results of simple vector sum by saying that with this method we are exposed to the influence of words' vector lengths. We can sometimes obtain reasonable representations but run the risk of obtaining only predominant or generic meanings. The fact that the ANOVA detected no significant differences between the Isolated and Vector Sum conditions in the second part of the article indicates that this affirmation may well be true. It also confirms what we had seen visually: those structures calculated using the Vector Sum method are still dependent on the dominant meaning of the terms.

When we draw the *Word/Context word* structures with the predication algorithm, it seems we are able to correct these two problems. Content that does not match the arguments was eliminated, and all pertinent meaning was well represented. This advantageous method for drawing contexts relies on the way in which the algorithm works. It primes terms extracted using the word which are more relevant to the context words. This method aims to ensure that the final product contains vectors representing the dimensions relevant to the word and to each retrieval context. This provides a very detailed list of neighbors, representing ample examples of each argument which are well distributed around the network. The fact that the ANOVA detected significant differences between the Predication Algorithm and Vector Sum conditions in the second part of the article indicates that the Predication Algorithm is less biased by the dominant meanings, and operates more satisfactorily, as we had seen visually.

Concerning results of the vector length correction applied in previous studies with a specific domain corpus (Jorge-Botana et al., 2009), we found that

using a more general corpus such as LEXESP, this technique also helps to ensure that some frequent and important words are represented in the network. This is due to the actual correction mechanism used to extract part of the list of neighbors. This mechanism gives the most representative terms from the semantic space priority as neighbors (although this priority is not mandatory). This means that representative nodes as well as local relationships were represented visually, introducing some psychologically plausible representation. Such frequent and important terms, however, do not necessarily form more links. In fact the opposite is usually true: these frequent and important terms with larger vectors do not usually occupy nodes with many links from other words. We have concluded that this may be due to terms with larger vectors generally being more general terms and occurring in a variety of contexts. For this reason they do not have such a strong relationship with all the topic-related terms. On the other hand, terms with shorter vectors often appear in a single context, making them unmistakable features of that topic.

Conclusion

In general, what is remarkable about the LSA model is that the structural similarity of the resulting vectors appears to parallel semantic similarities discerned by human subjects between words in the corpus – sometimes with surprising accuracy. Semantic spaces formed using LSA have offered pleasing results in synonym recognition tasks (Landauer & Dumais, 1997; Turney, 2001), even simulating the pattern of errors found in these tests (Landauer & Dumais, 1997). Using LSA it has even been possible to study the rate of knowledge acquisition relating to a term, via exposure to documents in which it does not appear (Landauer & Dumais, 1997). Such correspondences would seem to suggest some non-arbitrary relationship between the representations computed by LSA-type methods and our own cognitive representations of word meaning. This ensures that LSA is a good basis for applying objective rules from some models of cognitive processes and extracting reliable results

Since Kintsch (2001) proposed a psychologically plausible way to simulate comprehension of predication – using LSA as a lexical base and

applying objective cognitive rules to it – we have a very intuitive means for formal understanding of what the system does during comprehension of some linguistic structures, and the role of the context of word retrieval. The predication algorithm applied to word pairs (Kintsch, 2001) and the predication algorithm applied to dependency relations within sentences (Kintsch, 2008) are effective methods for differentially retrieving the meaning of words according to the context imposed by arguments in propositions, and has proved better than the traditional method of using simple sum of the vectors representing argument and predicate.

In this study, we have presented a protocol for visualizing the contexts that a word can take on, and have outlined the procedure to show the meanings of a word in isolation, the meanings of a word with arguments but without using the predication algorithm, and the meanings of a word with arguments using the predication algorithm. A well-managed context ensures good representation of the meanings we wish to retrieve, as shown intuitively by the visual nets, and rather more explicitly in the results of the ANOVA.

This kind of human-based method could be used in retrieval applications or in indexing or tagging machines, or even in the application of top-down processing rules – for instance for improving confidence measures in speech recognition machines, constraining the likelihood of a recognized term using the context. In addition, the nets produced by these methods could be used as a visual information retrieval interface (VIRI), allowing users to visually recognize the information that is needed, instead of writing a query in the search boxes or providing an overview of a semantic domain, and helping the user to know what information can be retrieved using the interface.

ACKNOWLEDGEMENTS

This work was supported by Grant SEJ2006-09916 from the Spanish Ministry of Science and Technology and PSI 2009-31932 from the Spanish Ministry of Education.

Capítulo 10

Monitoring the penalization/advantage of lexical ambiguity in vector model representations

(Artículo en revisión en la revista Cognition)

Monitoring the penalization/advantage of lexical ambiguity in vector model representations

Guillermo Jorge-Botana, Ricardo Olmos & José A. León

Universidad Autónoma de Madrid

Abstract:

For a long time now, the ambiguity of words has been an area of interest in many disciplines. Many studies have attempted to demonstrate how lexical ambiguity might be represented in the brain. For some researchers each sense has a distinct representation; for others, there is a common or "core" representation, plus a specific partial representation for each sense. Others propose that one meaning might be activated and another inhibited in a single representational space. There is still an added controversy, however, to the definition of ambiguity: the diversity of contexts a word appears in - the distribution of a word across possible contexts. Vector space models such as LSA(Latent Semantic Analysis) might be very useful for putting such a definition to the test. Having an objective, discrete model of lexical representation allows us to objectify some parameters to define ambiguity in a more measurable way - for example, defining ambiguity (even abstractness) phenomena with vector metrics. We investigate whether the empirical data on ambiguity and abstractness in the Lexical Decision Task and in Sense Retrieval can be modeled by means of an exclusively linguistic single representation model such as LSA. Our results support the idea that ambiguity effects in LDT and in Sense Retrieval can be emulated with such a model, as can the relations of abstract words with the other terms. An additional source of activation is needed, however, to explain the concrete superiority in the LDT (sensorial representations are one possibility).

1) Introduction

For a long time now, the ambiguity of words and their resulting disambiguation (Word Sense Desambiguation) has been an area of interest in many disciplines. Lexicographers have attempted to identify when the usage of a term modulates its meaning or configures a new one. To this end, lexicographers have proposed some tests to clarify the issue, although none has been definitive (see Kilgarriff, 1997). The fact is that in a dictionary it is difficult to define when the usage of a word pertains to a meaning other than those already established - in other words, when we come across cases of homonymy or polysemy. This dilemma has also been a concern in the areas of cognitive linguistics and psycholinguistics, where studies have attempted to demonstrate how lexical ambiguity might be represented in the brain. Some models have proposed the existence of different entries for each of the meanings of ambiguous words (Klein and Murphy, 2001) while others propose the existence of a common representation for all the meanings and specific representations for each of them (Rodd, Gaskell and Marslen-Wilson, 2002). The proportion of the representation that is common would depend on the degree of homonymy/polysemy the meanings of a word have. In addition, vector space models such as LSA (Deerwester, Dumais, Furnas, Landauer and Harshman, 1990), HAL (Burgess and Lund, 2000) or the topic model (Griffiths & Steyvers, 2004; Steyvers & Griffiths, 2007) have proposed that a single representation for each lexical unit, along with context information, is enough to account for this phenomenon.

One less well-explored dimension of the phenomenon of ambiguity examines the occurrence of a given word in a range of contexts. By definition, a word that occurs in several contextual units (defining each unit operationally) will have a greater degree of ambiguity. The distribution of each word among these units has to be specified in its representation, whatever that may be. To record this distribution, some indices drawn from corpus linguistics have been extracted. Entropy formulae (see: Nakov, Popova & Mateev, 2001), for example, where words with greater entropy contain less information about the contexts in which they appeared, or *Contextual Diversity*, normally measured as

the number of documents in which a word appears, which has a demonstrable effect on response times independent of imaginability (Adelman, Brown & Quesada, 2006). The idea of word distribution has been used in an attempt to obtain standard indices of the ambiguity of words in the technology industry.(see US patent 6,256,629 B1).

Finding this type of definition of ambiguity (objectively defining distributions) has an impact on theories of lexical representation, as we can identify what type of distributions facilitate effectiveness in certain tasks (naming, lexical decision, semantic tasks, etc.) and thus use them to justify empirical data. It might lead, for example, to phenomena such as the concretion/abstraction being explained in terms of this distribution, within a framework where no reference to primary representations is necessary.

Vector space models such as LSA have produced good results simulating lexical representation in humans (Landauer and Dumais, 1997; Kintsch and Mangalath, in press), and it is easy to parameterize vectors, so these models might help to define these indices and throw some light on the theories of the lexical representation.

2) Abstract and polysemic words: Both sides of the coin of ambiguity?

The full definition of ambiguity is still somewhat controversial. It is difficult to know what linguistic categories might be included (for a review see Charles Lin and Ahrens, 2009). Basically, the generic definition of lexical ambiguity is that a word can be interpreted in more than one way, and disambiguation is a function of the recall context. The difficulty lies in defining which linguistic phenomena to incorporate within the category, and how to position them. Some authors have supported a classification in the form of a continuum, with homonymy at one extreme and metonymy at the other. In the middle would be metaphors (Klepousniotou, 2002). This continuum is justified based on whether the meanings are independent of one another or interrelated, homonymy having its independent meanings and metonymy its completely related meanings.

Metonymy and polysemy are partners, and metaphors resemble a weak form of homonymy. A good criterion for identifying when ambiguity has independent or related meanings is that offered by Ahrens (1998): Two meanings belong to different senses if one of them is not an instance of the other that has been formed from metonymic or meronymic extensions (not the case for metaphors) - if the instances of both meanings cannot inherit from the same class of nouns, and if both meanings cannot appear in the same contexts. Two meanings are different aspects of a single thing (Ahrens calls them meaning facets) if they are formed from metonymic or meronymic extensions, if they can inherit from the same noun classes and if they appear in similar contexts.

As for the representation of different senses in the brain, one group of models proposes that each sense (regardless of homonymy or polysemy) is represented in a different way - in other words each sense has a representation (Klein and Murphy, 2001). Nonetheless, results such as those of Williams's (1992) asymmetric priming for both senses using a list of ambiguous words, suggest a common representation, though with specific representations for each meaning. In *Williams*'s experiments the predominant sense components of an ambiguous word remain activated in the context of another sense. However, the components of the non-predominant meaning do not remain activated in the presence of an incoherent context. This type of findings has led to a second set of theories, which propose that there is a common or “core” representation, plus a specific part representation for each sense, and that the proportion of one to the other varies along the homonymy /polysemy continuum (Rodd, Gaskell and Marslen-Wilson, 2002). The suggestion of a shared portion and separate portions is a valid explanation but does not invalidate a third group of models that proposes a single representation for each lexical unit regardless of how many senses it has. The key lies in the fact that one meaning might be activated and another inhibited in a single representational space, using the context (Kintsch, 2001; Kintsch and Mangalath, in press). And this might occur even if both senses are completely orthogonal (such as homonymy). In the case of homonymy, the facilitatory contexts would not share activated dimensions even if both senses were in the same representational space and this space correlated with a lexical unit (a word expressed orthographically or

phonologically). The conclusion would be that there is no possibility of meaning in the absence of context - this would be unrealistic given the characteristics of single vector representation (Jorge-Botana, León, Olmos & Hassan-Montero, in press). These postures are supported by some vector space models, which include the models based on Latent Semantic Analysis (LSA)

But there is still an added controversy to the definition of ambiguity. The criteria used to categorize the dependency or otherwise of senses (and when a sense might configure an independent representation) exclude other possible types of definition, such as that derived from the diversity of contexts a word appears in - the distribution of a word across possible contexts. A word might appear in many contexts, few contexts, or even only one in the case of pure monosemy. With this in mind, the present study will introduce another key phenomenon into the ambiguity equation: abstraction/concretion. Polysemic words/homonyms and abstract words would be then at the same conceptual level. We justify this working hypothesis since the two categories frequently covary and because their definitions might meet the requirements that define ambiguity: generalist, indeterminate, vague or lacking specificity (Kilgarriff, 1997). In the case of polysemy and homonymy, the definition would be that there is no single unitary focus for the referents - there may be several. For abstract words, the definition would be that there is a total lack of stable focal referents (Grossman, Koening, DeVita, Glosser, Alsop & Detre, 2002; Barsalou & Wiemer-Hastings, 2005). The possible structural similarities of polysemic and abstract words have been supported using objective measures such as Contextual Diversity (Adelman, Gordon, Brown & Quesada, 2006), a concept that is directly proportional to the number of contexts a word appears in. Words with more than one meaning (polysemic words) have greater Contextual Diversity than monosemic words, just as abstract words have more Contextual Diversity than concrete words. The concept of *Distribution generality also seems to support this* (Weeds, Weir and McCarthy; 2004). A given word X has a greater distribution generality than Y if X appears in more contexts than Y. The same observation has been made by Barsalou & Wiemer-Hastings (2005) about abstract concepts: The coordinates of concrete concepts are focused on a spatially circumscribed region of the subject area, whilst the coordinates for

abstract concepts are distributed across a variety of focal points, whether physical, mental or otherwise. With this in mind, we might propose a working hypothesis that both polysemy and abstraction/concretion might be explained using a theoretical framework that works with a common substrate (the distribution of words across contextual units). In other words, abstract words differ from concrete and polysemic words differ from monosemic in terms of the distributional properties of their representation.

This hypothesis explains abstraction/concretion without incorporating a factor that has been used to define abstract words in some of the most significant theories - that what distinguishes them from concrete words is that the latter are also represented by their imaginability. In other words, concrete words have support from primary sensorial representations (Paivio, 1986, 1991). For this reason abstract words and the concrete words are represented in different networks. By leaving imaginability out of the explanation we align ourselves with second order isomorphism theories (Shepard, 1987), which propose that the meaning of a linguistic symbol is not defined in terms of other levels of representation (perceptual representations) but rather by its relationship with other symbols. In other words, verbal information is largely a reflection of perceptual information, and this verbal information is enough for us to process without access to other representational levels such as perceptive factors (Kintsch, 2008). The fact that a purely linguistic model such as LSA, without references to any perceptive aspect, has properly simulated human representation and has been productive in some human tasks such as judging similarity (Landauer and Dumais, 1997) seems to suggest that the theories that propose such an isomorphism are in no way ridiculous, and that it is similarity to other symbols provided by a certain distribution of its defining dimensions, that might explain abstraction/concretion phenomena.

Although there is a large amount of data defending the perceptive richness of concrete words compared to abstract (Paivio, 1986, 1991) some empirical findings also seem to refute it. These other studies suggest that the difference between abstract and concrete words does not lie in their imaginability. They include behavioral experiments (Samson & Pillon, 2003) and

experiments using brain imaging when tasks require more sophisticated semantic analysis (Pexman, Hargreaves, Edwards, Henry & Gooyear; 2007), and even resulting in a decrease of the use of imagery when children become older (Schwanenflugel & Akin, 1994).. For example, Pexman et al. attributed the difference between abstract and concrete words to a simple question of the extent of activation, without clear differences in terms of the brain area activated.

The vector space models might be very useful for putting this type of hypothesis to the test. Having an objective, discrete model of lexical representation allows us to objectify some parameters to define ambiguity in a more measurable way. A model such as LSA, an exclusively linguistic model, can help to give a definition of the linguistic stimulus that is processed by the human system. Once the stimulus is defined, we can correlate their intrinsic parameters with the response to tasks for which we have empirical data. In this way, we can predict that certain parameters (a parameter might be the Contextual Diversity mentioned above) cause certain empirical phenomena. For example, if we objectify Contextual Diversity in an LSA vector we can measure this parameter in different categories of words that give different results in certain tasks (e.g. Lexical Decision Task). If the difference is significant, we can suggest that this parameter might be involved in the variability of results. Besides, we can simulate the phenomenology based on the relationship that a type of LSA vector, differentiated by this parameter, has with the other vectors in the semantic space - in other words check if the vectors whose distributions show greater Contextual Diversity can simulate the same effects that polysemic and abstract words show in the empirical data.

Based on the arguments above and on the advantages offered by a model of lexical representation that can be parameterized (such as LSA), we will analyze which parameters might define ambiguity, test a sample of polysemic words and also introduce the initial hypothesis that abstract words might also be ambiguous, just as polysemic words are. In addition, to attempt to clarify the extent to which we can talk of the actual substrate of processing, we will focalize our study on two phenomena for which the behavior of both

abstraction/concretion and polysemy/monosemy has been described: Associative difficulty and LDT (lexical decision). The fact that LSA is an exclusively linguistic model serves as a test of whether these phenomena can be justified using only linguistic properties. They will be presented in the following sections.

2) Associative difficulty

The definition of ambiguity has been stated in several different ways, but it is the definition comprising *difficulty of association*, offered by Brown & Ure (1969), which is more widely used than the vector space models. Here it is plausible to check the type of relationships words might take on within their semantic space, or what type of relationships are more unlikely given a word type and its vector representation.

Firstly, polysemy and homonymy seem to be defined by the fact that their representation has several subject area focal points. As a result, the relationships it might form with other words are modulated by a uniting context which amplifies one of the meanings. Duffy, Morris & Rayner (1988) found for example that ambiguous words with equally probable meanings that are not biased toward a given context draw longer fixation times from readers. When the meanings are not equally probable (there is a dominant meaning) there is no difference in fixation times compared with non-ambiguous words. Moreover, when there is a predominant meaning and we bias towards the non-predominant meaning, the differences in favor of the non-ambiguous meaning arise again. In summary, polysemic words seem to cause some processing difficulties in retrieving sense in the absence of a context.

And secondly, some studies have found that concrete words will activate other words primarily through semantic similarity, while abstract words do so by association (e.g. Crutch, 2006; Crutch, Ridha & Warrington, 2006; Crutch & Warrington, 2005; Warrington & Crutch, 2007). This type of relationships seem to have even been found in experimental data with aphasics and with the normal population (Crutch & Warrington, 2005; Duñabeitia, Avilés, Afonso,

Scheepers & Carreiras, 2009). With aphasic subjects, Crutch & Warrington find that an abstract word suffers more from interference if it is inserted in a string of words that are related by association than by synonymy relationships. The opposite occurs with concrete words. With normal subjects, Duñabeitia et al. find that abstract words promote fixation on pictures related by association. Theoretically speaking, this theory means that abstract words are related inside an associative network, while concrete words are related inside a categorical network. At the same time, the Context Availability Model (Bransford and McCarrel, 1974) predicts that it is easier to generate contexts for concrete than for abstract words. Accompanying the abstract words with a context, concrete words lose their advantage in response times for LDT, naming and meaning judgment. The context effect has a marked effect on abstract words but only a minor effect on concrete words (e.g. Schwanenflugel, Harnishfeger & Stowe, 1988; Schwanenflugel & Shoben, 1983; Schwanenflugel & Stowe, 1989)

This effect has also been observed in semantic-vectorial spaces. Wandmacher (2005) suggests some qualitative data in terms of the relationships of ambiguous words in an LSA semantic space without context. The procedure consisted of extracting semantic neighbors, categorizing the type of relationship each word has with each neighbor. For example there are semantic relationships (synonymy, antonymy, hypo/hypernymy, meronymy), morphological relationships, associative relationships and erroneous relationships. With this type of relationships we find as a rule that words assigned to a theme (concrete) have less erroneous relationships than words that are context-independent (abstract). In other words, the terms with most erroneous relationships seem to occur in many contexts: “think”, “example”. At the same time, nouns have a lower rate of erroneous relationships than verbs and adjectives, and verbs and adjectives have less semantic relationships than nouns - perhaps because verbs and adjectives are more abstract or ambiguous. All this seems to show that LSA models are also sensitive to semantic ambiguity, and might be a reflection of findings in empirical data.

