

Universidad Autónoma de Madrid

Escuela Politécnica Superior



Máster Universitario en Investigación e Innovación en TIC (i2-TIC)

TRABAJO DE FIN DE MÁSTER

RECOMENDACIÓN DE RECURSOS EDUCATIVOS ABIERTOS
UTILIZANDO SISTEMAS ADAPTATIVOS

Pablo Molins Ruano
Tutora: Pilar Rodríguez Marín

Junio 2018

Resumen

Resumen — Los recursos educativos abiertos (OER, por sus siglas en inglés) son aquellos que son distribuidos con una licencia que permite su libre modificación y redistribución, lo que supone una gran oportunidad para facilitar que el conocimiento se comparta y llegue a más personas. Pero cuántos más OER se crean y se distribuyen, más complicado se vuelve encontrar el recurso adecuado para cada persona y para cada contexto.

Crear sistemas de recomendación automáticos de OER que faciliten la labor de descubrimiento plantea una serie de retos. Por un lado, existen muchos perfiles de usuarios distintos que pueden estar interesados: profesores, alumnos en un curso, autodidactas. . . De esa multiplicidad de perfiles, llegamos también a una multiplicidad de objetivos, en las que incluso el mismo usuario puede cambiar de objetivo con el tiempo.

Hasta ahora, se han creado muchos sistemas enfocados a cada uno de esos objetivos, pero el panorama actual de “un problema, una propuesta” dificulta que se aprovechen las sinergias que pueden surgir en un entorno en el que se comparta más información entre algoritmos de recomendación.

Por otro lado, los OER son en sí mismos muy diversos. Existen muchas fuentes de OER, estructuradas de distintas formas, que ofrecen distintas facilidades de acceso y que se enfocan al final a distintos perfiles de usuario. Así mismo, los propios OER se presentan en muy distintos formatos (vídeos, texto, audio, software. . .).

En este trabajo se presenta una propuesta de arquitectura para la creación de un sistema de recomendación de OER multipropósito, basado en el principio de divide y vencerás, que pueda trabajar con múltiples fuentes, formatos y algoritmos de recomendación, para así lograr un sistema que aproveche todas las sinergias. Junto a esta arquitectura, se exploran propuestas de implementación para cada una de las cinco etapas que componen la arquitectura.

Índice general

I. Introducción	1
I.1. Motivación	3
I.2. Alcance y objetivos del trabajo	6
I.3. Estructura del documento	6
II. Estado del Arte	9
II.1. Open Educational Resources	9
II.1.1. ¿Qué es una licencia abierta?	9
II.1.2. Líneas de investigación abiertas en los OER	10
II.2. Trabajos relacionados	12
II.2.1. 1 st Workshop on Recommender Systems for Technology Enhanced Learning - 2010	12
II.2.2. Trabajos más recientes (2017-2018)	14
II.2.3. <i>Recommendation of Open Educational Resources. An approach based on Linked Open Data.</i>	16
III.Arquitectura propuesta	17
III.1.Descripción general	17
III.2.Retos a superar	22
III.2.1. Heterogeneidad de fuentes	22
III.2.2. Heterogeneidad de formatos	23
III.2.3. Heterogeneidad de estructura lógica	24
III.2.4. Diversidad de objetivos	24
III.3.Definición y justificación de cada etapa	25
III.3.1. Etapa 0: Recolección de OER desde distintas fuentes	25
III.3.2. Etapa 1: Transformación de los OER a una representación común	26
III.3.3. Etapa 2: Creación de los modelos de recurso y usuario	27
III.3.4. Etapa 3: Explotación de los datos	28
III.3.5. Etapa 4: Retroalimentación del proceso	29
III.4.Discusión: fortalezas y debilidades de la propuesta	30
IV.Evaluación de la viabilidad de la arquitectura	33
IV.1.Etapa 0: Recolección de OER desde distintas fuentes	33
IV.2.Etapa 1: Transformación de los OER a una representación común	35
IV.3.Etapas 2 - 3: Creación de los modelos de recurso y usuario - Explotación de los datos	36

ÍNDICE GENERAL

IV.4. Etapa 4: Retroalimentación del proceso	38
IV.5. Ejemplo de aplicación de la arquitectura	39
V. Conclusiones y trabajo futuro	43
Referencias	47

Índice de figuras

III.1. Etapas de la arquitectura	21
III.2. Etapa 0: Recolección de OER desde distintas fuentes.	25
III.3. Etapa 1: Transformación de los OER a una representación común.	26
III.4. Etapa 2: Creación de los modelos de recurso y de usuario.	27
III.5. Etapa 3: Explotación de los datos.	28
III.6. Etapa 4: Retroalimentación del proceso.	29
III.7. <i>Standards</i> (viñeta de xkcd)	31
IV.1. Captura de pantalla del prototipo de implementación de la arquitectura.	40

Introducción

La Declaración Universal de los Derechos Humanos, en su artículo 26.1, declara que **“toda persona tiene derecho a la educación.”** La proclamación de un derecho es un primer paso fundamental para que este derecho llegue a ser disfrutado por todas las personas, pero no es suficiente. Hacen falta medidas concretas que respalden los derechos.

¿Qué medidas pueden ser estas? Discutir todas ellas llevaría más espacio que el aquí disponible y no es el objetivo de este trabajo. Este trabajo, se centra en una de esas posibles medidas. En concreto, en los Recursos Educativos Abiertos, o como son más conocidos, los *Open Educational Resources* (OER).

Es difícil elegir una fecha para señalar el nacimiento de los OER, pero sí se puede decir que en la última década han florecido masivamente. Sin duda una fecha clave es la de junio de 2012, cuando la Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura (UNESCO) celebró en París el *2012 World Open Educational Resources (OER) Congress*. De dicho congreso salió la declaración de París de OER y en ella, se da una definición de qué son los OER y se hacen una serie de recomendaciones a los Estados miembros de la UNESCO para la expansión de los OER.

Empecemos con la definición. Para la UNESCO, los OER son **“materiales de enseñanza, aprendizaje e investigación en cualquier soporte, digital o de otro tipo, que sean de dominio público o que hayan sido publicados con una licencia abierta que permita el acceso gratuito a esos materiales, así como su uso, adaptación y redistribución por otros sin ninguna restricción o con restricciones limitadas”** (UNESCO, 2012). Se puede observar que la mayor parte de la definición se dedica a discutir la licencia que deben tener asociados los OER. Esto es lógico si pensamos que después de todo esa es la gran innovación que suponen. Recursos educativos han existido siempre y la idea novedosa es que sean abiertos, es decir, que cuenten con una licencia que permita adaptar y redistribuir sin restricción.

Al menos desde la invención de la imprenta cada vez ha sido más fácil acceder a recursos educativos. Internet ha acelerado más aún el proceso de universalización de acceso al conocimiento. Pero los OER llegan para señalarnos que con acceder no basta. Si queremos poder disfrutar de todo el potencial de un recurso educativo, también se tiene que poder adaptar y redistribuir.

La discusión en torno a por qué es importante permitir adaptación y redistribución recuerda

a la que se produce en torno al software libre. En ambos casos, se entiende que los recursos (educativos o de software) son más útiles cuando pueden ser modificados con libertad por los usuarios finales, adaptándolos así a sus necesidades concretas y evitando la idea de que un enfoque único es el ideal para todas las personas. Es fácil imaginar es más difícil encontrar un recurso que se adapte perfectamente a mis necesidades que uno que pueda ser un punto de comienzo para crear uno que sí se adapte a lo que busco.

Pero como decíamos, con la capacidad de adaptación no basta. La de redistribución es igual de importante. Si yo puedo modificarlo, pero no puedo compartir esa modificación, estoy privando a los demás de la posibilidad de aprovechar esa modificación. Esta privación puede tener dos consecuencias: O bien hago que los demás tengan que repetir trabajo (el que sería el mejor de los casos), o bien hago que las demás personas no puedan acceder al recurso porque no dispongan de los medios para adaptar el recurso a sus necesidades.

En el caso de la educación, esta libertad de redistribución es especialmente importante, ya que toda la educación suele basarse en el modelo estudiante/profesor que se caracteriza hasta cierto punto porque el profesor tiene más conocimiento y experiencia que el estudiante. Si logramos que los recursos educativos recojan la libertad de distribución logramos que el profesor pueda adaptar los recursos educativos a sus alumnos (o incluso a otros alumnos). Si el profesor no puede redistribuir esa adaptación a sus estudiantes, de nada sirve.

Habiendo establecido qué son los OER ahora nos tocaría responder a cuál es su interés. Los OER nacen con la aspiración de hacer más accesible a todas las personas una educación de calidad. Existen, al menos, tres grandes colectivos que se pueden ver beneficiados por la Introducción masiva de OER: profesores, estudiantes y personas que quieren ser estudiantes, pero encuentras especiales dificultades.