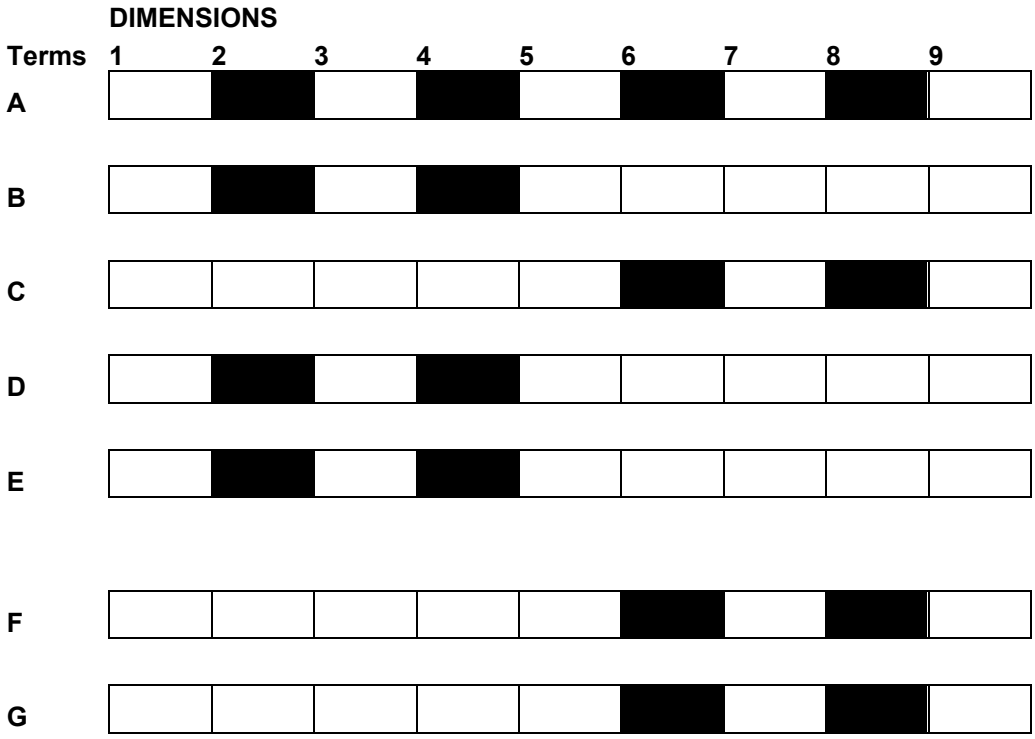


Figure 1: Simplified representation of some kind of words in Vector Space Models. Being A an ambiguous word and B, C, D, E, F and G unambiguous words.

Why is this type of relationships found both in studies with humans and using LSA simulations? How might we explain this using vector space models? Taking as a valid hypothesis the idea that LSA models are a plausible form of representing how words are stored in the brain (Landauer & Dumais, 1997) and analyzing the phenomena based on them, we can say that the most credible hypothesis is that the distribution of vector properties (only linguistic properties) might itself explain the relationships that these words may or may not enter. As Figure1 shows, the dimensions that represent one of the meanings of an ambiguous word (Figure 1: Dimensions 2 and 4 in vector A) promote similarity with some terms (with vector B) but at the same time these dimensions will be penalizing similarities with the other meanings (with vector C). The same effect will be found with the dimensions of the other meaning - in other words, that it will favor relationships with the terms with this meaning but penalize relationships with other meanings (A with C). In this way, vectors that represent many meanings will be penalized for taking on synonymy relationships, displaying the so-called associative difficulty. This effect is not found in unambiguous words, as the dimensions that represent them promote only

relationships with a single meaning, without penalizing others (D with E and F with G). This theoretical framework could be upheld in all linguistic definitions that fall within the definition of ambiguity, something that can contain phenomena such as polysemy, homonymy and even abstraction.

3) Advantage in LDT (Lexical Decision Task)

At the same time, there is a second effect described in some studies that concerns ambiguous polisemous words. Some studies have found that in lexical decision tasks (LDT) ambiguous words require a shorter response time (Hino & Lupker, 1996; Pexman & Lupker, 1999). Many of these studies argue that response times in LDT are favored by the fact that with a single orthographic representation more semantic entries will be activated (a pattern of activation that amalgamates several of the meanings found), and therefore in turn more orthographic entries (with top-down processing). This non-specific activation is enough to cross the decision threshold to say that it is a word, earlier than for non-ambiguous words (Joordens & Besner, 1994; Besner & Joordens, 1995; Pexman, Lupker & Hino, 2004). One of the models that has formalized this phenomenon with greatest precision is the "efficient then inefficient" model (Piercey & Joordens, 2000). This model arose in order to accommodate the dissonant behavior of ambiguous words in the lexical decision task (advantage) and reading (disadvantage) into PDP models. It proposes two steps when processing ambiguous words, the first where a non-specific activation is generated and a second step where activation moves toward certain patterns of meanings. If we assume that the lexical decision is previous to full semantic analysis and does not require detailed analysis, and if we assume that reading does require such semantic analysis, it follows that ambiguous words have an advantage for the LDT by generating more non-specific activation, but that this advantage is lost in reading as their multiple meanings mean the second step takes longer, as activation is channeled toward certain patterns.

If abstract and concrete words (like polysemic and monosemic words) are distinguished only in the way they are distributed across different contexts, LDT times should be proportional to their Contextual Diversity. Since they are

distributed across a greater number of contexts, abstract words should demand a smaller investment in terms of response time. However, this does not seem be the case. The experimental literature reveals the superiority of concrete words in the lexical decision task (LDT) (Schwanenflugel& Shoben, 1983; Bleasdale, 1987; Howell & Bryden, 1987; Schwanenflugel, Hamishfeger & Stowe, 1988; deGroot, 1989). Perhaps either abstract words have less contextual diversity or concrete words are activated by another type of non-linguistic entry.

According to vector models, the ambiguity effect in LDT might be due to the following: owing to the very distribution of the vectors of ambiguous words, the relationships with their n first semantic neighbors are lower in similarity than those non-ambiguous words have with theirs (associative difficulty). At the same time, however, ambiguous words have a certain advantage in forming relationships with the rest of the neighbors, as ambiguous vectors represent more content than unambiguous, and therefore may fit more easily into the patterns of the other terms.

In short, ambiguous vectors have more difficulty matching the patterns of their closest neighbors (associative difficulty) but match more closely with all others, the latter causing greater marginal activation than the unambiguous vectors. This would also be the case for abstract vectors in such a model. Abstract vectors have more difficulty matching the patterns of their closest neighbors, but match more closely with all others generating the non-specific activation. Therefore, in line with this framework, it is probable that concrete words are activated by another type of non-linguistic entry to generate more non-specific activation than abstract words do. We will examine this issue.

4) Aims

One of the characteristics that define ambiguous words is the type of relationships they promote with other terms and how they behave in the lexical decision task. In this study we propose the hypothesis that these phenomena might be explained using a single representation of an exclusively linguistic model such as LSA. However, since we expect that abstract vectors have larger

cosines with the second set of neighbors (generating the non-specific activation), LSA has difficulty explaining the empirical data on LDT (concrete superiority). With LSA we can only justify the associative difficulty of abstract words, but not the superiority of concrete words in LDT. Such a superiority would be justified with another source of activation (non-linguistic source).

The hypotheses are as follows:

1) For the first phenomenon - that ambiguous and abstract words have an associative difficulty in their relationships - we offer the hypothesis that the very distribution of the vectors of ambiguous and abstract words penalizes semantic relationships. Operationally, this can be defined by the fact that the first semantic neighbors of both types of vectors have lower cosines than unambiguous and concrete vectors. In other words, non-ambiguous and concrete vectors are related more closely with their closest neighbors than ambiguous and abstract vectors respectively. The fact that we could explain the associative difficulty in abstract words with the same pattern as ambiguous words, might confirm that abstraction can be considered an extreme variant of ambiguity, where there is a total lack of stable focalization in terms of its referents.

2) For the second phenomenon, the behavior of ambiguous words in the LDT, we offer the hypothesis that above a certain threshold (that defined by its n first neighbors) the penalization is inverted and ambiguous vectors seem to relate with their second set of neighbors more closely - in other words with greater cosines. The relationships with this second set of neighbors, far more numerous than the first, is the way of operationalizing the non-specific activation defined in studies as the cause of the superiority of ambiguous words in lexical decision tasks.

3) For the third phenomenon - the superiority of concrete words in the Lexical Decision Task within the experimental literature - in our space model (with only linguistic features) we expect the patterns of ambiguous vectors for

the second phenomenon to be repeated with abstract vectors. That is, that abstract vectors seem to relate with their second set of neighbors more closely than concrete, hence more activation is generated. This is because we propose that both ambiguous and abstract vectors have more Context Diversity than unambiguous and concrete vectors respectively. If this is the case and abstract vectors follow the same pattern, the superiority of concrete words in the LDT cannot be justified based on the linguistic distributional properties, so it seems necessary to put this down to additional activation derived from a non-linguistic entry to generate non-specific activation. This type of received activation would be enough to counter the activation of abstract words owing to the distribution of vectors across different linguistic contexts. In this way the dual model's reference to another type of entries would be justified, and although not in different networks, we might well propose that activation is received from another type of entries, just as other models have formalized connections between the semantic entries and the phonological and contextual entries (Seidenberg & McLelland, 1989). Similarly, we might hypothetically propose that if extralinguistic entries promote the activation necessary for the LDT, the fine associations might be largely promoted by the actual distribution of words across linguistic contexts.

5) Studies

Three studies will be carried out, the first introducing an artificial word in a semantic space, which will act in several ways along the ambiguous-unambiguous continuum. In the second study we will check whether the patterns obtained in this first simulation are repeated with monosemic vs. polysemic real words. Thirdly, we will check if this pattern is repeated with abstract vs. concrete real words, taking both types of words.

5.5) Study 1: Simulation with a made-up word.

5.5.1) Procedure

To check the hypothesis of the distributional properties of vectors as the cause of the effects explained in ambiguous words, we carried out a simulation task on the Spanish corpus Lexesp (Sebastián, Cuetos, Carreiras & Martí, 2000) in a semantic-vector space such as LSA. The simulation consisted of giving a made-up word several forms along the polysemy-monosemy continuum, in other words sometimes monosemic, sometimes polysemic with one of the meanings dominant and other times polysemic with both meanings equally probable. To do so, texts were created with two possible meanings of the made-up word: "noray" as a sport and "noray" as an old district demarcation. Several corpora were trained, where "noray" appeared in the texts with different proportions of the two meanings. For example, in one corpus "noray" appears in 25 documents as a sport and in 5 as a neighborhood. In total 5 corpora were trained with the following proportions: 30-0, 25-5, 15-15, 5-25, 0-30. As we can see, 30-0 and 0-30 represent pure monosemy, 25-5 and 5-25 represent polysemy with a dominant meaning and 15-15 represents equally probable polysemy.

The simulation was analyzed in two ways: (1) by recording the intensity of relationships with the other terms in the semantic space (relating this term vector with other term vectors) and checking whether the distribution of similarities with its neighbors might explain the associative difficulty and non-specific activation hypotheses. (2) studying the distribution of the vector to check possible causes of these effects based on the explanations of vector distribution (figure 1).

5.5.2) Method

The method is as follows: A made-up word, "noray", is introduced in two types of texts, where this word takes the value of an old district demarcation or a

traditional sport. 30 texts were prepared for each meaning, and separated into different proportions in terms of the occurrences of one sense or the other of the word. A corpus was trained with each of these proportions of occurrences. For example, a corpus was trained in which "noray" occurs 0 times in phrases concerning demarcation and 30 times in phrases concerning traditional sport, then 5 times in phrases concerning demarcation and 25 times in phrases concerning traditional sport, and so on. The proportions are as follows: 30-0, 25-5, 15-15, 5-25 and 0-30.

Each of the 5 semantic spaces are trained with the Spanish Corpus LEXESP[36] (Sebastián et al., 2000) and the phrases for each of the corresponding proportions. In the case of LEXESP a "by hand" lemmatized version is used (plural forms are transformed into the singular and feminine forms are transformed into masculine, while all verbs are standardized in their infinitive form). We chose sentences as processing units. We deleted words that appear in less than seven documents. This ensures a minimal representation of the terms analyzed. Then we applied Entropy pre-processing. We finally obtained a matrix with 18,174 terms in 107,622 documents, to which we applied the SVD algorithm, reducing the three resulting matrices to 270 dimensions. For this purpose, we used Gallito ®, an LSA tool implemented in our research group developed in .Net ® (VB.NET, C#) integrated with Matlab ®. The mean of the cosines between each term in the resultant semantic space is 0.044, and the standard deviation is 0.07.

Once the spaces were trained, each made-up word condition was analyzed in two ways: (1) by recording the type of relationships they promote, by extracting the first n semantic neighbors[37] and analyzing the distribution of similarities and (2) studying the distribution of the vector based on the construct of Contextual Diversity.

[36] In a study of Duñabeitia, J.A., Avilés, A., Afonso, O., Scheepers, C. & Carreiras, M.(2009), semantic pair similarities of the vector-words from this space have displayed good correlation with their analogous translations to English using spaces from an LSA model by http://lsa.colorado.edu/ and from HAL (Hyperespace Analogue to Lenguage) hosted at http://hal.ucr.edu/ , and also with judgment of Spanish natives speakers.

[37] Semantic neighbors of a term are extracted comparing the vector of the term with each of the terms in the LSA semantic space.

In the first instance (1), in order to test the hypothesis that the made-up word in its ambiguous condition has lower cosines with the first n semantic neighbors but closer cosines with the next n neighbors, we extract the semantic neighbors of the made-up word. We use these neighbors to sketch out the distribution of ranks of its neighbors based on their similarity (cosines), in other words, the first n neighbors in order of similarity in each of the conditions (30-0, 25-5, 15-15, 5-25 and 0-30). The number of neighbors n to be extracted will be 4000, although in an attempt to make a more graphical representation of the phenomenon, we will begin by showing the similarities with the first 30 neighbors and calculating an average of these similarities.

In the second case (2), the saturation levels in each dimension of the vector for the made-up word are recorded in each of the conditions, and a diagram extracted to check whether the scores in these dimensions are extreme or close to the mean. For reasons of clarity, this analysis was only carried out in the three most significant conditions, 30-0, 15-15 and 0-30. In addition we analyzed the score for each condition on each dimension, with a rank of 1, 2 or 3 (1 being the biggest score, 2 the intermediate score and 3 the lowest score). Absolute values of the scores were taken. In this way we were able to objectively check whether as the hypothesis states, the made-up word in its ambiguous condition (15-15) has more intermediate values (2nd order) than the other two conditions. This would be one way of demonstrating that the representation of ambiguous words has a more uniform distribution than that of non-ambiguous words, that have more extreme distributions. This might be the cause of effects such as associative difficulty and LDT response times.

5.5.3) Results:

5.5.3.1) Average similarities of the first 30 neighbors

The average of the cosines (figure 2) in the equally probable polysemy condition (15-15) with their first 30 neighbors are lower than for the dominant polysemy (5-25 and 25-5) and monosemy conditions (0-30 and 30-0). In other words, the closest neighbors of the equally probable polysemy will be activated

less than those of the dominant polysemy and monosemy conditions. We even noted that some dominant polysemy items behave in a similar way to pure monosemy (25-5 and 30-0), something that has been observed empirically in previous studies (Piercey & Joordens, 2000). The key conclusion that can be drawn from this first analysis is that the penalization we propose above becomes evident as the average for the equally probable polysemy condition is significantly lower than for the pure monosemy conditions. But how far does the penalization of ambiguity go? How would the same types of words behave extracting a greater number of semantic neighbors? Would we reach a point, as the hypothesis suggests, where penalization would become an advantage above a certain threshold?

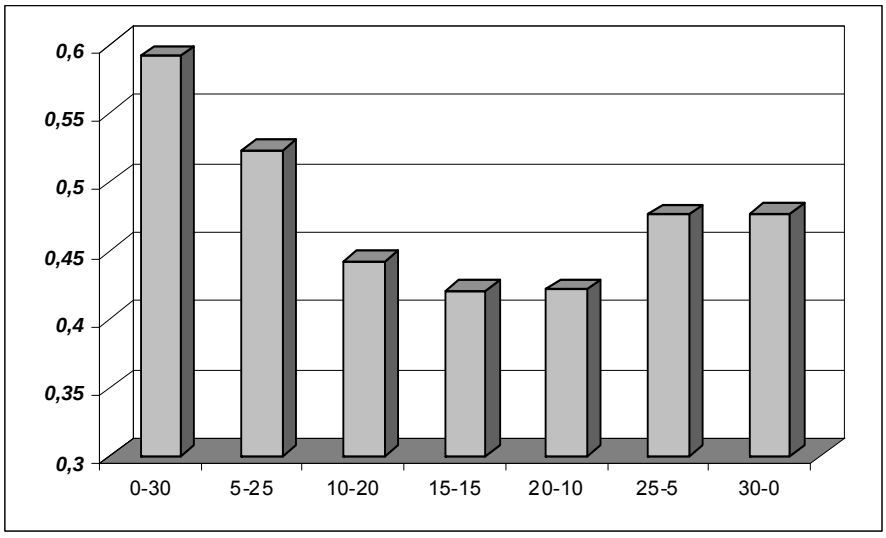


Figure 2. Average of cosines with the first 30 semantic neighbors of the seven conditions (0-30, 5-25, 15-15, 25-5 and 30-0)

5.5.3.2) Distribution of ranks for the first 4000 neighbors

The rank analysis on the first 4000 neighbors (figure 3) seems to suggest that the advantage is maintained until the neighbors go beyond a cosine of 0.2. Beyond this threshold the similarities begin to be less reliable (this threshold seems to coincide with the first 350 neighbors) - the advantage seems to dwindle and is even inverted in some of the conditions. One of the direct

conclusions is that the cited penalization operates up to a certain number of neighbors, and beyond this point the pattern seems to be inverted, and ambiguous terms are those that achieve greatest similarity with neighbors. On a theoretical level, we might then confirm the existence of a marginal activation corresponding to non-immediate neighbors. This activation is greater in ambiguous than in unambiguous words. Given the characteristics of the lexical decision explained above, this fact might be related to the lower response times in this task. As we can see, beyond neighbor 400 (at least up to 4000) the ambiguous words are the most closely related to their neighbors, possibly causing familiarity to be perceived with greater ease.

Now we have analyzed the distribution of neighbors we may have a vision of what penalization means and the possibilities, but what is the cause of this penalization? How might we verify the cause of this penalization statistically? What might be contained in the vectors of ambiguous words that promote such penalization? .