Los beneficios para los profesores son los más obvios. Ahora mismo, ya es muy raro el profesor que crea el contenido de su curso desde cero. Es habitual y razonable que los profesores busquen libros u otros recursos didácticos que faciliten su labor de crear el contenido de su curso. No solo les ahorra tiempo, sino que además el profesor que busca qué han hecho otros profesores antes está incorporando en cierta manera la experiencia de dichos profesores, mejorando así su curso incluso antes de empezarlo.

Para las personas que ya están cursando estudios tener acceso a distintos y diversos OER que traten el mismo tema puede ser una oportunidad de mejorar su entendimiento de forma más rápida. Creo que todos hemos vivido la sensación de estar intentando entender un concepto al leer un libro o escuchando a una persona hablar y no ser capaces hasta que nos topamos con una explicación alternativa. Los OER se pueden ver como un conjunto inmenso de explicaciones alternativas justo para esos casos.

Por último, un OER puede ser muy útil para personas que no tienen acceso a una educación de calidad. Y tenemos que pensar en estas personas de una forma muy amplia. Pueden ser personas que no tengan instituciones educativas a su alcance porque viven en regiones sin un sistema educativo desarrollado. Pueden ser personas que no tengan los recursos económicos para asistir (por pobreza, por tasas de acceso excesivas o la mezcla de pobreza que obliga a trabajar y no deja tiempo para los estudios). Pueden ser personas con un ritmo de vida que no encajen en el sistema educativo que tienen más próximo, por cuestiones como personas dependientes a su cargo o su ritmo laboral. Podemos pensar también en personas que quieran renovar su aprendizaje años después de su paso por las instituciones educativas sin volver a recorrer todo el camino.

Como decíamos antes, la declaración de París, además de la definición, enumera una serie de

recomendaciones a los Estados miembros. De las diez recomendaciones, dos están especialmente relacionadas con este trabajo: las recomendaciones de “Alentar la investigación sobre los recursos educativos abiertos” y “Facilitar la búsqueda, la recuperación y el intercambio de recursos educativos abiertos.” Son relevantes porque este trabajo busca realizar la primera sobre la segunda, es decir, investigar sobre los métodos de búsqueda y recuperación de OER. En concreto, se va a presentar y discutir una arquitectura para el desarrollo de sistemas de recomendación de OER.

Hasta ahora hemos realizado un breve repaso a la motivación social del trabajo. En el resto del capítulo discutiremos, desde una perspectiva más técnica y científica, primero la motivación detrás del trabajo realizado y a continuación delimitaremos el alcance del trabajo para terminar enunciando cuál será la estructura del resto del documento.

I.1. Motivación

Hasta ahora hemos discutido la importancia social de la educación y el potencial que suponen los OER. En esta sección vamos a centrarnos en justificar qué podemos aportar, desde las tecnologías de la información y el conocimiento, a la búsqueda y recomendación de OER. Cuáles son los retos tecnológicos que se presentan y por qué son interesantes en sí mismos.

Empecemos con una reflexión filosófica clásica modificada: *si un OER cae en un repositorio y nadie está para utilizarlo, ¿hace algún sonido?*. Los OER al final son recursos educativos. Y como todo recurso educativo, necesitan de un estudiante para tener un sentido. Siguiendo con la analogía, queremos que los OER suenen. Queremos que sean utilizados. Para ello, es necesario no solo que se distribuyan, sino que sean encontrados por los usuarios finales. Hemos hablado ya de la importancia de licencias abiertas que permitan adaptar y redistribuir los recursos. Veamos ahora la otra cara de la misma moneda: la búsqueda de OER.

Encontrar el OER adecuado para cada usuario no es una cuestión sencilla, pero si no la resolvemos no sirve de nada. Imaginemos por un momento el reto sin introducir aún la informática. Supongamos que a un profesor de economía se le acerca una alumna y le dice que no ha entendido el concepto del interés compuesto y que quiere alguna alternativa. Haciendo memoria, el profesor podrá llegar a la conclusión de que conoce varios recursos que podría recomendar:

- Hace unas semanas, en su red social favorita, un contacto había compartido un vídeo que lo explica.
- Durante la carrera, leyó varios libros con capítulos dedicados al concepto.
- Está bastante seguro de que existe un artículo en Wikipedia sobre el interés compuesto.
- Una compañera de departamento le enseñó una página web que simula la evolución del dinero en un sistema con interés compuestos.

Todas estas opciones servirían para explicar el interés compuesto, pero posiblemente haya más. Ahí tenemos el primer reto: el profesor no puede recomendar a la alumna algo que no haya visto antes.

Pero el profesor, incluso partiendo de su conocimiento incompleto, puede hacer una buena recomendación a su alumna. Por ejemplo, si enumerara todas las opciones anteriores posiblemente estaría incluyendo una opción que satisficiera su pregunta. Pero es evidente que este no es el caso ideal. Soltar a la alumna un conjunto desordenado de recursos no es tan útil como una lista ordenada. En el caso del conjunto desordenado, la alumna tendrá que ir saltando de recurso en recurso hasta encontrar alguno con el que logre entender el concepto, pero así no garantizamos que la alumna llegue al mejor recurso posible, solo a uno lo suficientemente bueno (sea lo que sea que defina a un recurso como bueno). Además, la idea de que no es suficiente un conjunto desordenado se ve reforzado si pensamos lo que ocurriría si el conjunto fuera muy grande. Si el profesor le dice a la alumna “en la biblioteca encontrarás algo”, muy posiblemente esté en lo cierto, pero es como si no hubiera dicho nada. Segundo problema: un conjunto desordenado de recursos no es una respuesta suficiente.

Si el profesor quisiera guiar más a la alumna y dar esa lista ordenada ideal, encabezada por la mejor de todas las opciones, tendría que conocer muy bien a la alumna. Deberá conocer qué y qué no sabe para poder dar solo opciones que tenga unos requisitos previos más alineados con el conocimiento de la alumna. Deberá, así mismo, conocer los gustos personales de la alumna, ya que parece claro que no todos los alumnos aprenden igual. Puede que a esta alumna le sea más útil ver un vídeo que leer una explicación, o que prefiera algo interactivo a cualquiera de las dos anteriores. Tercer problema: hay que conocer a la alumna.

Es evidente que en el ejemplo anterior podríamos intentar introducir un sistema informático para realizar la labor que ha realizado el profesor. Idealmente, el profesor nunca debe ser sustituido sino complementado, pero no para toda persona que necesita un profesor hay uno disponible. Los tres problemas que hemos enumerado siguen estando igual: no se puede recomendar lo que el sistema no conoce antes, no basta con un conjunto desordenado y hay que conocer a la alumna para hacer una buena recomendación. Pero, además, a esos tres problemas tenemos que añadir nuevos derivados de la introducción de la tecnología.

Es complicado encontrar métricas que nos permitan caracterizar a los recursos y a los estudiantes de manera que podamos automatizar el proceso de emparejar a unos con otros. Algunas, como el idioma, son inmediatas y relativamente fáciles, pero mientras que a un profesor humano no se le ocurriría recomendar un texto en un idioma extranjero a su alumna, el sistema, en principio solo ve bits en un lado y el otro.

Otras cuestiones son más difíciles de cuantificar. Por ejemplo, los profesores pueden ver rápidamente qué conocimientos previos son requeridos para entender un OER y ya saben gracias a su relación con la alumna si tiene esos conocimientos. El sistema informático no tiene de forma natural esa jerarquía de conocimiento y tampoco sabe de forma directa qué sabe el usuario. Podría preguntar en alguna ocasión, pero mientras que los alumnos sí aguantan a profesores que hagan exámenes, parece dudoso que los usuarios acepten que el sistema haga un examen cada vez que buscan algo. Al sistema solo le queda inferir el conocimiento del alumno en base a las interacciones con los recursos recomendados o a una retroalimentación rápida y no repetitiva.

Otro ejemplo parecido al conocimiento sería el contexto social. Es algo que los profesores humanos suelen olvidar, porque en la mayoría de los casos se mueven en el mismo contexto social que los alumnos, lo que hace que los ejemplos y las explicaciones partan de una base común que permita el aprendizaje. Pero todo profesor que haya intentado explicarle algo a una persona de una cultura distinta sabe que es importante. Si quiero explicar un estilo arquitectónico en Buenos Aires con ejemplos de edificios de Madrid no seré tan efectivo como si uso ejemplos locales. Un profesor humano no suele cometer estos errores, pero, además, cuando lo hace puede darse cuenta

rápido. El sistema informático no tiene ni una cosa ni la otra: no conoce de contextos sociales y no puede detectar al usuario levantando una ceja o resoplando ante algo extraño.