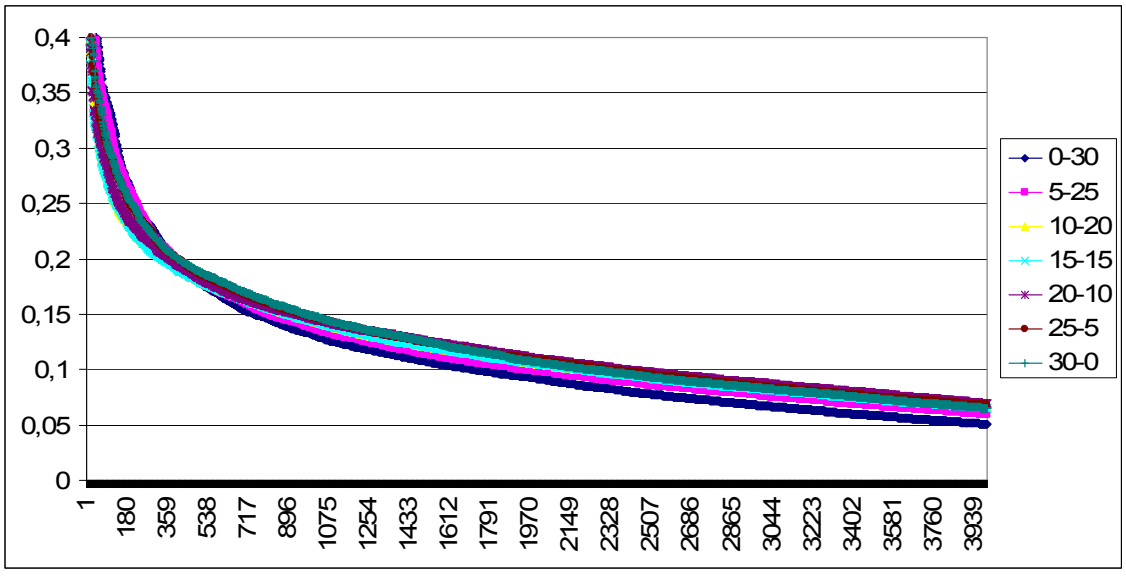


Figure 3. Distribution of ranks for the first 4000 neighbors in conditions 30-0, 25-5, 15-15, 5-25 and 0-30

5.5.3.3) Saturation of the dimensions

To check whether the distribution of the vectors may have played a role in the potential similarity with neighbors, we analyzed the vector of the artificial word in terms of the saturations in each dimension - in other words if the scores in the dimensions are extreme or close to the average. In the figure 4, we can see that the 0 – 30 condition has the most extreme coordinates in the latent semantic space, then the 30 – 0 condition and last the 15 – 15 condition, which has predominantly intermediate coordinates. This analysis supports the hypothesis that ambiguous words have the most differentiated and weakest representation, the opposite of non-ambiguous words whose definition is precisely located in certain dimensions, where it scores higher. This vector distribution might cause penalization in relationships with its closest neighbors, and a subsequent advantage to its relationships with more distant neighbors (as explained in figure 1), with these two phenomena causing the effects described: lack of semantic relationships for ambiguous words (in the absence of context) and generalized marginal activation. Still, how might we statistically demonstrate that the made-up word that operates as our ambiguous condition has more intermediate positions in each of the term vector's 270 dimensions?

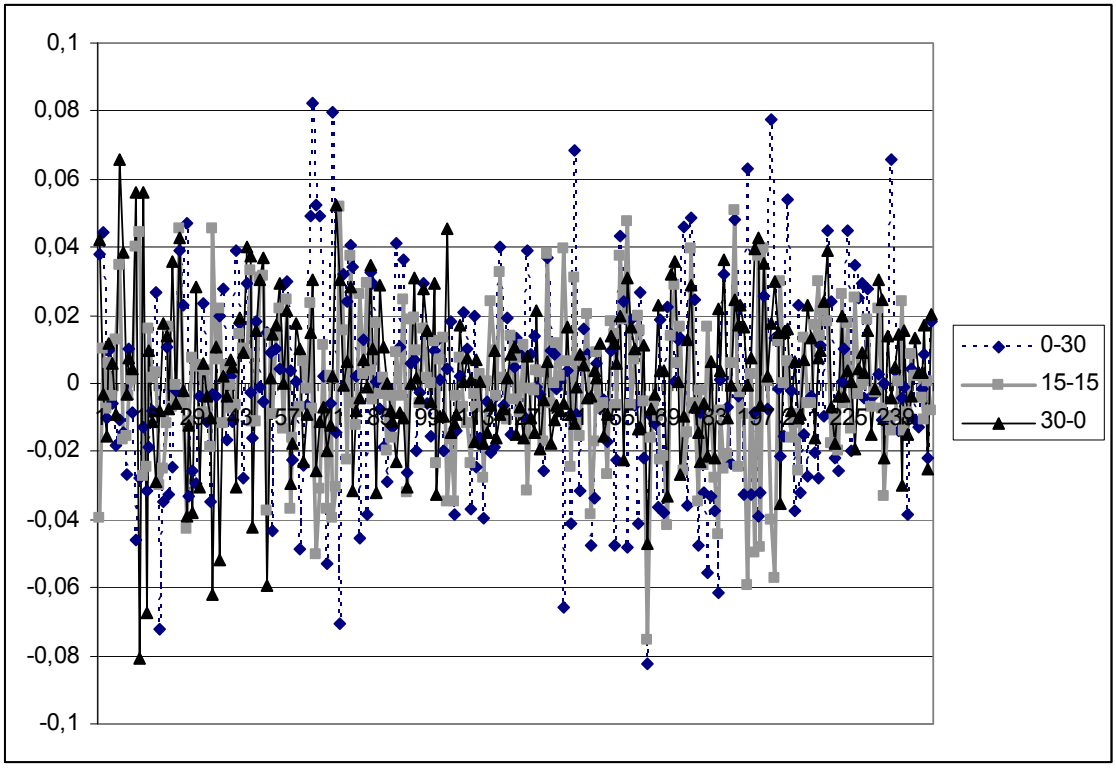


Figure 4. Coordinates of the dimensions of "noray" in three of its conditions.

5.5.3.4) Rank analysis

To objectively check that the aspect of the vector for each condition is significant we proceeded to analyze the score for each condition in each dimension, with a rank of 1, 2 or 3 (depending whether it scores on the dimension in first, second or third order).

As explained in the method section, each set of coordinates for each dimension is assigned a rank. In other words, in the same dimension a rank is assigned to each of the coordinates (with an absolute value) for each condition. 1 for the largest coordinate, 2 for the second largest coordinate and 3 for the smallest coordinate. The hypothesis is that the 15 – 15 condition should have more intermediate values (rank 2) than the 0 – 30 and 30 – 0 conditions, which will have more extreme ranks (more ranked 1 and 3). The results (figure 5) show that the monosemic conditions have more extreme distributions. In 52% of the dimensions the 0 – 30 condition has the higher coordinate. The 30 – 0

dimension had the highest coordinate in 32% of the dimensions (and the lowest coordinates, 41%), and the 15 – 15 condition in only 16% (and 50% of intermediates). If we analyze the hypothesis that the higher coordinate must be shared evenly between the three conditions (33% in each condition), the Chi-square test (table 1) rejects this hypothesis with a value of 47.88 ($p < 0.001$). The same can be said of the intermediate coordinate (Chi-square = 33.90, $p < 0.001$) and the smallest condition (Chi-square = 9.64, $p < 0.01$). The monosemic conditions (0-30 and 30-0) make the scores on the dimensions more extreme than the polysemic conditions (15-15). Polysemy attenuates the dimensions where the word might stand out by a smoothing caused by its multiple meanings. This clearly supports the aspect of the vector in figure 1, and formalizes the possible causes of the cited phenomena for ambiguous words: lack of semantic relationships (in the absence of context) and generalized marginal activation.

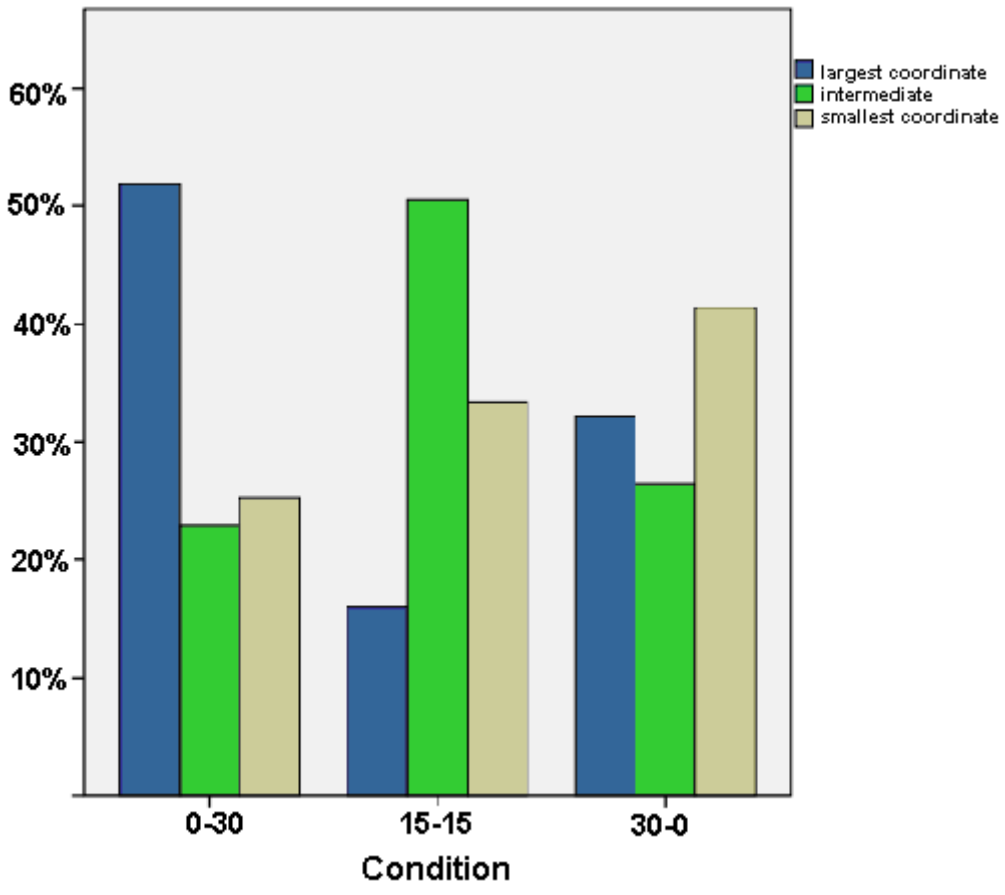


Figure 5. Rank analysis.

		Rank			
		Highest Score	Intermediate Score	Lowest Score	Total
Condition	0 - 30	51.8%	22.9%	25.3%	100.0%
	15 - 15	16.1%	50.6%	33.3%	100.0%
	30 - 0	32.1%	26.5%	41.4%	100.0%

Table 1. Rank percentages.

5.5.4) Discussion

This first simulation has revealed several things. Firstly, we have found that with a single representation model such as LSA, it is possible to simulate the empirical effects showing how ambiguous words have difficulty forming semantic relationships (synonymy, antonymy, hypo/hypernymy, meronymy) and is often related with another word by association (Warrington & Warrington, 2005; Crutch, 2006; Crutch, Ridha & Warrington, 2006). This also could be related with the delayed latencies found in some studies (Duffy et al, 1988). The mechanism is economical: the very structure of the vectors of ambiguous words promotes a penalization that impedes close relationships with their closest neighbors, meaning these relationships are weaker. This could be seen when we extracted the first 30 neighbors: the ambiguous condition obtained lower average similarity scores. We also checked this extracting the first 4000 neighbors, and observing the distribution of ranks. In this distribution we also observed the pattern that might offer the key to turn this penalization to our advantage for tasks such as the LDT. This paradoxical behavior of ambiguous words, a disadvantage that becomes an advantage, has already been detailed in some models that attempt to accommodate the LDT data, such as the "inefficient then efficient" model (Piercey & Joordens, 2000). The assumptions of this model are that the advantage of ambiguous words in lexical decision tasks is due to the simple non-specific activation that is produced before detailed semantic processing. In this simulation, the activation might be represented by the change of pattern in ambiguous words using their first n neighbors - in this case approximately the first 350 neighbors. Beyond this

threshold a superiority of the ambiguous condition (15-15) is recorded in terms of similarities with neighbors - in other words, beyond this threshold ambiguous words should have potentially a greater capacity for activating their more distant neighbors. Given that these more distant neighbors are far more numerous than the close neighbors, the ambiguous condition should have a certain advantage in tasks that demand generalized activation.

As for the causes of these effects, as we showed in Figure 1, we have confirmed the hypothesis that the 15 – 15 condition has more intermediate values (rank 2) than the 0 – 30 and 30 – 0 conditions, and this may be what provokes penalization in relationships and the marginal activation that favors LDT. In terms of the distribution of words across contexts (basically what vectors show), the effect of this type of variables was described in some results that show that Contextual Diversity of the words better explains LDT effects than the frequency (Adelman et al, 2006). The aspect of the vector for the ambiguous condition scores in the most dissipated and least intense way, and will penalize relationships with its first neighbors, while promoting similarity with the following. In the next simulation we introduce real words into the space and check whether we can confirm that the same pattern that might be covered by ambiguity is followed with linguistic categories. We will begin with the polysemic vs monosemic words.

6.5) Study 2

Given the results with the invented words, our hypothesis is that with two groups of real words, one monosemic and another polysemic, both effects will be checked: penalization and the same change of pattern using n neighbors. For this reason, we introduce real words into the space and analyze the distribution of its neighbors in the form of ranks. In addition, to objectify the fact that the distributions of both types of words are different, we calculate the entropy index (for a review see Nakov, 2001). This index shows that the difference between one type and another is a matter of Contextual Diversity

(Adelman et al, 2006), and this concept objectively identifies the number of contexts in which a word is represented.

6.5.1) Procedure

The procedure for extracting the sample of words to be studied was as follows: We take polysemic words from two sources (Estévez, 1991; Jorge-Botana, León, Olmos & Hassan-Montero, 2010). Using these two sources, the criteria to select the words for the polysemy group were: 1) that the words should have significantly more entries in the RAE (Spanish Royal Academy) dictionary than the monosemic words ($T(33)=5.65$, $p < 0.01$) and 2) that in Lexesp at least two meanings of each polysemic word were represented, checked using the authors' criteria with a visual sample of the first 100 semantic neighbors. In total 24 polysemic words. The group of monosemic words was extracted from the Lexesp corpus and are matched[38] with the polysemy group on indices of frequency of the corpus ($T(47)=0.45$, $p = 0.65$), relative frequency of the corpus ($T(47)=0.45$, $p = 0.65$), concretion value ($T(46)=1.70$, $p = 0.09$), concretion surveys ($T(46)=0,6$, $p = 0.57$), imaginability value ($T(46)=1.44$, $p = 0.15$) , imaginability surveys ($T(46)=0.03$, $p = 0.98$). The vectorial semantic space from which the neighbors are extracted will be the same as in simulation 1.

Two statistical analyses are carried out:

On the one hand, we check whether the distribution of similarities with neighbors might explain the hypothesis of associative difficulty and of non-specific activation - for this reason the semantic neighbors were extracted and the distribution of similarities was analyzed.

And on the other hand, the entropy index was calculated for each word in the raw occurrence matrix on the Lexesp corpus. This index indicates the amount of information that a word offers in relation to the contexts in which it appears. A greater entropy tends to shows a greater lack of focal points in terms of subject

[38] They were matched with the rates specified in the LEXESP corpus (Sebastián et al, 2000).

matter. If a word lacks subject area focalization, its occurrence is not enough to predict that we are speaking of a certain theme. The entropy is usually measured using the global weight formula (It use to be the product of a local term weight, l_{ij}, and a global term weight, g_{ij}). If a word appears in an extreme number of contexts its global weighting would be low (its entropy would be high). According to previous results, we would expect polysemic words, which appear in a greater number of contexts, to have greater entropy than the monosemic words, and as a result a lower global weight. With this prediction in mind, the global weighting of all previous words was calculated,

(1) Global Weigh formula

$g_i = 1 + \Sigma_j (p_{ij} \log(p_{ij}) / \log(n))$

where $p_{ij} = tf_{ij} / gf_i$

where

tf_{ij} is the number of occurrences of term ***i*** in document ***j***

gf_i is the total number of times term ***i*** occurs in all documents

n is the number of documents

6.5.2) Method

The method is practically the same as that carried out in simulation 1 to extract the distribution of the ranks of the first n neighbors. The only difference is that in the first study there was only one word per condition. In this second study there are 25 monosemic word and 24 polysemic words. The first 5000 neighbors of each of them was extracted and an average was taken in a single score. We finally obtained two distributions of ranks, one for polysemic words and another for monosemic words.

6.5.3) Results

6.5.3.1) Distribution of ranks for the first 5000 neighbors

The results show the same pattern as before (figure 6), in this case with real words, where the polysemic words have lower similarities with their closest neighbors. Beyond 600 neighbors (around cosine 0.2) this pattern changes, and they tend to have greater similarities with the other neighbors.

Figure 6. Distributions of ranks, one for polysemic words and another for monosemic words. The horizontal axe represents the first 5000 neighbors sorted by similarity. The Vertical one represents the cosines of such similarity. . For instance, the 1 rank is the closest term and corresponds to a cosine bigger than 0,6. The 1001 rank corresponds to a cosine lower than 0,2.

6.5.3.1) Entropy index

The results (figure 7) show that entropy is greater in polysemic words (they have a lower global weight) than in monosemic words ($T(47)=3.29$, $p<0.01$). This finding coincides with earlier results. According to these findings, polysemic words have a greater tendency towards dispersal of contextual themes.

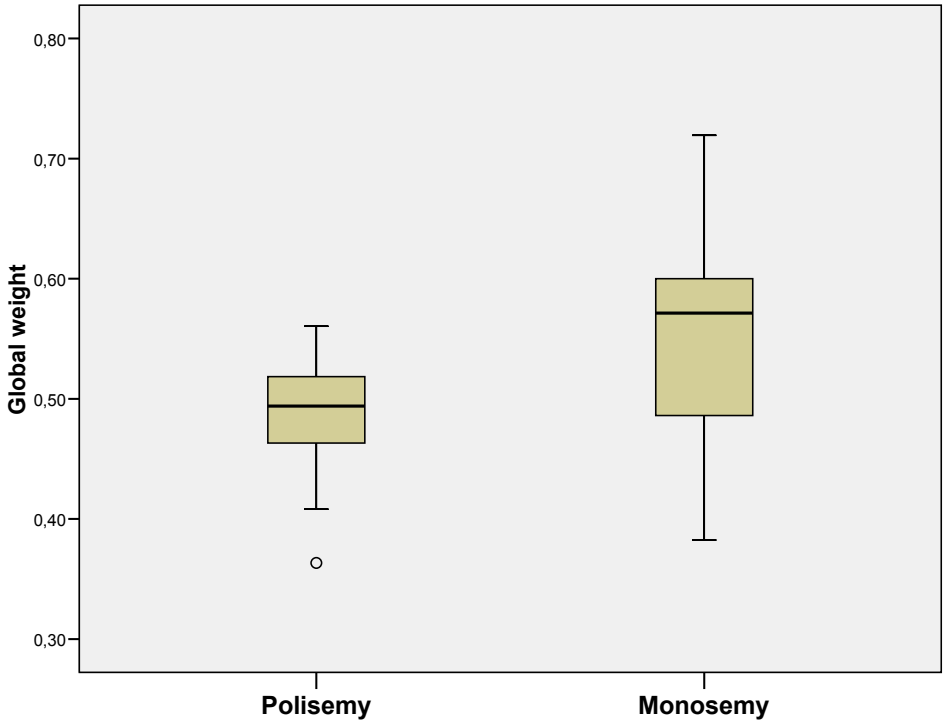


Figure 7. Global weight means for polysemic and monosemic words.

6.5.4) Discussion

In this second study we find the same pattern as with the artificial word in study 1. Firstly, we once again find the penalization in the relationship with its closest neighbors, again suggesting that it is possible to simulate the fact that ambiguous words (in the absence of linguistic context) are penalized when relating with the closest words. Secondly, we once again find the

inverted pattern beyond a certain threshold - in this case approximately 400 neighbors (also a cosine of 0.2). This seems to fit well with the empirical effects found in the LDT that show an advantage for ambiguous words, achieving better response times (Besner & Joordens, 1995; Joordens & Besner, 1994; Piercey & Joordens, 2000). The lexical decision is conceived as something that precedes full semantic analysis, and in the case of ambiguous words the marginal activation of memory patterns may be enough to break through the threshold that leads to an earlier decision than for unambiguous words. The greater similarities with more distant neighbors might mean that the ambiguous words achieve a higher activation level and break through the familiarity threshold earlier.

The fact that the monosemic words have less entropy (greater global weight) than the polysemic words confirms that the vectors of both types of words differ in the same way as in study 1 - in other words in terms of the distribution of scores across the contexts they appear in. Polysemic words occur in a greater number of contexts and represent the subject area of these contexts less clearly. As in study 1, this might be considered the primary cause of the penalization and the consequent change of pattern. The explanation is once again found in the arguments presented for Figure 1.

A theory that proposes a representation analogous to LSA (single, vectorized representation) might be useful and productive in predicting the resulting linguistic phenomena. One of them, for example, is the fact that some authors have suggested that deep dyslexia is caused by an inability to inhibit spurious activation rather than difficulty with phonological processing (Colangelo & Buchanan, 2004; Colangelo, Buchanam & Westbury, 2004). Since ambiguous words have more meanings and therefore more probability of spurious activation, they generate more semantic errors than unambiguous words. Colangelo & Buchanan (2004) argue that this lack of inhibition produces the same effect in deep dyslexics as tasks that require only implicit access to the lexicon in the normal population, such as LDT.

The existence of this marginal activation in our studies can help to objectify the way this is generated.

6.6) Study 3

Given the interest in verifying whether abstract words can be thought of as a variant of ambiguity and whether, as with polysemic words, we can simulate the phenomena described without the use of any representation other than a purely linguistic one (the properties of vectors in a purely linguistic representation such as LSA), we then proposed checking whether the same effect was produced in two groups of words (abstract and concrete) taken from a previous study *(*Duñabeitia et al., 2009). In this study it was suggested that the difference between these two groups of words was due to the fact that they were represented in different networks. In this third study we checked to see if a more prudent explanation might be available (those found in studies 1 and 2) or whether reference to connections with another type of representation is justified.

Similarly, to objectify the fact that the distributions of the two types of word are different, we calculate the entropy index. In this way we would have a index of what some authors call Contextual Diversity (Adelman et al., 2006) , which offers information on the contexts a term might appear in.

The fact is that if abstract words showed the same pattern (compared to concrete) as polysemic words (compared to monosemic) we might suggest two things:

1) The hypothesis of the contextual distribution of representations is a justified explanation of associative difficulty. We might propose the participation of purely linguistic representations for detailed semantic analysis.

2) Given that the two types of words show opposite behavior in LDT in the empirical data (superiority of concrete words), we could justify the

existence of connections with another type of representations that would inflate non-specific activation.

6.6.1) Procedure

The procedure is as follows: abstract and concrete words are taken from a previous study (Duñabeitia et al, 2009). In total 39 abstract words and 38 concrete words. The experimental control for words was taken from this same study, since the words are controlled for the following variables: frequency and grammatical category (all were nouns). The space from which the neighbors will be extracted is the same as in study 1, but without the made-up word. In this study only the semantic neighbors were extracted and the distribution of similarities was analyzed

6.6.2) Method

The method is similar to that carried out in study 2 to extract the distribution of ranks of the first n neighbors. The first 5000 neighbors are extracted and a list is made with the similarities (cosines) with each ranked neighbor. Finally the average of each rank is taken in each condition. For example, rank one has an average of 1, that is the similarity with the first neighbor (with itself), rank two has an average of 0.7, rank three has an average of 0.69, and so on until rank five thousand. The result will be a distribution like that of studies 1 and 2.

6.6.3) Results

6.6.3.1) Distribution of ranks for the first 5000 neighbors

We find the same pattern as the original simulation (study 1). Abstract words tend to have lower similarities with their n first neighbors (figure 8), with n surprisingly reaching the threshold where the cosine breaks through 0.2. As for the activation levels produced beyond this threshold, the opposite effect is produced, in other words there seems to be greater marginal activation in the abstract words (beyond 0.2).

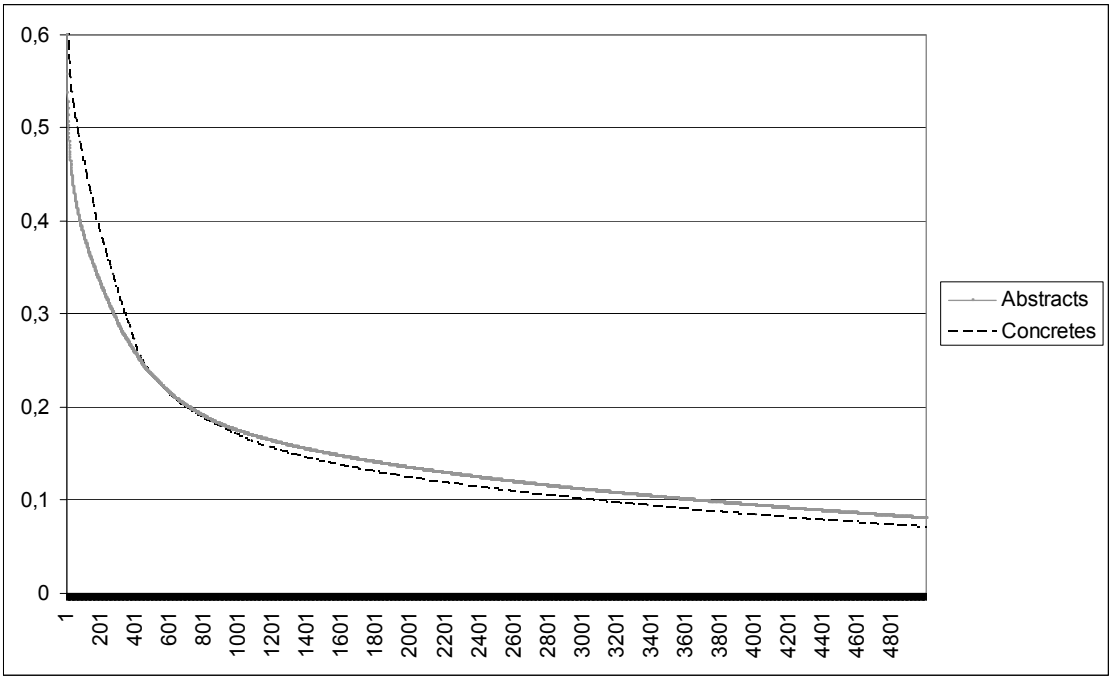


Figure 8. Distributions of ranks, one for abstract words and another for concrete words. The horizontal axe represents the first 5000 neighbors sorted by similarity. The Vertical one represents the cosines of such similarity. For instance, the 1 rank is the closest term and corresponds to a cosine bigger than 0,6. The 1001 rank corresponds to a cosine lower than 0,2.

6.6.3.1) Entropy index

The results (figure 9) show that entropy is greater in abstract words (have lower global weight) than in concrete words ($T(75)=2.42$, $p<0.05$). This finding coincides with the earlier results.

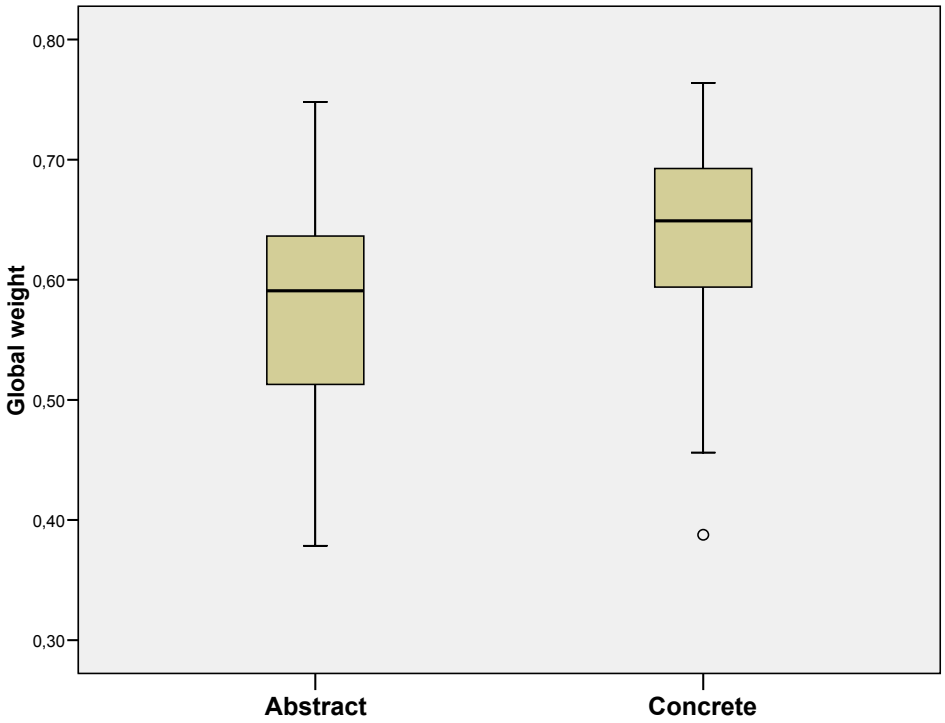


Figure 9. Global weight means for abstract and concrete words.

6.6.4) Discussion

The results show the same pattern as study 1 shows with polysemic vs. monosemic words (this time with abstract vs concrete). Firstly the penalization is repeated again in terms of the relationships with the closest neighbors. We once again simulate the fact that concrete words activate other words primarily by semantic similarity whilst the abstract words do so by association (e.g. Crutch, 2006; Crutch, Ridha, & Warrington, 2006; Crutch & Warrington, 2005; Warrington & Crutch, 2007). This may indicate that the way of relating to and activating the words that are semantically closest to concrete and abstract words can be justified by the contextual distribution of linguistic entries without reference to any other type of representation. In other words, that to generate meaning, the relationship with other linguistic symbols is more important than the basic perceptual fundamentals.

If this is the case, it is paradoxical to find the same inverted pattern as in the other studies - in other words, that abstract words have the potential to activate terms in the semantic space more (beyond the first 500 neighbors). Given that both types of words show opposing behavior in the LDT (superiority of concrete words in LDT), and that this task is favored by this type of non-specific activation, we may find that another type of representation participates in the lexical decision task, boosting the activation of concrete words (as the empirical data from the LDT shows). In other words, if the study shows that abstract words have the same pattern of Contextual Diversity as polysemic words, and also generate more marginal activation than concrete words, where does the activation necessary for concrete words to achieve higher activation than abstract words in the empirical data come from? Our data rules out the idea that the advantage of concrete words in the LDT is due to the contextual distribution of a purely linguistic system (the distribution would in any case give an advantage to abstract words). Therefore, to justify the advantage in the activation of concrete words, we should formulate our hypothesis with another type of representation. The most plausible hypothesis is that this activation derives from primary perceptual representations.

As for the second order isomorphism, the data from this study does not completely rule it out. What this theory proposes is that the meaning of a symbol is not defined by references to other levels of representation (perceptual representations) but rather by its relationship with other symbols. In other words, verbal information is largely a reflection of perceptual information, but this verbal information is enough for processing without the need for perceptual representaions. The marginal activation required for the LDT cannot be considered a process that generates meaning, and it may be that the true process of generating meaning is more similar to the extraction of closest neighbors than to this almost spurious activation capability. To summarize, in view of our data we could defend the hypothesis that whilst the connections taken from of perceptual areas take part in generating the activation that explains the advantage of concrete words in the LDT, this is not the case in tasks that require more detailed semantic analysis, where the penalization that produces the diversity of contexts comes into play. Some studies have found

that there are no differences between abstract and concrete words in terms of the areas activated in the brain, when the task requires detailed semantic analysis, such as the categorization task (Pexman et al. 2007). This type of findings, added to the fact that other studies have indeed found these differences for a lexical decision task (Binder, Westbury, McKiernan, Possing & Medler, 2005) may support our hypothesis.

6.7 Overall discussion:

Single representation

In this study we have proposed that some of the empirical data concerning lexical ambiguity may be explained simply using vector models where each lexical unit is represented by a single vector. The discussion on this point makes important assumptions, as lexical ambiguity is almost ubiquotous in human language and it is (almost) impossible to find an example free from ambiguity. One hypothesis is that the distribution of vector properties itself could explain the relationships that these might or not have with other terms, and the way of activating the environment - in other words the way that mentioning a term leads to the activation of specific content, and the form this takes. Thus, compared to the theories that propose different entries for each meaning of an ambiguous word, we have proposed that a model like LSA (a purely linguistic model) might account for some empirical phenomena without leaving aside the initial assumption: static entries for terms, with biases based on the moment of acquisition, and context-free (Kintsch, 2001). They are static in the sense that they emulate permanent knowledge, biased in the sense that the meanings of words are represented in vectors based on their appearance in real language usage, and context-free in the sense that the vectors do not refer to any context (or more precisely they're an amalgam of many) - the interaction with one of them is what dynamically generates a meaning (for a review see Jorge-Botana et al, 2010).

The empirical data that we have tackled with our LSA model can be put into two groups. On the one hand there is the fact that ambiguous words evoke terms differently from non-ambiguous words. On the other hand there is the fact that ambiguous words achieve lower response times in LDT. We have tried to justify this empirical data with LSA spaces that incorporate ambiguity in three different ways: 1) in a controlled way (using an artificial word, with controlled proportions), 2) checking real monosemic/polysemic words and 3) with real concrete/abstract words. The inclusion of these last two forms (concrete/abstract) as categories of ambiguity is justified because our hypothesis states that the concretion/abstraction continuum and polysemy/monosemy might be explained using the same general framework. Both have a common denominator, which can be defined by what some authors call Contextual Diversity (Adelman et al, 2006) - when a word is used in a greater variety of contexts, it will have greater contextual diversity. In the study by Adelman et al. on Contextual Diversity, they refer to both linguistic categories as modulating this index. Words with more than one meaning have more contextual diversity than monosemic words, and abstract words in turn have greater contextual diversity than concrete words.

Penalization of relationships

The first hypothesis we have tested is that the vectorial distribution of the terms in an LSA model was enough to account for the fact that ambiguous words evoke content differently from non-ambiguous words. This refers to the fact that some studies have found that ambiguous words' relationships are dictated by a context that favors one of the meanings (Duffy, Morris, & Rayner, 1988, Joordens & Besner, 1994), or that concrete words will activate other words primarily by semantic similarity whilst abstract words do so by association (e.g. Crutch, 2006; Crutch, Ridha & Warrington, 2006; Crutch & Warrington, 2005; Warrington & Crutch, 2007). To explain this it has been shown that the vector distribution of ambiguous words could penalize the similarity with other words. When ambiguous vectors are compared with the other vectors in the semantic space, the dimensions of a meaning always penalize similarities with the vectors of the other meaning. There is no

penalization for unambiguous words. To use the terminology of Adelman et al.(2006), the words with contextual diversity suffer this penalization.

The hypothesis that this penalization exists is ratified by the data from the first study with a made-up word, which show that the similarities of ambiguous words with their closest neighbors are lower than the similarities of unambiguous words with theirs. It is also ratified by the group of polysemic/monosemic words controlled for concretion, frequency and imaginability. The similarities between polysemic words and their closest neighbors are lower than the similarities for monosemic words. This in turn was ratified using a group of abstract/concrete words that were also controlled. The similarities between abstract words and their closest neighbors are lower than the similarities for concrete words. The hypothesis that this penalization is caused by contextual diversity itself - expressed in the distributions of the vector properties - is demonstrated by the fact that the polysemy conditions in the first study have scores more fairly distributed among all the dimensions of the vector. As for the real words in studies 2 and 3, polysemic words have more entropy than the monosemic words, and abstract words have more entropy than concrete.

To summarize, the data extracted from the studies shows that the penalization described as the cause of empirical findings is justified, and that it can be simulated with polysemic/monosemic words and with abstract and concrete words. The fact that it is simulated with both categories of words suggests that the processes that require similarity with the closest content, both in the polysemic/monosemic construct and in abstraction/concretion might be explained within the same framework: the contextual diversity of each word on a purely linguistic level (we should remember that LSA is a model without references to the real world). This last assertion fits with the assumptions of second order isomorphism (Kintsch, 2008).

Marginal activation

The second fact that we tested is that of the data from ambiguous words in the LDT. Some studies have found that in lexical decision tasks (LDT) ambiguous words require a shorter response time. According to some theories *(Besner & Joordens, 1995; Joordens & Besner, 1994; Piercey & Joordens, 2000)*, ambiguous words produce a mixed state in the semantic units - in other words an activation pattern that represents several of the meanings. To these authors the lexical decision (LDT) involves an evaluation of similarity, comparing the representation activated with localized patterns in memory, just as with the extraction of neighbors in our studies. They also assume that the lexical decision is previous to full semantic analysis, and that in the case of ambiguous words the marginal activation of memory patterns may be enough to break through the threshold that leads to a decision earlier than for unambiguous words. Other authors propose two steps in the processing of the words: first a non-specific activation is generated (where the LDT operates) and second the activation moves toward certain patterns of meanings. This leads us to the advantage of ambiguous words in LDT, contrasting with the disadvantage they seem to have in tasks where the mechanisms of this second stage are required, such as the relatedness decision (Pexman, Lupker & Hino, 2004). The separation of these two stages is analogous to that proposed by authors that have taken LSA as the basis for their simulations Kintsch (1998). In these models, a term first activates a certain number of lexical units (activation) which are later reactivated or differentially inhibited according to competing potential contexts.

In line with these proposals, our hypothesis has been that penalization of vectors of ambiguous words might also explain the advantage of ambiguous words in the lexical decision. Owing to the distribution of the vectors of ambiguous words, ambiguous vectors have more difficulty matching the patterns of its closest neighbors (associative difficulty) but fit easier with all the others, giving rise to non-specific activation higher than for unambiguous words. This non-specific activation may be enough to cross the decision

threshold to say that is a word - something that happens in the absence of detailed semantic analysis (Piercey & Joordens, 2000, pp 658).

In all the studies carried out we find that the penalization (which impedes close relationships with ambiguous words) operates up to a certain number of neighbors. Beyond this, the pattern seems to be inverted and ambiguous words achieve greater similarity with the remaining neighbors (most of them). This second wave of activation, much more numerous that the first, may allow ambiguous words to have this non-specific activation around them that might facilitate the strategies taken when tackling the LDT. As for polysemic words, the data are similar to the empirical data that refers to the LDT. The participants invest less time in the lexical decision task if they are presented with a polysemic word. This can be explained with the data from our studies, as the polysemic words seem to generate greater similarities with most of the semantic neighbors (less with the first). However, the empirical data from the abstract words, the advantage of the concrete words at the LDT, does not seem to agree with our data. In our study, the abstract words are those that show more activation potential over the words in the semantic space. For this reason, it is plausible to suggest that to justify the concrete advantage, a model that does not receive activation from another type of representation that is not linguistic (such as LSA) is not enough. Therefore another type of entries might be generating this activation. The most plausible hypothesis is that this activation derives from primary perceptual representations as proposed in the dual model (Paivio, 1986, 1991).