Pero el caso que presentamos aquí no es toda la película. Hemos presentado el caso de una alumna buscando una recomendación, pero también puede darse el caso de que un profesor busque recomendaciones, o de adultos que busquen recursos por su propia iniciativa, sin estar dentro de un curso clásico y por lo tanto sin profesor. De esta variedad de roles, surgen también una variedad de objetivos. (Manouselis, Drachsler, Vuorikari, Hummel, y Koper, 2011) enumeran algunos de estos posibles objetivos con los que un usuario llega a un sistema de recomendación en el ámbito educativo:

- Recomendación de recursos mientras el usuario realiza una tarea (el ejemplo anterior).
- Encontrar recursos relevantes dado un filtro.
- Encontrar todos los recursos relevantes dado un filtro.
- Recomendar una secuencia de recursos.
- Recomendación de la nada, que sea novedosa. Aprender algo nuevo.
- Recomendar recursos recientemente creados.
- Encontrar otros usuarios con intereses similares.
- Recomendar caminos de aprendizaje alternativos (por ejemplo, cambiando el orden de estudio del temario).

Todos estos objetivos son hasta ahora tratados por sistemas distintos, cada uno especializado en una tarea. Sin embargo, los usuarios entienden el proceso de aprendizaje como un todo. El usuario tiene un objetivo final de aprender o enseñar y sería más útil un sistema que de forma integrada ofreciera solución a todos los retos.

En resumen, en esta sección, hemos explicado por qué es importante tener métodos para recomendar en el ámbito de los OER y a qué retos se enfrentan estos:

1. No se puede recomendar lo que no se conoce.
2. Un conjunto desordenado no es una respuesta suficiente. Es mejor una lista ordenada.
3. Para ordenar la lista es necesario conocer al usuario.
 - Algunas de las cuestiones a tener en cuenta son fáciles para un sistema informático, como el idioma de los recursos y del usuario.
 - Otras, sin embargo, no son tan fáciles. Los conocimientos previos o el contexto social deben ser inducidos por el sistema a través de las interacciones de los usuarios con los OER o de los propios OER.
4. Cada usuario puede tener un rol distinto en el proceso de enseñanza/aprendizaje.
5. Todos los usuarios tienen un objetivo final común (enseñar/aprender) pero lo traducen en objetivos concretos distintos.

I.2. Alcance y objetivos del trabajo

Hasta ahora, hemos presentado motivos por los que es interesante realizar investigación en educación y, más concretamente, en la búsqueda y recomendación de OER. En esta sección, vamos a detallar exactamente qué aportaciones busca realizar este trabajo, detallando el alcance de la propuesta y los objetivos planteados para el trabajo.

El diseño, la implementación y la evaluación de un sistema de recomendación de OER completo como el que se pretende concebir debe empezar por una reflexión previa sobre la arquitectura que el sistema final debería seguir. **La definición de esta arquitectura es el objetivo principal de este trabajo.**

Esa arquitectura debe estar inspirada y basada en propuestas o trabajos similares previos, de distintas áreas de conocimiento que se hayan aproximado a las distintas problemáticas que se plantean. Este trabajo, por tanto, se ha centrado en la definición de la arquitectura para la recomendación personalizada de OER tras el estudio de la literatura, con el objetivo de ser un punto de partida para trabajos futuros a desarrollar durante los estudios enfocados a una tesis doctoral.

La arquitectura propuesta, como cualquier arquitectura, está enfocada a definir una serie de componentes que se relacionan entre sí para lograr el objetivo final. En la definición de la arquitectura se detallarán las condiciones y características que tienen que cumplir cada uno de dichos componentes, pero en este trabajo **también se ha buscado detallar propuestas concretas** de cómo podrían ser dichos componentes **para una implementación.**

Por tanto, este trabajo pretende, tras un estudio de la literatura científica, hacer una propuesta de arquitectura para la recomendación personalizada de OER dirigida tanto a docentes como estudiantes. Esta arquitectura busca ir más allá de ser un simple buscador de OER y busca, a través del modelado de usuario, ser capaz de detectar los OER más adecuados para cada circunstancia, entendiendo que existen múltiples contextos y objetivos que deben convivir, aceptando para ello que pueden coexistir varios algoritmos de recomendación en el sistema.