Conclusion

To summarize, in view of the results from all the studies, it may be justified that a single representation for each term might account for some phenomena surrounding lexical ambiguity. On the one hand, to the generate the meaning, the LSA vectors might watch what type of relationships a word might have with the other. This is of special interest in the case of the abstract words, as a model like LSA without reference to the perceptual

world, might also justify the type of relationships that the terms have. On the other hand, the data shows that to account for the effects surrounding LDT, the data from our studies might only account for empirical data in relationship with the polysemic words, but not of the abstract words. The vectors of a model such as LSA cannot account for the activation produced to justify the advantage of concrete words in the lexical decision task. Therefore, we might suggest that this necessary activation might come from primary perceptual representations. As for the Abstraction-Concretion construct, our data seems to suggest that the allusion to primary representations becomes unnecessary for generating meaning (where associative difficulty operates) and that a single representation and purely linguistic model might account for this process. However, it is necessary appeal to this type of representations when we speak of the lexical decision response. This paradox has already been described in some studies, which show that in the lexical decision task concrete words, and not abstract, activate areas related with perception (Binder, Westbury et al, 2005). However, when tasks that require more sophisticated semantic analysis are run, there is no difference in terms of the brain areas, but rather the activation of abstract words is broader (Pexman et al. 2007). This suggests that the need to activate the primary entries is dependent on the strategies carried out in each task.

Capítulo 11

Conclusiones

En el presente trabajo de tesis hemos dado un largo paseo por la técnica de LSA. En un plano teórico, hemos examinado la técnica y algunas de las formas en las que ha sido llevada a cabo por otros autores. En segundo término, hemos examinado LSA como modelo de la adquisición y la representación del lenguaje, además de reflexionar en torno a como puede LSA simular algunos procesos cognitivos. También hemos presentado la herramienta que hemos desarrollado y con la que hemos trabajado. Y por último, hemos presentado los cuatro experimentos independientes que hemos llevado a cabo para responder a algunas de las preguntas que nos hemos hecho. Hemos agrupado las conclusiones en unos pocos epígrafes:

11.1.- Sobre las experiencias previas de LSA como modelo de la representación adquisición y procesamiento léxico

La primera conclusión que queremos resaltar es que LSA es una técnica ampliamente difundida en el ámbito de la ciencia cognitiva con la que se han llevado a cabo tanto modelos de representación, adquisición y procesamiento del lenguaje. Para ello, se han entrenado grandes corpus y con ellos se ha podido simular algunas facetas de la representación y adquisición del conocimiento. Con todo, el LSA no está exento de limitaciones ni de críticas ni es una herramienta infalible que simule siempre fehacientemente el comportamiento humano, ni por supuesto muchos expertos están de acuerdo en que sea una teoría sobre adquisición de conocimiento. Aún incluso teniendo en cuenta esto, entre nuestras conclusiones podemos resaltar que LSA simula relativamente bien los fenómenos más importantes. Uno de ellos es por ejemplo la tarea de identificación de sinónimos. El LSA se ha probado aplicando exámenes de vocabulario. Concretamente, sobre un espacio semántico de LSA se aplicó el Test of English as a Foreign Language (TOEFL) (Landauer y Dumais, 1997). El TOEFL consiste normalmente en decidir cuál de cuatro palabras es más parecida a una palabra enunciado. La tasa de aciertos fue del 65%, coincidiendo con la tasa media de respuestas de los estudiantes que se enfrentan con el TOEFL.

El LSA también parece simular de una manera eficiente el hecho de que para la adquisición del conocimiento sobre las palabras no es preciso estar expuesto de una manera lineal a cada una de ellas (Landauer y Dumais, 1997). Por ejemplo, un estudiante típico estadounidense de séptimo grado (12-13 años), aumenta de media su léxico alrededor de 10 a 15 palabras cada día. La media de lectura diaria de estos estudiantes viene a ser de unos 50 párrafos, donde únicamente hay, por regla general, una media de tres palabras que desconocen. Aparentemente, los estudiantes adquieren conocimientos sobre el sentido de más palabras que las que encuentran entre los párrafos que leen ese día. Landauer habla de que una arquitectura como LSA ofrece un buen modelo para entender como las microrelaciones de unas palabras con otras pueden ser suficientes para captar significado de una palabra de la cual no tenemos mucha experiencia. Un niño podría leer la palabra *automóvil* un día. Esa palabra no la conoce pero sí conoce la mayor parte de las palabras que aparecen junto a ella. Supongamos que lee tres veces la palabra y sigue sin comprender su significado. Otro día, este mismo chico podría leer un párrafo en el que aparece la palabra *coche* pero no *automóvil*. De pronto, podría emerger el significado de *automóvil.* El cerebro habría captado que las palabras *automóvil* y *coche* son muy similares, aparecen en contextos muy parecidos (las palabras que las rodean son muy similares) luego se trata de palabras muy afines semánticamente. Este mecanismo de generalización por similitud contextual es el que explica, según estos autores, la adquisición del vocabulario sin necesidad de que los chicos tengan una exposición lineal a las palabras. Es decir, la adquisición de conocimiento sobre una palabra no se describe con una función lineal en base a la aparición de esa palabra en los textos que lee. La arquitectura LSA es capaza de hacer esto porque explota la información que emana de las concurrencias de las palabras, y no sólo de las ocurrencias de dos palabras en un mismo texto (relaciones de primer orden) sino que puede hacerlo también de palabras que no ocurren nunca juntas pero tienen palabras comunes que ocurren con ellas (relaciones de n órdenes). Aún así, no es menos cierto que los modelos LSA tienen un sesgo a sobrevalorar las relaciones en que las palabras ocurren en un mismo texto (relaciones de primer orden) frente a los demás órdenes. Algunos autores han realizado simulaciones con las concurrencias controladas de las palabras y han apuntado

que aunque las relaciones de segundo orden suelen influir en la función de similitud de las palabras, esta suele sobreponderar las relaciones de primer orden (Denhiére et al., 2007).

También es resaltable que existen algunas limitaciones de LSA en cuanto a la captación del sentido de las polisemias y homonímias. En los espacios LSA las palabras están representadas en un único punto dentro del espacio *n-dimensional* y esta representación no es más que un promedio de los significados dentro de los *n* documentos del corpus. Si una palabra tiene tres significados alejados, al amalgamarlos en una única representación y situarla en un solo punto del espacio podría quedar en *tierra de nadie* y no querer decir ni una cosa ni otra. Este tipo de limitaciones han tenido tentativas de resolución en algunos modelos en los que se gestiona eficientemente el contexto de recuperación de las palabras. Por ejemplo, para Kintsch (2001, 2007) la respuesta a esta cuestión depende de cómo hacer emerger el contexto en el que surge la palabra. Considerando el contexto de la palabra la ambigüedad desaparece. Tomemos como ejemplo la palabra homónima *banco* y sus acepciones. Si extraemos los vecinos semánticos de *banco* en relación al contexto donde aparece y surgen *sucursal* y *dinero* entonces ya no hay dudas de a qué se refiere a una entidad financiera. El significado de la palabra *banco* quedaría representada como la suma vectorial de su propio vector y los vecinos semánticos más parejos con el contexto (esta es la lógica del su algoritmo de predicación, Kintsch, 2001). Lo importante de esto es que una única representación que es una simple amalgama de significados, puede ser un buen sustrato si se combina con algoritmos que gestionen el contexto de recuperación. Estos mismos mecanismos han servido para extraer el significado de las metáforas predicativas (Kintsch, 2000). El sentido metafórico de una palabra puede ser extraído con los mismos mecanismos que el sentido de una polisemia. La diferencia estribaría en que sería necesario extraer listados de vecinos más profundos para captar los sentidos comunes entre las palabras y sus contextos.

En suma, el reto con el que se enfrenta el LSA es que por sí solo no hace emerger el significado adecuado de la palabra. Le hace falta un sistema que

gestione el entorno lingüístico en el que las palabras ocurren en el momento de darles significado. LSA puede ser un sistema adecuado para almacenar las representaciones de las palabras, es decir, en términos de memoria semántica. Por eso, algunos autores utilizan algoritmos adicionales que sirvan de gestores de dicho contexto para alcanzar significados más precisos del lenguaje (Kintsch , 2007; Lemaire, et al, 2007). En esta última limitación se ha centrado buena parte de este proyecto de esta tesis.

11.2.- Sobre las experiencias previas de LSA como evaluador de resúmenes

El LSA se ha utilizado con éxito también para medir el nivel de conocimiento que los estudiantes tienen sobre un tema determinado (Wolfe, 1998). La calidad de los resúmenes se suele establecer de varias formas y la más habitual es comparando el resumen del estudiante con el texto instruccional y en función del coseno evaluar la calidad del resumen. En general se han encontrado buenas correlaciones entre LSA y los jueces humanos a la hora de evaluar textos. Esto ha hecho que se hayan implementado sistemas para ser empleados en el ámbito académico. Autores como Kintsch, Steinhart, Stahl, y LSA Research Group (2000) han utilizado LSA como sistema tutor computerizado para evaluar la calidad de los resúmenes en estudiantes de Primaria dentro de programa instruccional automatizado que permitiese a estudiantes aprender a cómo resumir un texto. En suma, existe un interés creciente en cuanto a la evaluación automática de resúmenes académicos y los datos muestran que algunos sistemas se han comportado de manera eficiente (Para una compilación ver Trusso, 2005). El único problema es que la variabilidad en que la técnica se ha aplicado ha hecho que no se hayan esclarecido las condiciones en las que los motores LSA funcionan de una manera efectiva y aquellas condiciones superfluas o incluso contraproducentes. El siguiente punto presentará este problema.

11.3.- Sobre la evaluación de respuestas en corpus pequeños

LSA es una herramienta eficiente para la evaluación de respuestas y así lo ha demostrado en multitud de investigaciones (Para una compilación ver Trusso, 2005). Aún así, en nuestro grupo de investigación partíamos de la hipótesis de que en muchos de los estudios se había dado por hecho sin justificación ninguna que la reducción de la dimensionalidad era beneficiosa, aunque los corpus fueran de un tamaño muy restringido. Es más, el hábito más repetido en estudios precedentes ha sido reducir las matrices en muy pocas dimensiones, arguyendo que con una temática restringida y específica de un dominio, era más apto reducir a menos dimensiones el espacio. Aún así, teníamos también la convicción de que los espacios extraídos de corpus pequeños pueden suponer una buena solución para diseñar tutores virtuales en el aula, dado su bajo coste y su manejabilidad. Con esa motivación empezamos a hacer una gran batida de las dimensionalidades que fueron empleadas en otros trabajos concluyendo que la reducción *per se* de la dimensionalidad no implica necesariamente una garantía de su efectividad (Jorge-Botana et al, 2010). A este efecto, realizamos un experimento en el que se ponían a prueba diversas dimensionalidades, además de otros parámetros, como la información tangencial de los corpus, la forma de construir pseudocumentos, las medidas de similitud entre los textos y las fórmulas de ponderación de los términos. Todos estos parámetros se pusieron a prueba con dos tipos de respuestas, aquellas redactadas por expertos en la materia y aquellas redactadas por legos. Nuestros resultados muestran algunos patrones: uno de ellos es que no siempre la reducción de dimensionalidad es más beneficiosa que no llevarla a cabo y sólo en algunas combinaciones parece funcionar mejor, como por ejemplo, en las condiciones en las que se emplea Folding-In para formar los pseudodocumentos y distancias euclídeas como medidas de similitud. Además, dicha reducción es más beneficiosa para los corpus con información tangencial y con respuestas redactadas por expertos. Otro de los resultados recomienda emplear la distancias euclídeas como medida de similitud entre las respuestas de los alumnos y las repuestas ideales que el propio sistema posee. El hecho de utilizar las distancias euclídeas en lugar de los cosenos es debido a que estas últimas evitan un

efecto nocivo en las respuestas cortas y de poco contenido. En este tipo de respuestas, la mera aparición de una palabra clave puede producir que el vector que representa esa respuesta este muy cerca vectorialmente de la respuesta ideal. Sin embargo, cualquier evaluador humano puntuaría tales respuestas con una puntuación muy baja. Por ejemplo, si instamos a los alumnos a redactar un texto que describa la fobia específica y una de las respuestas es “la fobia específica es un miedo a cosas”, el coseno entre el vector que represente la respuesta y el vector de la respuesta ideal puede ser alto. Nótese incluso que dos de las palabras es una simple paráfrasis de la pregunta (fobia específica). Sin embargo, las distancias euclídeas tienen en cuenta que el desarrollo de las respuestas sea suficiente, además de que sea similar al texto ideal redactado por un experto.

La conclusión final de esta primera investigación es que cuando se usan corpus de pequeño tamaño tenemos que llevar un cierto agnosticismo de partida en cuanto al funcionamiento de la reducción de la dimensionalidad y otros parámetros. Lo recomendable sería programar algunas pruebas preliminares de forma heurística que pueda arrojar la mejor dimensionalidad. Por ejemplo, se pueden probar los criterios propuestos por Wild et al.(2005), el cual recomienda que la matriz reducida comparta del 50% al 30% de dimensionalidad con la matriz original. A la luz de nuestros resultados (Jorge-Botana, León, Olmos, Escudero, 2010), creemos que el criterio de Wild es eficiente. Además, siempre es beneficioso ponderar los términos en la matriz de ocurrencias con la fórmula de Log-Entropía, lo que ayuda a optimizar el efecto de la dimensionalidad y a evitar la sobrevaloración de ciertas palabras vacías. Otro de los aspectos recomendados es el uso de las distancias euclídeas en vez de los cosenos, cuestión esta que se hace casi obligada y que ha sido encontrada en otro de nuestros trabajos (Olmos et al., 2009).

11.4.- Sobre el sesgo de representación de los términos

Además del campo académico, LSA ha sido empleado para extraer el sentido a las palabras en base a listados de vecinos semántico. Este tipo de listados se pueden extraer las palabras que llevaría una posible definición de la palabra de referencia. Una de las sensaciones que teníamos cuando extraíamos vecinos semánticos a una determinada palabra en nuestras pruebas preliminaras es que los vecinos suelen estar muy restringidos a la propia aparición de esa palabra. Por ejemplo, al sacar vecinos a "real", extraemos "Madrid". Como no pasa desapercibido, Real_Madrid puede ser casi considerado como un bigrama. La extracción de vecinos por medio del coseno suele promover este tipo de sesgos. A partir de aquí nos planteamos algún tipo de corrección para evitar este efecto indeseable (o por lo menos complementar las extracción que veníamos haciendo). Una de las formas que pusimos a prueba para corregir este tipo de extracciones es incorporar la longitud de vector (Jorge-Botana et al, 2009), una medida mucho más rica que la frecuencia (Rehder et al; 1998), como factor corrector a la medida de similitud (coseno). En el segundo experimento incorporamos dicha medida y obtuvimos muy buenos resultados. Este tipo de correcciones puede servir para dar a LSA de una forma aproximada una virtud que no tiene y que si tienen las ontologías: la capacidad de extraer vecinos que pertenezcan a niveles superiores de la definición de una palabra o lo que puede llamarse definiciones hipernímicas. Cuando en el segundo experimento pretendíamos sacar los vecinos a "tormentas" (recordemos que es un espacio semántico referido a la psicopatología), obteníamos vecinos en un nivel inferior o igual a esta palabra, por ejemplo, otros objetos motivo de fobias específicas, sangre, perros, volar, etc., pero no obteníamos una definición que incorporase vecinos de categorías superiores. Incorporando la longitud de vector, conseguimos extraer también este tipo de definiciones más acordes como subtipo, miedo, temor, específica, etc.

La consecuencia de esto es que se puede actuar sobre LSA obligando a que las medidas de similitud no solamente promuevan relaciones locales sino que también incorporen otro tipo de relaciones. Se trata de hacer que la

longitud de vector o cualquier otra medida de representatividad participen como factor en la función de extracción de vecinos y de esa manera conseguir que la definición en forma de lista contenga también elementos lo suficientemente representativos. Creemos que este tipo de correcciones serán de suma importancia, no solamente para la aplicabilidad en la industria sino porque los modelos cognitivos que se extraigan de LSA tendrán que incorporar los sesgos provenientes de la frecuencia de los términos o de la diversidad contextual en los que aparecen (Adelman, Brown & Quesada, 2006).

11.5 Sobre la generación dinámica del significado

Otra de las conclusiones interesantes del presente trabajo de tesis es en torno a la extracción de los sentidos de las palabras, sean estas homónimas (banco-parque, banco-silla), polisémicas (rama-árbol, rama-laboral) o palabras técnicas que toman sentidos diferentes según sea su acompañamiento (como fobia específica o fobia social). Todas estas palabras tienen una cosa en común y es que definen su sentido en base a su acompañamiento. La forma de afinar ese sentido es una de las cuestiones para las que LSA supone una buena base. LSA no distingue entre los diferentes sentidos de las palabras y sus significados. El modo en que LSA representa la polisemia es con un vector único en el que las palabras están representadas en un único punto dentro del espacio *n-dimensional* y esta representación no es más que un promedio de los significados dentro de los *n* documentos del corpus. Si una palabra tiene tres significados alejados, al promediarlos y situarla en un solo punto del espacio podría quedar en *tierra de nadie* y no querer decir ni una cosa ni otra o podría quedar más cercano al sentido predominante si es el caso de una polisemia desequilibrada. En resumen, en el mundo real una palabra es polisémica pero en estos modelos su representación está libre de contexto, simplemente es una amalgama de sentidos ponderados por sus ocurrencias en el corpus. El significado de cada representación cobra sentido solamente en el momento en que un contexto potencia ciertas dimensiones y hace emerger ciertas propiedades. Es en ese momento cuando se genera el significado, haciéndose de una manera dinámica y evanescente (Kintsch, 2001; Kintsch, 2007).

En el segundo y el tercer experimento (Jorge-Botana et al, 2009; jorge-botana, León, Olmos y Hassan-Montero, 2010) hemos podido comprobar las ventajas de un algoritmo como el de predicación (Kintsch, 2001) para generar significados a palabras con varios sentidos. En uno de ellos, se han extraído los sentidos a palabras insertas en un corpus diagnóstico. Por ejemplo, la palabra "fobia" cobra un sentido de fobia específica u otro según vaya acompañada por la palabra "tormentas" o de fobia social si va acompañado de "público. Fobia tiene en este caso un sentido polisémico, amplificado por el hecho de que se opere sobre un corpus específico de esta temática. En el otro experimento, empleamos un corpus de carácter general como el Lexesp (Sebastián, Cuetos, Carreiras & Martí, 2000) para ahora sí, simular la comprensión de polisemias en el lenguaje coloquial. Ambas simulaciones, la del corpus restringido y la del corpus general dan cuenta de cómo se puede producir la generación de significados y de cómo pueden ser las redes que dan sentido a esto, unas redes en las que algunos nodos son inhibidos y otros activados, generando un significado dinámico. Este algoritmo, el de predicación (Kintsch, 2001), sesga los vectores únicos de la representación que LSA tiene de una palabra, en base al contexto en el que se recupera un de sus sentidos. El procedimiento es activar en primer lugar todos los vecinos semánticos de dicha palabra y en segundo lugar, sobre-activar o inhibir dichos vecinos en base a la representación vectorial del contexto. Esta forma de activación ciega en el primer estadio y de selección en el segundo ha sido sugerido también por algunos modelos de acceso al léxico (Piercey y Joordens, 2000) en los que en una primera etapa es una simple activación previa al análisis semántico fino. Es en la segunda etapa dónde se lleva a cabo la selección en base a la inhibición/activación de los nodos activados.

En suma, nuestra opinión está acorde con la opinión de Landauer (1999) al decir que LSA ofrece un modelo computacional que simula correctamente muchos fenómenos que tienen que ver con el uso del lenguaje y su adquisición. Pero pensamos también que el problema con el que se enfrenta el LSA es que por sí solo no hace emerger el significado adecuado de las

palabras. Le hace falta un sistema que gestione el contexto de recuperación de manera eficiente y lo integre con la representación estática que proporciona el espacio semántico. Autores como Kintsch (2007) o Lemaire, et al Larose (2007) que consideran al LSA como un sistema adecuado para almacenar las representaciones de las palabras, es decir, en términos de memoria semántica. Por eso, estos autores utilizan algoritmos adicionales que sirvan de gestores de dicha memoria para alcanzar significados más precisos del lenguaje. Esta última limitación es la que hemos querido hacer explícita en este trabajo de tesis y los resultados son esperanzadores: Esta limitación se convierte en una ventaja si se provee a LSA de una buena gestión del contexto de recuperación. El conocimiento sobre los procesos descritos en psicolingüística puede ayudar a conseguirlo.

11.6.- Sobre LSA como modelo para algunos fenómenos empíricos en torno a la ambigüedad léxica

Cómo ya ha sido tratado con anterioridad, LSA representa la polisemia por medio de un vector único en el que las palabras están representadas en un único punto dentro del espacio *n-dimensional. Dado* que LSA ha mostrado buenos resultados en representar los términos del léxico de las personas (Landauer y Dumais, 1997), este tipo de representación tiene consecuencias para la forma de concebir las representaciones en la mente humana.

Dada esta plausibilidad, un modelo como LSA puede ayudar a dar una definición de la señal lingüística que es procesada por el sistema humano. Por ejemplo, permite objetivar algunos parámetros para definir la ambigüedad de una manera objetiva. Sabido es que aún existe una laguna en dar una definición extensional, completa y objetiva de la ambigüedad léxica (Charles Lin, Ahrens, 2009). Una vez definida la señal, podemos correlacionar los parámetros establecidos en ella con la respuesta a tareas de las que tenemos datos empíricos. De esta manera, podemos predecir que ciertos parámetros provocan cierta fenomenología empírica. Además, como método de descarte, podemos identificar las tareas en las que un modelo de representación única y exclusivamente lingüístico no puede explicar. Por ejemplo, definiendo los

parámetros que pueden definir la ambigüedad en un vector LSA, podemos correlacionar dichos parámetros con la respuesta empírica de los humanos ante esas palabras en una determinada tarea. Si la correlación es significativa, podemos decir que la representación de las palabras en la mente puede ser parecida a LSA. Por el contrario, si no hay correlación, podemos decir que un modelo LSA no es del todo suficiente para explicar toda la casuística y que por tanto, las representaciones de la mente, o no son análogas o necesitan de otro tipo de activaciones que LSA no contempla.

En la última de nuestras simulaciones nos propusimos comprobar lo que LSA podía decir sobre algunos fenómenos que han sido estudiados empíricamente en torno a la ambigüedad. El primer efecto a estudiar fue la dificultad que muestran las palabras ambiguas de generar significado en ausencia de contexto (Brown y Ure, 1969; Duffy, Morris y Rayner, 1988). Respecto a esto, hemos postulado que la propia distribución de los vectores de LSA podría explicar dicho efecto, es decir, el vector de una palabra polisémica, dada su menor focalización o su mayor diversidad contextual (Adelman et al, 2006), tiene una penalización en la función de similitud con sus principales vecinos semánticos. El segundo efecto es la ventaja que muestran las palabras polisémicas en la tarea de decisión léxica. Tradicionalmente se ha postulado que dicha tarea, la de Decisión Léxica, es beneficiada por la activación inespecífica que generan los términos por medio de sus representaciones semánticas. El mecanismo es el siguiente: la activación de la representación ortográfica de un términos activa también sus representaciones semánticas y por medio de éstas, de nuevo se activan representaciones ortográficas. El hecho de que las palabras polisémicas posean mayor número de significados, hace que se active mayor número de representaciones semánticas y por ende, mayor activación ortográfica. Esta última activación hace que el umbral necesario para decir que una palabra es una palabra se rebase en menor tiempo. En nuestra simulación, hemos postulado que si debido a sus propiedades distribucionales, las palabras no-ambiguas poseen potencialmente más capacidad para relacionarse con sus vecinos semánticos principales, debido también a estas propiedades, las palabras ambiguas tendrán potencialmente más capacidad para activar a resto de términos (la gran

mayoría) de vecinos. Esta activación de los términos no principales puede ser suficiente para explicar la activación inespecífica requerida en la Tarea de Decisión Léxica.

A su vez, hemos querido poner a prueba esta hipótesis también para las palabras concretas y abstractas. Hemos propuesto que la desventaja de las palabras abstractas para generar significado y el hecho de que mantengan relaciones asociativas en lugar de semánticas, es debido a las propiedades distribucionales de los vectores (al igual que las palabras ambiguas). Algunos estudios han observado que las palabras abstractas tienen más diversidad contextual (Adelman et al, 2006), lo que indicaría que sus vectores tiene una menor focalización o lo que es lo mismo, mayor diversidad contextual.

Por el contrario, dado que en los datos con humanos existe una ventaja de las palabras concretas en la Tarea de Decisión Léxica, si es el caso de que en LSA, los vectores de las palabras abstractas, excluyendo los principales vecinos, activan mayor número de términos, se podrá postular que en la tarea de decisión léxica, las palabras concretas están siendo activadas por otra fuente de activación. De esta manera, un modelo exclusivamente lingüístico como LSA no podría dar cuenta de la ventaja de las palabras concretas en la Tarea de Decisión Léxica.

Nuestros resultados (Jorge-Botana, Olmos y León, Submitted) han mostrado que los vectores de las palabras polisémicas poseen potencialmente la dificultad de asociación con sus primeros vecinos semánticos, lo que puede dar cuenta de la dificultad de las palabras ambiguas de generar significado en ausencia de contexto. Además, como predijimos, esta penalización se convierte en ventaja para activar al resto de vecinos, lo que puede generar la activación necesaria para explicar la ventaja de las palabras ambiguas en la Tarea de Decisión Léxica. Respecto a las palabras abstractas, los resultados muestran también que los vectores de las palabras abstractas poseen potencialmente la dificultad de asociación con sus primeros vecinos semánticos y que a su vez estos mismos vectores, a partir de un determinado rango de vecinos, empiezan a tener una ventaja para relacionarse con el resto de

términos. Este hecho muestra que para las palabras abstractas, un modelo como LSA puede dar cuenta del efecto de sufren en cuanto a sus relaciones con otros términos (la dificultad de asociación) pero no puede dar cuenta de la ventaja que las palabras concretas tienen el la Tarea de Decisión Léxica, dado que en el modelo LSA son las palabras abstractas las que generarían la activación que les concedería la ventaja en la Tarea de Decisión Léxica.

En resumen, una representación única para cada término puede dar cuenta de algunos fenómenos en torno a la ambigüedad léxica. Por un lado, para la generación del significado, los vectores de LSA pueden dar cuenta de que tipo de relaciones puede mantener una palabra con las demás. Esto es de especial interés en el caso de las palabras abstractas pues un modelo como LSA, sin ninguna referencia al mundo perceptivo, puede justificar también el tipo de relaciones que mantienen los términos. Por otro lado, los datos muestran que para dar cuenta de los efectos en torno a la LDT, los datos de estas simulaciones pueden sólo dar cuenta de datos empíricos en relación a las polisemias pero no de las palabras abstractas. Los vectores de un modelo como LSA no pueden dar cuenta de la activación que se produce para justificar la ventaja de las palabras concretas en la tarea de Decisión Léxica. Por tanto, se puede sugerir que esa activación necesaria puede provenir de representaciones perceptivo-primarias. Respecto al constructo Abstracción-Concreción, nuestros datos parecen sugerir que la alusión a representaciones primarias no se hace necesaria para la generación de significado (dónde opera la dificultad de asociación) y que un modelo de representación única y eminentemente lingüística puede dar cuenta de este proceso. Sin embargo, si que es necesario apelar a este tipo de representaciones cuando hablamos de la respuesta a la Decisión Léxica. Esta paradoja ha sido ya descrita en algunos trabajos que muestran que en la tarea de Decisión Léxica, las palabras concretas, a diferencia de las abstractas, activan áreas relacionadas con la percepción (Binder, Westbury, et al, 2005). Sin embargo, cuando se ponen a prueba tareas que requieren análisis semántico más sofisticado, no hay diferencia en cuanto a las zonas cerebrales, sino que la activación de las abstractas es más amplia (Pexman y Lupker. 2007). Esto sugiere que la

necesidad de activación de las representaciones primarias es dependiente de las estrategias llevadas a cabo en cada tarea.

REFERENCIAS BIBLIOGRAFICAS

REFERENCIAS BIBLIOGRÁFICAS

Adelman, J. S., Brown, G. D. A., & Quesada, J. F. (2006). Contextual diversity, not word frequency, determines word naming and lexical decision times. *Psychological Science, 17*, 814–823.

Alaniz Macedo. A., Campos Pimentel. M.C, Camacho-Guerrero, J.A. (2002), An infrastructure for open latent semantic linking. Conference on Hypertext and Hypermedia archive. *Proceedings of the thirteenth ACM conference on Hypertext and hypermedia*. College Park, Maryland, USA.

Álvarez, C.A, Alameda, R. y Domínguez, A. (1999). *Procesamiento ortográfico y silábico.* En Manuel de Vega y Fernando Cuetos (Eds.). *Psicolingüística del español* (pp.89-130). Trotta: Madrid.

Anderson, J.R. (1990). *The adaptative character of thought*. Hillside, NJ: Erlbaum.

Andrews ,S.(1996).Lexical retrieval and selection processes:effects of transposed-letter confusability. *Journal of Memory and Language*, 35, 775-800.

Ans, A., Carbonnel, S., & Valdois, S..(1998).A connectionist multiple-trace memory model for polysyllabic word reading. *Psychological Review*, 105, 678–723.

Barsalou, L.W., & Wiemer-Hastings, K. (2005). Situating abstract concepts. In D. Pecher and R. Zwaan (Eds.), *Grounding cognition: The role of perception and action in memory, language, and thought* (pp. 129-163) . New York: Cambridge University Press.

Besner, D., & Joordens, S. (1995). Wrestling with ambiguity—Further reflections: Reply to Masson and Borowsky (1995) and Rueckl (1995). *Journal of Experimental Psychology: Learning, Memory, and Cognition, 21*, 515-519.

Berry, M. W., Dumais, S. T. y O'Brien, G.W. (1995). Using linear algebra for intelligent information retrieval. *SIAM: Review*, 37, 573-595.

Binder JR, Westbury CF, McKiernan KA, Possing ET & Medler DA. (2005). Distinct brain systems for processing concrete and abstract concepts. *Journal of Cognitive Neuroscience*. 17, 905-917.

Blackmon, M. H., Polson, P. G., Kitajima, M., & Lewis, C. Cognitive Walkthrough for the Web. In Proceedings of *CHI'2002*, (2002). *Proceedings of the conference on Human Factors in Computing Systems*, 463-470.

Blackmon, M.H. (2004). Cognitive walkthrough. In W.S. Bainbridge (Ed)., *Encyclopedia of Human-Computer Interaction* (Vol.1,pp.104-197). Great Barrington MA: Berkshire Publishing.

Blackmon, M.H.; Mandalia, D.(2004). *Steps of the cognitive Walkthrough for the web(CWW):. Navigation System Analysis*. Institute of Cognitive Science. University of Colorado: Boulder.

Blackmon,M.H.; Kitajima,M.; Polson, P.(2003). Repairing Usability problems Identified by the cognitive Walkthrough for the web(CWW). *Proceedings of ACM CHI 2003:Conference on Human Factors in Computing Systems*, 497-504.

Bleasdale F. A. (1987). Concreteness-dependent associative priming: Separate lexical organization for concrete and abstract words. *Journal of Experimental Psychology: Learning, Memory, & Cognition, 13*, 582-594.

Börner, K., Chen, C., & Boyack, K. (2003). Visualizing knowledge domains. *Annual Review of Information Science & Technology, 37,* 179–255.

Börner, K., Sanyal, S., & Vespignani, A. (2007). Network Science. *Annual Review of Information Science & Technology, 41*, 537-607.

Bransford, J.D. & McCarrell, N.S. (1974). A sketch of a cognitive approach to comprehension. In W. Weimer & D. Palermo (Eds.), *Cognition and the symbolic processes* (pp. 189-229). Hillsdale, NJ: Lawrence Erlbaum Associates.

Brown, W. P., & Ure, D. M. J. (1969). Five rated characteristics of. 650 word association stimuli. *British Journal of Psychology*, 60, 233-249.

Brumby, D. & Howes, A. (2004). Good enough but I´ll just check: Web page search as attentional refocusing. *Proceedings of the 6th International Conference on Cognitive Modelling*, 46-51.

Burek, G.G,Vargas-Vera, M. (2004). Document retrieval based on intelligent query formulation. Techreport ID: kmi-04-13 [Previously known as KMI-TR-148] June 2004.

Burek,G.G,Vargas-Vera, M., Moreale, E.(2004). Indexing Student Essays Paragraphs Using LSA over an Integrated Ontological Space. *Technical Report KMI-04-08.*

Burgess, C., & Lund, K. (2000). The dynamics of meaning in memory. In Dietrich, E., & Markman, A. B. (Eds.), *Cognitive Dynamics: Conceptual Change in Humans and Machines*. (pp. 117-156). Lawrence Erlbaum Associates, Publishers..

Burgess, C. (2000). Theory and operational definitions in computational memory models: A response to Glenberg and Robertson. *Journal of Memory and Language*, 43, 402-408.

Campbell, R.S., & Pennebaker, J.W. (2003). The secret life of pronouns: Flexibility in writing style and physical health. *Psychological Science*, 14, 60-65.

Carreiras,M.,Perea,M,&Grainger,J..(1997).Effects of orthographics neighborhood in visual word recognition: cross-task comparisons. *Journal of Experimental Psychology: Learning Memory & Cognition,23*, 857-871.

Cattell, J. M. (1886). The time it takes to see and name objects. *Mind,11*, 53–65.

Cederberg, S. y Widdows D. (2003). Using LSA and noun coordination information to improve the precision and recall of automatic hyponymy extraction. *Human Language Technology Conference archive Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003* - Vol 4. Edmonton, Canada. 111-118.

Chambers, N., Tetreault, T., y Allen, J. (2005). Approaches for Automatically Tagging Affect. In J.G. Shanahan, Yan Qu & J. Wiebe (Eds.), *Computing Attitude and Affect in Text: Theory and Applications. S*priner.

Chen, C. (1997). Tracking latent domain structures: An integration of Pathfinder and Latent Semantic Analysis. *Artificial Intelligence & Society, 11 (1-2),* 48-62.

Chen, C. & Czerwinski, M. (1998). From latent semantics to spatial hypertext: An integrated approach. *In proceedings of the 9th ACM Conference on Hypertext and Hypermedia (Hypertext '98*), Pittsburgh, PA, 77-86.

Chomsky, N.: 1991, 'Linguistics, a Personal View', in A. Kasher (Ed.), *The Chomskyan Turn, Blackwell*, Oxford.

Chu-Carroll, J., Carpenter, B.(1999.) Vector-based Natural Language Call Routing. *Computational Linguistics*, 25,(3), 361-388.

Chung, C. K., & Pennebaker, J.W. (in press). The psychological functions of function words. In K. Fiedler (Ed.), *Frontiers of social psychology: Social communication*. Hove, UK: Psychology Press.

Colangelo, A., Buchanan, L., & Westbury, C. (2004). Deep Dyslexia and Semantic errors: a test of the failure of inhibition hypothesis using a semantic blocking paradigm. *Brain and Cognition, 54*, 232-234.

Colangelo, A., & Buchanan, L. (2005). Semantic ambiguity and the failure of inhibition hypothesis as an explanation for reading errors in deep dyslexia. *Brain & Cognition.*

Collins, A. & Quillian, M. R. (1969). Retrieval time from semantic memory. *Journal of Verbal Learning and Verbal Behavior 8 (2),* 240-248.

Collins, A. & Loftus, J. (1975). A spreading activation theory of semantic memory. *Psychological Review, 82,* 407-428.

Coltheart,M. , Davelaar,E., & Jonasson,J.F. .(1977).Access to the internal lexicon. Attention & performance VI, 535-555.*In S. Dornic Hillsdale*, N.J:Erlbaum.

Coltheart, M., Rastle, K., Perry, C., Ziegler, J., & Langdon, R. (2001).DRC: A Dual Route Cascaded model of visual word recognition and reading aloud. *Psychological Review*,108, 204–256.

Coltheart, M., Rastle, K., Perry, C., Ziegler, J., & Langdon, R. (2001).DRC: A Dual Route Cascaded model of visual word recognition and reading aloud. *Psychological Review*,108, 204–256.

Cox, S. & Shahshahani, B. A Comparison of some Different Techniques for Vector Based Call-Routing. *Proc. 7th European Conf. on Speech Communication and Technology*, Aalborg, September 2001

Crutch, S. J., Ridha, B. H., & Warrington, E. K. (2006). The different frameworks underlying abstract and concrete knowledge: Evidence from a bilingual patient with a semantic refractory access dysphasia. *Neurocase*, 12, 151–163.

Crutch, S. J. & Warrington, E. K. (2005). Abstract and concrete concepts have structurally different representational frameworks. *Brain, 128*, 615-627.

Dam, G. & Kaufmann, S. (in press). Computer Assessment of Interview Data using Latent Semantic Analysis. *Behavior Research Methods* .

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing By Latent Semantic Analysis. *Journal of the American Society for Information Science, 41,* 391-407.

DeGroot, A.M.B.(1989). Representational aspects of Word imageability and Word frecuency as assessed through Word association. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 15*, 824-845.

Denhière G., Lemaire, B. (2004) A Computational Model of Children's Semantic Memory, in *Proceedings of the 26th Annual Meeting of the Cognitive Science Society* (CogSci'2004), 297-302.

Denhière, G., Lemaire, B. (2004) Representing children's semantic knowledge from a multisource corpus. In *Proceedings of the 14th Annual Meeting of the Society for Text and Discourse*, Chicago.

Denhière, G., Lemaire, B. (2004). A computational model of a child semantic memory, submitted to the 26h Annual Meeting of the Cognitive Science Society, August 2004.

Denhiére, G., Lemaire, B., Bellissens, C. & Jhean-Larose, S. (2007). A semantic space modeling children's semantic memory. En T. Landauer, D. McNamara, S. Dennis & W. Kintsch (Eds.) *The handbook of Latent Semantic Analysis* (pp. 143-167). Mahwah, NJ: Erlbaum.

Deniston, M(2003) Concept Searching: An Overveiw and Discussion of Concept Search Models and Technologies. *FIOS*. Electronic discovery simplified.

Doyle. J. K., Ford ,D.N., (1998) Mental models concepts for system dynamics research. *System Dynamics Review*, Volume 14, Issue 1 , Pages 3 - 29

Duffy, S. A., Morris, R. K., & Rayner, K. (1988). Lexical ambiguity and fixation times in reading. *Journal of Memory and Language*, 27, 429-446.

Dumais,S. (2003),Data-Driven approaches to information access, *Cognitive Science* 2,491-524.