Tras la definición de la arquitectura se realiza un repaso de técnicas y métodos que actualmente están siendo usados que podrían ser integrados en la implementación de la arquitectura. Dicha implementación ya escapa a los límites de este trabajo, aunque al final del mismo se detalla un ejemplo de aplicación de la arquitectura realizado como parte del trabajo de asignaturas del máster.

I.3. Estructura del documento

En la introducción hemos presentado la motivación detrás de este trabajo, primero desde una perspectiva social y después ya desde una perspectiva más técnica. A continuación, detallaremos la estructura del resto del documento.

El capítulo II repasa el estado del arte relacionado con este trabajo. En él primero se discutirá qué es y qué no es un OER, para a continuación detallar qué líneas de investigación están abiertas en el ámbito de los OER. Este trabajo no toca todas esas líneas abiertas, sino que se centra en una en concreto (la recomendación de OER) y por ello el capítulo termina con un repaso de algunas contribuciones relevantes en el área presentando una panorámica que recoge las aproximaciones que se han explorado.

Después del capítulo de estado del arte entramos ya en el capítulo III a lo que es el contenido novedoso de este trabajo: la arquitectura que se presenta para diseñar sistemas de recomendación de OER multipropósito y multicontexto. La arquitectura se detalla primero con unas guías generales que sirven para mostrar una panorámica antes de entrar a los detalles. Para explicar esos detalles primero se repasan algunos de los retos principales que hay detrás del diseño de la arquitectura, para que queden justificadas las decisiones de diseño. A continuación, se detallan ya las distintas etapas en las que se divide la arquitectura. Por último, el capítulo se cierra con una discusión sobre las bondades, así como las limitaciones de la arquitectura propuesta.

Para ayudar a ilustrar el potencial de la propuesta e ir sentando las bases de una futura implementación de la propuesta, en el capítulo IV se discuten propuestas sobre cómo podrían desarrollarse cada una de las etapas descritas para la arquitectura. Este capítulo cierra mostrando un ejemplo de aplicación de la arquitectura que se ha realizado como parte de varias asignaturas del máster.

Finalmente, el último últimos capítulos, V es el destinado al trabajo futuro y las conclusiones. Primero se repasan las contribuciones de este trabajo y finalmente se resume qué recorrido se prevé para esta línea de trabajo.

Estado del Arte

En este capítulo repasaremos la literatura científica más relevante por su relación con este trabajo. Primero, empezamos con cuestiones más genéricas relativas a los OER, empezando con el debate (no cerrado) en torno a qué y qué no es un OER para pasar después a ver las principales líneas de investigación abiertas en el área. A continuación, ya nos centramos en el tema en el que se centra este trabajo: la búsqueda y recomendación de OER. Repasaremos investigaciones recientes similares a esta propuesta que nos permiten contextualizarla y observar que el problema a tratar es un problema vivo en la comunidad científica.

II.1. Open Educational Resources

Recordemos que según (UNESCO, 2012), los OER son “materiales de enseñanza, aprendizaje e investigación en cualquier soporte, digital o de otro tipo, que sean de dominio público o que hayan sido publicados con una licencia abierta que permita el acceso gratuito a esos materiales, así como su uso, adaptación y redistribución por otros sin ninguna restricción o con restricciones limitadas.” Muchos artículos citan esta definición o definiciones muy similares, todas teniendo en común la idea de que son materiales con fines educativos y que deben ser distribuidos bajo licencias abiertas. Sin embargo, qué es y qué no es una licencia abierta no está tan claro y es por tanto nuestra primera parada.

II.1.1. ¿Qué es una licencia abierta?

Mientras que la parte de qué es un recurso educativo está libre de polémica (al menos en el área de los OER) la de qué es una licencia abierta genera más discusión y confusión. (Mishra, 2017) hace una discusión muy interesante al respecto. Uno de los principales errores es que mucha gente confunde cualquier recurso disponible online de forma gratuita con un recurso abierto. Y esto no es cierto, ya que la inmensa mayoría de recursos online están protegidos por el copyright que es, sin ninguna duda, una licencia no abierta.

Pero, ¿qué es una licencia abierta exactamente? No existe una única definición y esta es sin duda una de las causas principales de confusión. Vamos a citar tres de las definiciones principales.

Por un lado, tenemos la definición de (Möller, Stallman, Lessig, Hill, y Beesley, n.d.), que es la utilizada por Creative Commons (fuertemente inspirada en la del software libre de (Stallman, 2002), algo razonable ya que participa también en la de licencia libre). Según esta, una licencia libre tiene que respetar 4 libertades:

- La libertad de usar la obra (y ejecutar para recursos informáticos).
- La libertad de estudiar la obra y utilizar la información obtenida.
- La libertad de redistribuir copias.
- La libertad de distribuir obras derivadas.