Duñabeitia, J.A., Avilés, A., Afonso, O., Scheepers, C., & Carreiras, M. (2009). Qualitative differences in the representation of abstract versus concrete words: Evidence from the visual-world paradigm. *Cognition, 110*, 284-292.

Dybkjær, L, Bernsen, N.O.,(2001) Usability evaluation in spoken language dialogue systems. Annual Meeting of the ACL archive. *Proceedings of the workshop on Evaluation for Language and Dialogue Systems* - Volume 9, Toulouse, France.

Dybkjær, L., Bernsen, N.O., and Dybkjær, H.(1998): A methodology for diagnostic evaluation of spoken human-machine dialogue. *International Journal of Human Computer Studies (Special issue on Miscommunication)*, 48, 1998, 605-625.

Estévez, A. (1991). Estudio normativo sobre ambigüedad en castellano. *Cognitiva, 3*, 237-271.

Fodor, J. A. (1983). The modularity of mind. Cambridge, MA: MIT Press.

Foltz, P. W. (1996). Latent Semantic Analysis for text-based research. *Behavior Research Methods, Instruments and Computers, 28*, 197-202.

Foster K,I. & Shen,D. .(1996).No enemies in the neighborhood: absence of inhibitory neighborhood effects in lexical decision and semantic categorization. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 22, 696-713.

Foster,K.I, & Hector,J.(2002).Cascaded vs. non-cascaded models of lexical and semantic processing: the turple effect . *Memory & Cognition*, 30, 7, 1106-1116.

Franceschetti, D.R., Karnavat, A., Marineau, J., McCallie, G.L., Olde, B.A., Terry, B.L., & Graesser, A.C. (2001). Development of physics test corpora for latent semantic analysis. *Proceedings of the 23th Annual Meeting of the Cognitive Science Society* (pp. 297-300). Mahwah, NJ: Erlbaum.

Frazier, L. & K. Rayner (1982). Making and correcting errors during sentence comprehension: eye movements in the analysis of structurally ambiguous sentences, *Cognitive Psychology* 14, 178-210.

Freeman, J.T., Bryan, F., Thompson, T., & Cohen, M. (2000) Modelling and Diagnosing Domain Knowledge Using Latent Semantic Indexing, *Interactive Learning Environments*. Volume 8, Number 3.

Fruchterman, T. M. & Reingold, E. M. (1991). Graph drawing by force-directed placement. *Software-Practice & Experience, 21*, 1129–1164.

Furnas, G. W., Gomez, L. M., Landauer, T. K., & Dumais, S. T. (1982). Statistical semantics: How can a computer use what people name things to guess what things people mean when they name things? In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 251-253). ACM.

Gil-Leiva, I. y Rodriguez-Muñoz, J.(1997) Análisis de los descriptores de diferentes áreas de conocimiento. *Revista Española de Documentación Científica*, vol. 20, nº 2, p. 150-160

Glenberg, A. M. (1997). What memory is for. *Behavioral and Brain Sciences*, *20,* (1), 1-55.

Glenberg, A. M., & Robertson, D. A. (2000). Symbol grounding and meaning: A comparison of high-dimensional and embodied theories of meaning. *Journal of Memory and Language*, 43,(3), 379–401.

Goldman, S., Varma, S. & Coté, N. (1996). Extending capacity-constrained construction integration: Toward "smarter" and fexible models of text comprehension. In B. Britton & A.C. Graesser (Eds.), *Models of understanding text* (pp. 73-113). Hillsdale, N.J.: Erlbaum.

Gracia,J.M.(2002) Álgebra lineal tras los buscadores de internet. Universidad del País Vasco. Matemática aplicada y estadística. Informe Técnico.

Graesser, A., Singer, M. & Trabasso, T. (1994). Constructing inferences during narrative text comprehension. *Psychological Review*, 101, 371-395.

Graesser, A., McNamara, D.S., Louwerse, M., & Cai, Z. (2004). Coh-Metrix: Analysis of text on cohesion and language. *Behavioral Research Methods, Instruments, and Computers*, 36, 193-202.

Graesser, A., Wiemer-Hastings, P., Wiemer-Hastings, K., Harter, D., Person, N., y the Tutoring Research Group (2000). Using Latent Semantic Analysis to evaluate the contributions of students in AutoTutor. *Interactive Learning Environments*, 8(2):129–147.

Griffiths, T. L., & Steyvers, M. (2004). Finding scientific topics. *Proceedings of the National Academy of Sciences, 101*, 5228-5235.

Grossman, M, Koenig, P, DeVita, C, Glosser, G, Alsop, D, Detre, J, Gee, J, (2002). The neural basis for category-specific knowledge: An fMRI study, *Neuroimage 15*, 936-948.

Guerrero-Bote, V. et al. (2006). Binary Pathfinder: An improvement to the Pathfinder algorithm. *Information Processing & Management, 42 (*6*)*, December 2006, 1484-1490.

Haley D., Thomas,P., Nuseibeh,P., Taylor J., & Lefrere P (2003). E-Assessment using Latent Semantic Analysis. *3rd International LeGE-WG Workshop: GRID Infrastructure to Support Future Technology Enhanced Learning*. Berlin, Germany. 3 December, 2003.

Haley, D.T., Thomas, P., De Roeck, A., & Petre, M. (2005), A research Taxonomy for Latent Semantic Análisis-Based Educational Applications, *Technical Report nº 2005/09.* The Open University.

Haley, D.T., Thomas, P., Petre, M. & De Roeck, A. (2007) Seeing the Whole Picture: Comparing Computer Assisted Assessment Systems using LSA-based Systems as an Example. *Technical Report Number 2007/07*. Open University - Department of Computing.

Hebb, D.O. (1949). The organization of behavior. New York: John Wiley.

Hino, Y. Lupker, S.J & Pexman P. M. (2002). Ambiguity and synonymy effects in lexical decision, naming, and semantic categorization tasks: Interactions between orthography, phonology, and semantics. *Journal of Experimental Psychology: Learning, Memory, & Cognition, 28*, 686-713.

Holcomb, P.J., Grainger, J., & O'Rourke, T. (2002). An electrophysiological study of the effects of orthographic neighborhood size on printed word perception. *Journal of Cognitive Neuroscience*, 14, 938-950.

Howell, J. R., & Bryden, M. P. (1987). The effects of word orientation and image ability on visual half-field presentations with a lexical decision task. *Neuropsychologia, 25*, 527-538.

Hu, X., Cai, Z. Louwerse, M., Olney, A., Penumatsa, P., Graesser, A.C., & TRG (2003). Paper presented at the 2003 International Joint Conference on Artificial Intelligence. México.

Hu, X., Cai, Z., Wiemer-Hasting, P., Graesser, A., & McNamara, D. S. (2006). Strengths, limitations, and extensions of LSA. In T. Landauer, D. S. McNamara, S. Dennis, & W. Kintsch (Eds.), *LSA: A road to meaning*. Mahwah, NJ: Erlbaum.

Ide, N., & Véronis, J. (1998). Word sense Disambiguation. The state of Art. *Computational Linguistics*, 24, (1), 1-41.

Jared, D.,& Seidenberg,M.S. .(1990).Naming multisyllabic words. *Journal of experimental Psychology:Human perception and performance*,16, 92-105.

Jenni , M., & Ben Ali, I. (2004). Automatic answering tool for e-learning environment.Ecole Supérieure des Sciences et Techniques de Tunis Research Unit of Technologies of Information and Communication (UTIC). Technical Report 2004.

Johnston, J. C., & McClelland, J. L. (1973). Visual factors in word perception. *Perception & Psychophysics*, 14, 365-370.

Joordens, S., & Besner, D. (1994). When banking on meaning is not (yet) money in the bank: Explorations in connectionist modeling. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 20*, 1051-1062.

Jorge-Botana, G., León, J.A.,Olmos, R., & Escudero, I. (2010). Latent Semantic Analysis Parameters for Essay Evaluation using Small-Scale Corpora. *Journal of Quantitative Linguistics,* Vol 17, 1,. 1–29.

Jorge-Botana, G., León, J.A.,Olmos, R. & Hassan-Montero, Y. (in press). Visualizing polysemy using LSA and the predication algorithm. *Journal of the American Society for Information Science and Technology.*

Jorge-Botana, G., Olmos, R., & León J.A. (2009) Using LSA and the predication algorithm to improve extraction of meanings from a diagnostic corpus. *Spanish Journal of Psychology*. Vol. 12, 2, 424-440.

Juvina, I. & van Oostendorp, H. (2005). Bringing cognitive models into the domain of web accessibility. In *Proceedings of the HCII2005 Conference*, Las Vegas, USA.

Juvina, I., van Oostendorp, H., Karbor, P. & Pauw, B. (2005). Towards modeling contextual information in web navigation. In B. G. Bara & L. Barsalou & M. Bucciarelli (Eds.), *In Proceedings of the 27th Annual Meeting of the Cognitive Science Society*, CogSci2005. Austin, Texas: The Cognitive Science Society, Inc, (pp. 1078-1083).

Juvina, I., van Oostendorp, H., Karbor, P., & Pauw, B. (2005). Toward Modeling Contextual Information in Web Navigation. CogSci 05, Stresa, Italy. http://www.cs.uu.nl/people/ion/

Kamada, T. & Kawai, S. (1989). An algorithm for drawing general undirected graphs. *Information Processing Letters, 31*, 7–15.

Kaur, I. ; Hornof, A. J. (2005). A comparison of LSA, WordNet and PMI-IR for predicting user click behavior. *Proceedings of ACM CHI 2005: Conference on Human Factors in Computing Systems*, New York: ACM, pp. 51-60.

Kersten, A. W., & Earles, J. L. (2004). Semantic context influences memory for verbs more than memory for nouns. *Memory & Cognition*, 32, 198-211.

Kersten, A.W., & Smith, L.B. (2002). Attention to novel objects during verb learning. *Child Development*, 73, 93-109.

Kiefer, B. (2005). *Information Retrieval: Perspectives on Next Generation Search*. V.P. Hosting Applications (Research and Development). http://www.caseshare.com/articles-and-white-papers/72-catalyst-concept-search-perspectives-on-next-generation-information-retrieval.html

Kiekel, P. A., Cooke, N. J., Foltz, P. W., & Shope, S. M. (2001). Automating measurement of team cognition through analysis of communication data. In M. J. Smith, G. Salvendy, D. Harris, and R. J. Koubek (Eds.), *Usability Evaluation and Interface Design*, pp. 1382-1386, Mahwah, NJ: Lawrence Erlbaum Associates.

Kilgarriff, A. (1997). I don't believe in word senses. *Computers and the Humanities* 31 (2), 91-113.

Kintsch, W. & Mangalath, P. (in press) The construction of meaning. *Topics in Cognitive Science.*

Kintsch, E. (1990). Macroprocesses and microprocesses in the development of summarization skill. *Cognition and Instruction*, 7, 161-195.

Kintsch, W. (1998), *Comprehension: A paradigm for cognition*. New York: Cambridge University Press.

Kintsch,W.(1998).The Representation of Knowledge in Minds as Machines .*International Journal of Psychology*. Volume 33, Number 6 / December 1, 1998 pp:411 - 420

Kintsch, W., Patel, V.L., & Ericsson, K. A. (1999). The role of long-term working memory in text comprehension. *Psychologia* 42, 186-198.

Kintsch, W.(2000). Metaphor comprehension: A computational theory. *Psyhonomic Bulletin and Review*., 7, 257-266.

Kintsch, W.(2001) Predication. *Cognitive Science*. 25, 173-202.

Kintsch, W. and Bowles, A. (2002) Metaphor comprehension:What makes a metaphor difficult to understand?. *Metaphor and Symbol*, 2002, 17, 249-262.

Kintsch, W. (2002). On the notion of theme and topic in psychological process models of text comprenhension. Thematic: interdisciplinary studies. In M. Louwerse y W. Van Peer (Eds.) Amsterdam: John Benjamins.

Kintsch, W., & Bowles, A. (2002). Metaphor comprehension: What makes a metaphor difficult to understand? *Metaphor and Symbol*, *17*, 249-262.

Kintsch, W. (2008). Symbol systems and perceptual representations. In M. de Vega, A. M. Glenberg, & A. C. Graesser (Eds.), *Symbols and embodiment: Debates on meaning and cognition* (pp. 145-164). Oxford: Oxford University Press.

Kitajima, M., Blackmon, M.H., & Polson, P.G., (2000) A Comprehension-based Model of Web Navigation and Its Application to Web Usability Analysis, in People and Computers XIV. (pp. 357-373), Springer. (Disponible en: http://www3.nibh.jp/~kitajima/CognitiveModeling/WebNavigationDemo/CoLiDeSTopPage.html).

Klein, D. E., & Murphy, G. L. (2001). The representation of polysemous words. *Journal of Memory and Language, 45*, 259-282.

Klepousniotou, E (2002) The Processing of Lexical Ambiguity: Homonymy and Polysemy in the Mental Lexicon. *Brain and Language*. 81,(1-3), pp.205-223.

Kontostathis, A. & Pottenger, W.M. (2002). Detecting Patterns in the LSI Term-Term Matrix. Workshop on the Foundation of Data Mining and Discovery, IEEE International Conference on Data Mining.

Kontostathis, A. & Pottenger, W.M.. (2006) A framework for understanding LSI performance. *Information Processing and Management*, 42, 1, 56-73.

Kontostathis, A., Pottenger, W.M., & Davison, B.D. (2005) Identification of critical values in Latent Semantic Indexing (LSI). In T.Y. Lin, S. Ohsuga, C. Liau, X. Hu and S. Tsumoto (Eds.), *Foundations of Data Mining and Knowledge Discovery*, (pp. 333-346). Springer-Verlag.

Kurby, C. A., Wiemer-Hastings, K., Ganduri, N., Magliano, J. P., Millis, K. K., & McNamara, D. S. (2003). Computerizing reading training: Evaluation of a latent semantic analysis space for science text. *Behavior Research Methods, Instruments, & Computers*, 35, 244-250.

Laham,D.,Bennett,W., & Derr,M.(2000). Latent semantic Analysis for Career Field Analysis and Information Operations Air Force Research Laboratory, Mesa Branc.

Landauer, T. K., & Dumais, S. T. (1996). How come you know so much? From practical problem to theory. In D. Hermann, C. McEvoy, M. Johnson, & P. Hertel (Eds.), Basic and applied memory: Memory in context. Mahwah, NJ: Erlbaum, 105-126.

Landauer T. K., & Dumais S. T.(1997). A solution to Plato's problem: The latent semantic analysis theory of the acquisition, induction, and representation of knowledge. *Psychological Review, 104*, 211-240.

Landauer, T. K., Laham, D., Rehder, R., & Schreiner, M. E.(1997). How well can passage meaning be derived without using word order? A comparison of Latent Semantic Analysis and humans. *In Proceedings of the 19th Annual Conference of the Cognitive Science Society*, pp. 412–417 Mahwah, NJ. Erlbaum.

Landauer , T. K., (1999). Latent semantic Analysis is a Theory of the Psychology of Language and Mind. *Discourse Processes*, 27, 303-310.

Landauer, T.K., Foltz, P.W., Laham, D. (1998). An introduction to Latent Semantic Analysis. *Discourse Processes*, 25, 259-284.

Landauer,T.K., & Laham,D. (1998). Learning Human-like Knowledge by Singular Value Decomposition: A Progress Report. *Advances in Neural Information Processing Systems.*

Landauer, T.K., & Psotka, J.(2002) Simulating text understanding for educational applications with Latent Semantic Analysis: Introduction to LSA. US Army Research Institute.

Landauer, T. K. (2002). On the computational basis of learning and cognition: Arguments from LSA. In N. Ross (Ed.), *The psychology of learning and motivation,* 41, 43-84.

Landauer T.K., Laham,D., & Foltz,P. (2003). Automatic essay assessment. *Assessment in Education: Principles, Policy and Practice*,10, 3, pp. 295-308(14).

Lee, T.S., & Nguyen, N. (2001) Dynamics of subjective contour formation in the early visual cortex. *In Proceedings of the National Academy of Science*, vol. 98, Issue 4, 1907-1911.

Lee M., Cimino, L.M., Zhu H, J., Sable C., Shanker, V., & Ely, J. (2006). Beyond information retrieval Medical question answering. *In Proceedings of the American Medical Informatics Association*. Washington DC, USA. 469–473.

Lemaire B., Bianco M. (2003) Contextual Effects on Metaphor Comprehension: Experiment and Simulation. *In Proceedings of the 5th International Conference on Cognitive Modelling* (ICCM'2003), 153-158.

Lemaire, B., Benoit & Bianco, M.(2003). Contextual Effects on Metaphor Comprehension: Experiment and Simulation. In F. Detje, D. Dörner, and H. Schaub, (Eds.), *Proceedings 5th International Conference on Cognitive Modelling* (ICCM), 153-158, Bamberg, Germany.

Lemaire,B. & Denhière, G. (2006). Effects of High-Order Co-occurrences on Word Semantic Similarity; *Current Psychology Letters- Behaviour, Brain and Cognition*, vol. 18.

Lemaire, B., Denhière, G., Bellissens, C., Jhean-Larose, S. (2006) A Computational Model for Simulating Text Comprehension. *Behavior Research Methods* 38,(4), 628-637

León, J.A., Olmos, R., Escudero, I., Cañas, J.J. & Salmerón, L. (2006). Assessing short summaries with human judgments procedure and latent semantic analysis in narrative and expository texts. *Behavior Research Methods, Instruments, and Computers, 38,(4),* 616-627.

Lin, Chien-Jer Charles, & Ahrens, Kathleen. (2010). Ambiguity advantage revisited: Two meanings are better than one when accessing Chinese nouns. Journal of *Psycholinguistic Research. 39*. 1-19.

Macdonal, M. C. (1994). Probabilistic constraints and asyntactic ambiguity resolution. *Language and Cognitive Processes*, 9, 157–201.

Madhusudanan, A. (2006), Brainnet 1 - A Neural Netwok Project - With Illustration And Code - Learn Neural Network Programming Step By Step And Develop a Simple Handwriting Detection System. The code project. http://www.codeproject.com.

Magomedov, B. (2006). Hopfield model of neural network for pattern recognition.The code Project. http://www.codeproject.com.

Mandl, T. (1999): Efficient Preprocessing for Information Retrieval with Neural Networks. In: Zimmermann, Hans-Jürgen (ed.): *EUFIT '99. 7th European Congress on Intelligent Techniques and Soft Computing.* Aachen, Germany.

Mandl, T. (1998). Tolerant and adaptative Information Retireval with neural Networks. *Technical report. Information science-University of Hildesheim*

Marr, D. (1985). *Vision*. San Francisco, CA: Freeman.

Mayes, M., Drewes, B., & Thompson (2002), W. SAS Text Miner, Distilling Textual Data for Competitive Business Advantage, A SAS white paper.

McClelland, J. L. and Rogers, T. T. (2003). The Parallel Distributed Processing Approach to Semantic. *Cognition. Nature Reviews Neuroscience*, 4, 310-322.

McClelland, J.L. and Patterson, K. (2002) Rules or connections in past tense inflections: what does the evidence rule out? Trends Cogn. Sci. 6, No. 11 465–472; and Reply to Pinker and Ullman. *Trends in Cognitive Science*. 6, 464–465.

McCulloch, W. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biophysics, 7:115 - 133.

McKoon, G. & Ratcliff, R. (1992). Spreading activation versus compound cue accounts of priming: Mediated priming revisited. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 18, 1155-1172.

McNamara, D.S., Louwerse, M.M., Cai, Z., & Graesser, A. (2005, January 1). Coh-Metrix version 1.4. from http//:cohmetrix.memphis.edu.

Medin, D. (1989). Concepts and conceptual structure. *American Psychologist*, 44, 1469-1481.

Mehler, A.& Sichelschmidt, L. (2006): Reconceptualizing Latent Semantic Analysis in Terms of Complex Network Theory. A Corpus-Linguistic Approach. Accepted at the *Second International Conference of the German Cognitive Linguistics Association* - Theme Session: Cognitive-linguistic approaches: What can we gain by computational treatment of data? 5.-7. Oktober 2006, Ludwig-Maximilians-Universität München, 23-26.

Mill, W. & Kontostathis, A. (2004) Analysis of the values in the LSI term-term matrix . Technical Report.

Moya-Anegón, F., Vargas, B., Chinchilla, Z., Corera, E., Gonzalez, A., Munoz, F., et al. (2007). Visualizing the Marrow of Science. *Journal of The American Society for Information Science and Technology, 58,* 2167-2179.

Murray, H.A. "Thematic Apperception Test" (1943) Cambridge, Massachusets.

Nakov P. (2000): Getting Better Results with Latent Semantic Indexing. In *Proceedings of the Students Presentations at the European Summer School in Logic* Language and Information (ESSLLI'00), pp. 156-166.

Nakov, P., Popova, A. & Mateev, P. (2001). Weight functions impact on LSA performance. Paper presented at *Recent Advances in Natural Language Processing – RANLP 2001*, Tzigov Chark, Bulgaria.

Nakov P., E. Valchanova, & G. Angelova. (2003) Towards Deeper Understanding of LSA Performance. *In Proc. Recent Advances in Natural Language Processing*. pp. 311-318, Borovetz, Bulgaria.

Nakov,P.I. , Popova,A.., & Mateev, P.(2001) Weight Functions Impact on LSA Performance, (Sofia University Press, Sofia, 2001).

Olde, B. A., Franceschetti, D.R., Karnavat, Graesser, A. C. & the Tutoring Research Group (2002). The right stuff: Do you need to sanitize your corpus when using latent semantic analysis? *Proceedings of the 24th Annual Conference of the Cognitive Science Society* (pp.708-713). Mahwah, NJ: Erlbaum.

Olmos, R., León, J.A., Escudero, I. y Jorge-Botana, G. (2009), Análisis del tamaño y especificidad de los corpus en la evaluación de resúmenes mediante el LSA: Un análisis comparativo entre LSA y jueces expertos. *Rev. Signos*. 42, 69, pp. 71-81.

Olmos, R., León, J.A., Jorge-Botana, G. & Escudero, I. (2009). New algorithms assessing short summaries in expository texts using Latent Semantic Analysis. *Behavior Research Methods*. *41,* (3), 944-950.

Ozcan, R.; Aslandogan, Y.A. (2005). Information Technology: *Coding and Computing, 2005. ITCC 2005. International Conference* on Vol1, 4-6 April 2005, 794 - 799.

Padó, S. y Lapata, M. (2008). Dependency-based Construction of semantic Space Models. *Computational Linguistics*, 33(2):161–199, 2007.

Paivio, A. (1986), *Mental Representations*. New York: Oxford University Press.

Paivio, A. (1991), *Images in Mind: The Evolution of a Theory*. New York: Harvester-Wheasheaf.

Pakhomov S., Buntrock, J. D. & Chute, C. G. (2006). Automating the assignment of diagnosis codes to patient encounters using example-based and machine learning techniques. *Journal of the American Medical Informatics Association*, 13(5), 516-525.

Pedersen,T. , Patwardhan, S. , & Michelizzi., J. (2004) Wordnet: Similarity - measuring the relatedness of concepts. In Appears in the *Proceedings of the Nineteenth National Conference on Artificial Intelligence* (AAAI-04), http://citeseer.ist.psu.edu/644388.html.

Pennebaker J. W. , Francis M. E. , & Booth R. J. (2001). Linguistic Inquiry and Word Count. Erlbaum Publishers, Mahwah, NJ.

Pennebaker, J. W., Mehl, M.R., & Niederhoffer, K. (2003). Psychological aspects of natural language use: Our words, our selves. *Annual Review of Psychology*, 54, 547-577.

Perea, M. & Rosa, E. (2000). Efectos de la competición en el reconocimiento visual de palabras con la técnica de "priming enmascarado": Una aproximación psicofísica. *Anales de Psicología*,16, 2, 215-225.

Perea, M., & Lupker, S. J. (2003). Does judge activate COURT? Transposed-letter confusability effects in masked associative priming. *Memory and Cognition*, 31, 829-841.

Pexman, P. M. & Lupker, S. J. (1999). The impact of semantic ambiguity on visual word recognition: Do homophone and polysemy effects co-occur? *Canadian Journal of Experimental Psychology*, 53, 323-334.

Pexman, P. M., Hino, Y., & Lupker, S. J. (2004). Semantic ambiguity and the process of generating meaning from print. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 30,* 1252-1270.

Pexman, P. M., Hargreaves, I. S., Edwards, J. D., Henry, L. C., & Goodyear, B. G. (2007). Neural correlates of concreteness in semantic categorization. *Journal of Cognitive Neuroscience, 19*, 1407-1419.

Piercey, B. J., Parkinson, S. R., & Sisson, N. (1992). Effects of semantic similarity, omission probability and number of alternatives in computer menu search. *International Journal of Man-Machine Studies*, 37, 653-677.

Piercey, C. D., & Joordens, S. (2000). Turning an advantage into a disadvantage: ambiguity effects in lexical decision versus reading tasks. *Memory & Cognition* , 28, 657-666.

Pollatsek,A,Perea,M.,&Binder,K..().The effects of "neighborhood size" in reading and lexical decision. *Journal of Experimental Psychology: Human Perception & Performance*, 25,1142-1158.

Polson, P.G., Lewis, C., Rieman, I., & Wharton, C. (1992). Cognitive walkthroughs: A method for theory-based evaluation of user interface. *International Journal of man-machine Studies*, 36, pp. 741-773.

Quesada, J. (2006) Creating your own LSA space. In T. Landauer, D. McNamara, S.Dennis y W.Kintsch(Eds).Latent *Semantic Analysis: A road to meaning*. Erlbaum.

Quesada, J. (2007). Creating Your Own LSA Spaces. In T. K. Landauer, D. McNamara, S. Dennis & W. Kintsch (Eds.), *The handbook of Latent Semantic Analysis* (pp. 71-88). Mahwah, NJ: Erlbaum.

Quesada, J., Kintsch, W., & Gómez-Milán, E. (2001). Theory of complex problem solving using the vector space model. Latent Semantic Analysis applied to empirical results from adaptation experiments. *Cognitive Research with Microworlds*, 147-158.

Quesada, J., Kintsch, W. & Gómez-Milán, E. (2002). A theory of complex problem solving using Latent Semantic Análisis. In W.D. Gray & C.D. Schunn (eds.). *Proceedings of the 24th Annual Conference of the Cognitive Science Society*, Fairfax, VA. Lawrence Erlbaum Associates, Mahwah, NJ. 750-755.

Quirin, A., Cordón, O., Santamaria, J., Vargas-Quesada, B, & Moya-Anegón, F. (2008). A new variant of the Pathfinder algorithm to generate large visual science maps in cubic time. *Information Processing and Management, 44, (*4), 1611-1623.

Rodd, J., Gaskell, G. & Marslen-Wilson, W. (2002). Making sense of semantic ambiguity: semantic competition in lexical access. *Journal of Memory and Language, 46*, 245-266.

Rosch, E., & Mervis, C., B. (1975). Family resemblances: Studies in the internal structures of categories. *Cognitive Psychology, 7*, 573-605.

Rehder, B., Schreiner, M. E., Wolfe, M. B., Laham, D., Landauer, T. K. & Kintsch, W. (1998). Using Latent Semantic Analysis to assess knowledge: Some technical considerations. *Discourse Processes, 25*, 337-354.

Rodd, J., Gaskell, G. & Marslen-Wilson, W. (2002). Making sense of semantic ambiguity: semantic competition in lexical access. *Journal of Memory and Language*, 46, 245-266.

Rosch, E., y Mervis, C. B(1975) . Family resemblances: Studies in the internal structures of categories. *Cognitive Psychology*, 7, 573--605.

Rumelhart D.E y McClelland,(1992), *Introducción al procesamiento distribuido en paralelo*. Alianza Editorial, Madrid.

Rung-Ching Chen, Ya-Ching Lee, Ren-Hao Pan, (2006), Adding New Concepts On The Domain Ontology Based on Semantic Similarity, *International Conference on Business and Information.* July 12-14, 2006, Singapore.

Rusell, M. Latent semantic Analysis. EE3J2 DATAMINING. Lecture 8. University of Birmingham. Technical Report.

Sainz, J.S. (1991). *Conceptos naturales y conceptos artificiales*. En Mayor, J. y Pinillos, J.L. Tratado de Psicología General. Martinez-Arias, M.R. y Yela, M., 181-302. Madrid: Alhambra Longman.

Sainz.J.S. Mousikou.P.,Jorge-Botana.G .(2003). Conjoin letters into words:brain response correlates of lexical and attentional mechanims in lexical substitution errors. .*Congress of the Federation of European Psychophysiology Societies*. FEPS5.Bordeaux (F), September 10-14, 2003.

Salton, G. (1968). *Automatic Information Organization and Retrieval*. New York: McGraw-Hill.

Salton, G. and McGill, M. J. (1983) *Introduction to Modern Information Retrieval*. McGraw-Hill, New York, NY.

Samson, D., & Pillon, A. (2003). Concreteness effects in lexical tasks: Access to a mental image? *Brain and Language, 87*, 25-26.

Shepard, R. N. (1987). Toward a universal law of generalization for psychological science. *Science*, 237:1317-1323.

Schunn, C. D. (1999). The presence and absence of category knowledge in LSA. In the *Proceedings of the 21st Annual Conference of the Cognitive Science Society.* Mahwah, NJ: Erlbaum.

Schunn, C. D., & Vera, A. H. (1995). Causality and the categorization of objects and events. *Thinking & Reasoning*, 1(3), 237-284.

Schütze, H (1998). Automatic Word Sense Discrimination*, Journal of Computational Linguistics*, Volume 24, Number 2, 97 - 123.

Schwanenflugel PJ, Shoben EJ. (1983) Differential context effects in the comprehension of abstract and concrete verbal materials. *Journal of Experimental Psycholgy, Learning and Memory and Cognition, 9*, 82–102.

Schwanenflugel PJ, Stowe RW. (1989). Context availability and the processing of abstract and concrete words. *Reading Research Quarterly 24,* 114–26.

Schwanenflugel, P. J., & Akin, C. E. (1994). Developmental trends in lexical decisions for abstract and concrete words. *Reading Research Quarterly, 29*, 251-263.

Schwanenflugel P.J., Harnishfeger K.K., & Stowe R.W. (1998). Context availability and lexical decisions for abstract and concrete words. *Journal of Memory and Language, 27*, 499–520.

Schvaneveldt, R.W.(1990). *Pathfinder associative networks: Studies in knowledge organization*. Norwood. NJ: Ablex.

SCImago. (2007). SJR — SCImago Journal & Country Rank. Retrieved September 10, 2008, from http://www.scimagojr.com/.

Sears,C.R., Lupker,S.L, & Hino,Y. (1999). Orthographic neighborhood effects in perceptual identification and semantic-categorization tasks: A test of multiple read-out model. *Perception & Psychophysics*, 61, 1537-1554.

Sebastián-Gallés, N., Martí, M.A., Carreiras, M., & Cuetos, F. (2000). *LEXESP: Una base de datos informatizada del español*. Barcelona. Universitat de Barcelona.

Seidenberg ,M.S. (1987). Sublexical structures in visual word recognition:Access units or orthography redundancy?. Attention and performance XII: The psycology of reading , in M. Colheart(ed)Hillsdale , NJ:Erlbaum.

Seidenberg, M. S. and McClelland, J. L. (1989). A Distributed, Developmental Model of Word Recognition and Naming. *Psychological Review*, 96, 523-568.

Serafin, R. & Di Eugenio, B. (2003). FLSA: Extending Latent Semantic Analysis with features for dialogue act classification. In Proceedings of ACL04, *42nd Annual Meeting of the Association for Computational Linguistics*. Barcelona, Spain, July. (pp 692-es).

Serafin, R. & Di Eugenio. B. (2004). *FLSA: Extending Latent Semantic Analysis with features for dialogue act classification*. Annual Meeting of the Association for Computational Linguistics, Barcelona, Spain, July. (pp 692-es).

Silva, R. A. V., Martinez, A. S., & Ruiz, E. E. S. (2004). Categorizacao e análise de informacoes médicas. *In IX Congresso Brasileiro de Informaática em Saúde,* 2004, Ribeira˜o Preto. Anais do IX Congresso Brasileiro de Informática em Saude– CDROM, 2004.

Skoyles, John R. (1999) Autistic language abnormality:Is it a second-order context learning defect? – The view from Latent Semantic Analysis. In Barriere, I. and Chiat, and Morgan, S. G. and Woll, B., Eds. *Proceedings Child Language seminar*, pages 1, City University, London.

Steyvers, M., & Griffiths, T. (2007). Probabilistic topic models. In T. K. Landauer, D. McNamara, S. Dennis & W. Kintsch (Eds.), *Latent Semantic Analysis* (pp. 427-448). Mahwah, N.J.: Erlbaum.

Tanenhaus, M.K. & Trueswell, J.C. (1995). Sentence Comprehension. In Eimas & Miller (Eds.) Handbook in Perception and Cognition, Vol 11: *Speech Language and Communication*. Academic Press, pp. 217-262.

Tsai, Mei-Chih, Chu-Ren Huang, Keh-Jiann Chen, Kathleen Ahrens. 1998. Towards a Representation of Verbal Semantics – An Approach Based on Near Synonyms. *Computational Linguistics and Chinese Language Processing, 3*(1), 61-74.

Turney, Peter (2001) Mining the Web for Synonyms: PMI-IR versus LSA on TOEFL. In De Raedt, Luc and Flach, Peter, Eds. *Proceedings of the Twelfth European Conference on Machine Learning* (ECML-2001), pages pp. 491-502, Freiburg, Germany.

Usolab (2002). Algunos términos que los usuarios no entienden en la web de banca. Usolab, Febrero 2002. Disponible en: ttp://www.usolab.com/articulos/feb_02.php".

Van Dijk, T. (1972). *Some aspects of text grammars. A study in theoretical linguistics and poetics*, The Hague: Paris: Mouton.

Van Dijk, T. A., & Kintsch, W. (1983). *Strategies of discourse comprehension*. New York: Academic.

Van Heuven,W.J.B, Grainger,T., & Dijkstra,J.(1998).Orthographics neighborhood effects in bilingual word recognition. *Journal of Memory & language*, 39, 458-483.

Voice eXtensible Markup Language version 1.0 W3C Note 05 May 2000. http://www.w3.org/TR/Voicexm.l

Wade-Stein, D., & Kintsch, E. (2004). Summary Street: Interactive Computer Support for Writing. *Cognition and Instruction* (22), 333-362.

Walker,M.A, Litman,D. J., Kamm, C. A. & Abella, A. .(1998). Evaluating Spoken Dialogue Agents with PARADISE: Two Case Studies. *Computer Speech and Language*, 12(3):317–347.

Wandmacher, T. (2005). How semantic is Latent Semantic Analysis? *In Proceedings of RECITAL'05* (Dourdan, France).

Warren, R.M., & Warren, R.P. (1970): "Auditory illusions". *Scientific American, 223*, 30-36.

Weeds, J., D. Weir & D. McCarthy (2004) Characterizing measures of lexical distributional similarity. In *Proceedings of the 20th International Conference of Computational Linguistics*, COLING-2004. Geneva, Switzerland. pp 1015-1021 .

Widdows, D., & Dorow, D. (2002). *A Graph Model for Unsupervised Lexical Acquisition. In Proceedings of the 19th International Conference on Computational Linguistics*, Taipei, 1093-1099.

Wiemer-Hastings, P., Graesser, A.C., Harter, D., & the Tutoring Research Group (1998). The foundations and architecture of AutoTutor. *Proceedings of the 4th International Conference on Intelligent Tutoring Systems* (pp. 334-343). Berlin, Germany: Springer-Verlag.

Wiemer-Hastings, P., Wiemer-Hastings, K., and Graesser, A. (1999). Improving an intelligent tutor's comprehension of students with Latent Semantic Analysis. In *Proceedings of the International Conference on Artificial Intelligence in Education,* Le Mans, France, (pp 535-542).

Wiemer-Hastings, P., Wiemer-Hastings, K. & Graesser, A. (1999). Improving an intelligent tutor's comprehension of students with Latent Semantic Analysis. In S.P. Lajoie and M. Vivet (Eds.), *Artificial Intelligence in Education* (pp. 535-542). Amsterdam: IOS Press.

Wiemer-Hastings, P., Wiemer-Hastings, K., & Graesser, A. (1999). How Latent is Latent Semantic Analysis? In *Proceedings of the 16th International Joint Congress on Artificial Intelligence*, pp. 932–937, Morgan Kaufmann, San Francisco, 1999.

Wiemer-Hastings, P. (2000). *Adding syntactic information to LSA*. In *Proceedings of the 22nd Annual Conference of the Cognitive Science Society*, Erlbaum, Mahwah, NJ, 989–993.

Wiemer-Hastings, P., & Zipitria, I. (2001) Rules for syntax, vectors for semantics. In *Proceedings Of the 23rd Cognitive Science Conference*.pp,2001

Wiemer-Hastings, P. & Zipitria, I. (2001). Rules for syntax, vectors for semantics. In *Proceedings of the 23rd Cognitive Science Conference*. Mahwah, NJ: Lawrence Erlbaum Associates.

Wild. F, Stahl. C , Stermsek. G, Neumann. G : Parameters Driving Effectiveness of Automated Essay Scoring with LSA, in: *Proceedings of the 9th International Computer Assisted Assessment Conference* (CAA), 485-494, Loughborough, UK, July, 2005.

Wild. F,Stahl. C.,Stermsek. G.,Neuman.G.(2006) *Parameters driving effectiveness of automated essay scoring with LSA*. Report. Vienna University of Economics and Business Administration.

Williams, J. (1992). Processing polysemous words in context: Evidence for interrelated meanings. *Journal of Psycholinguistic Research* , 21, 193–218.

Wolfe, M. & Goldman, S. (2003). Use of latent semantic analysis for predicting psychological phenomena: Two issues and proposed solutions. *Behavior Research Methods, Instruments, and Computers*, 35(1), 22-31.

Wolfe, M., B. Schreiner, M. E., Rehder, B., Laham, D., Foltz, P. W., Kintsch, W. & Landauer, T. K (1998). Learning from text: Matching readers and texts by Latent Semantic Analysis. *Discourse Processes*, 25, 2&3, 309-336.

Yu, C., Cuadrado, J., Ceglowski, M., Payne, J. (2002) Patterns in Unstructured Data, Discovery, Aggregation, and Visualization, A Presentation to the Andrew W. Mellon Foundation.

Zipf, G. K. (1932): *Selected Studies of the Principle of Relative Frequency in Language*. Cambridge (Mass.).

Zhu, W., & Chen, C. (2007). Storylines: Visual exploration and analysis in latent semantic spaces. *Computers & Graphics, 31,*(3), 338-349.

Zwaan, R.A. & Radvansky G.A. (1998). Situation Models in Language Comprehension and Memory. *Psychological Bulletin, 123, (2)*, 162-185.