Pero además de las 4 libertades también tiene que cumplir con 4 condiciones adicionales:

- Disponibilidad de los datos fuente (Por ejemplo, la partitura de una composición musical, los modelos usados en una escena 3D o los datos de una publicación científica).
- Uso de un formato libre, no sujeto a patentes, y si lo está, que conceda un permiso libre de regalías, ilimitado, irrevocable y de ámbito mundial.
- Sin restricciones técnicas que limiten nada de lo anterior.
- Sin otras restricciones o limitaciones, ya sean legales, contractuales o de cualquier otro tipo.

Esta definición de las 4 libertades es muy similar a otra de las más mencionadas en la literatura, la de (Wiley, 2010), que establece las “4R”: *Reuse*, *Revise*, *Remix*, *Redistribute*. Nótese que las dos primeras se corresponden con las de (Möller et al., n.d.), pero las dos últimas cambias de orden: *Remix* es la libertad de distribuir obras derivadas y *Redistribute* es la de redistribuir copias.

La otra definición nos viene de (Open Knowledge International, n.d.), y es algo más exhaustiva en qué criterios deben cumplir las licencias abiertas, ya que enumera 9 libertades que deben ser permitidas: Uso, redistribución, modificación, separación, compilación, no discriminación, propagación, aplicación para cualquier propósito y sin coste. Y otras 7 libertades del autor que no deben ser limitadas por terceros (y a las que por tanto el autor tiene derecho): Reconocimiento, integridad, compartir igual, aviso, fuente, prohibición de restricciones técnicas y no agresión.

Sea cuál sea la definición, todas comparten alguna versión de las “4R”, y es a la que nos vamos a referir a partir de ahora ya que las “4R” conforman una base común y se podría argumentar que el resto de término se derivan necesariamente de esos 4 puntos, siendo las definiciones más extensas sencillamente definiciones más explícitas.

II.1.2. Líneas de investigación abiertas en los OER

Una vez aclarado qué vamos a considerar como un OER centraremos nuestra atención en un rápido repaso sobre las líneas de investigación actualmente abiertas en las áreas relacionadas con los OER. Para ello, vamos a partir de una revisión de la literatura, realizada por (Wiley, Bliss, y McEwen, 2014). En esta revisión comienzan los autores también repasando qué es un OER y discutiendo qué licencias encajan con eso. Justo después ya entran en enumerar las cuatro principales líneas de investigación:

Modelos de compartir OER: Está claro que es necesario compartir los OER, pero no está muy claro cuál es la mejor manera de hacerlo. Hay debate en cuanto a los formatos y también hay aún más debate en cuánto a dónde y cómo recogerlos.

Algunos sistemas se centran en etiquetar los recursos con metadatos y almacenarlos en bases de datos o repositorios para que quedarse a disposición de terceras personas. Ejemplos de esto son OER Commons (OER Commons, n.d.) o MERLOT (MERLOT, n.d.).

Otros sistemas se centran en recolectar los OER como libros. Ejemplos son FlatWorldKnowledge (FlatWorld, n.d.) o CK12 (CK-12 Foundation, n.d.).

Otra opción igualmente extendida es la agrupación en forma de cursos. A estas iniciativas se las conoce como *Open Courseware (OCW)* y el ejemplo más conocido es la página del MIT OCW (MIT, n.d.), donde la conocida universidad publica OER producidos durante sus cursos.

Modelos de producción de OER: Anterior al debate de cómo distribuirlo está el debate de cómo producirlos. Aquí existen dos modelos enfrentados: el modelo de producción institucional y el modelo de producción pública.

El modelo de producción institucional es el utilizado por el MIT OCW y consiste en que profesores u otros profesionales respaldados por una institución son los que crean los recursos y son certificados por dicha institución. Este modelo en principio ofrece más garantías sobre el contenido de los OER pero resulta especialmente caro.

El otro modelo, el de producción pública, es el modelo de Wikipedia (Fundación Wikimedia, n.d.) donde cualquiera puede contribuir a la creación y supervisión de los OER. Este modelo es sin duda más barato, pero se pierde la garantía de tener la reputación de una institución detrás, a lo que los defensores argumentan que se suple con la participación de más gente en el proceso.

Existen algunos modelos híbridos entre ambos extremos, pero son los menos.

Beneficios de los OER: Existen estudios que han mirado el impacto de la introducción de los OER en cursos. Por ejemplo, aseguran que la existencia de OER hace más fácil y rápido la creación de nuevos cursos (Caudill, 2011) y es evidente que pueden ser una gran oportunidad de ahorro para los estudiantes, pero todavía se pueden recopilar más evidencias empíricas del impacto de la introducción de los OER en la educación.

Problemas de los OER: Lo último que los autores tratan son cuatro líneas de investigación que a su juicio están más verdes que las tres anteriores y que necesitan más recursos en el futuro:

El problema del descubrimiento cómo podemos hacer más fácil que los usuarios encuentren los OER que necesitan. Este es el problema principal en el que se centra este trabajo.

El problema de la sostenibilidad: cómo podemos asegurar la sostenibilidad económica de los OER. Ya hemos visto que depende del modelo de producción es más caro o más barato, pero en todos los casos hacen falta recursos para mantener los OER.

El problema de la calidad: si un usuario encuentra un OER, ¿cómo puede conocer si es un OER riguroso y veraz? Aún en el modelo de producción institucional (que en principio nos da esa garantía) tenemos el problema de que los OER necesitan ser mantenidos y revisados, ya que no todos envejecerán igual de bien.

El problema de la localización: el contexto sociocultural influye en el proceso de enseñanza-aprendizaje y esto se ve reflejado en los OER. Si recurso y estudiante no comparten un contexto, el recurso no es entendible y por lo tanto no es útil.

Curiosamente, la mayoría de estos puntos vienen recogidos también en la declaración de la (UNESCO, 2012). La declaración en su punto e. reclama estrategias para crear OER de calidad, en el g. reclama que se elaboren y adapten los recursos a múltiples idiomas y contextos y en el i. se pide que se facilite la búsqueda, recuperación y el intercambio de los recursos.

De estos retos, la mayoría tienen más que ver con políticas educativas que con cuestiones técnicas que podamos abarcar en este trabajo. El problema de la sostenibilidad dependerá de hasta qué punto gobiernos e instituciones estén dispuestos a destinar recursos a los OER como parte de sus estrategias educativas. El problema de la localización viene de que la mayoría de las instituciones con medios para crear los OER se encuentran en sociedades occidentales y especialmente en países de habla inglesa.

Pero el problema del descubrimiento sí que puede y debe ser resuelto (o al menos facilitado) desde la tecnología. Como ya hemos adelantado, la arquitectura que se presenta en este trabajo va especialmente dirigida a este punto. A continuación, veremos trabajos centrados en esta línea.

II.2. Trabajos relacionados

En la sección previa hemos repasado qué son los OER y las principales líneas abiertas de investigación en torno a ellos. En esta nos vamos a centrar en la que atañe más directamente a este trabajo: la búsqueda y recomendación de OER. Para ello, repasaremos varias publicaciones similares a este trabajo. Empezaremos con unas primeras investigaciones del 2010 para compararlas después con publicaciones más recientes y ver la evolución.

II.2.1. 1st Workshop on Recommender Systems for Technology Enhanced Learning - 2010

En el 2010 se celebró un primer workshop sobre sistemas de recomendación en el área de recursos educativos (Manouselis, Drachsler, Verbert, y Santos, 2010) que vamos a tomar como punto de partida de las investigaciones a revisar en este trabajo, ya que de él podemos sacar una buena panorámica de qué retos y soluciones había.

De las 15 contribuciones que se presentaron en el workshop, se han estudiado 6 en más profundidad, eligiendo de entre el total según proximidad a la temática del trabajo o por ser representativas de otras contribuciones similares. De entre todas ellas, lo primero que salta a la vista es la predominancia del uso de técnicas de filtrado colaborativo. En concreto, las investigaciones de (Brusilovsky et al., 2010), (Mödritscher, 2010), (Shelton, Duffin, Wang, y Ball, 2010) y (Sicilia, García-Barriocanal, Sánchez-Alonso, y Cechinel, 2010) hacen referencia directa a estas técnicas.

En (Brusilovsky et al., 2010) se propone un sistema de navegación social (que se puede entender hasta cierto punto como una forma de filtrado colaborativo) para un repositorio de recursos educativos abiertos de temática relacionada con la informática. En este sistema, lo que crean son recomendaciones de navegación por la página. A los usuarios, según los datos

analizados previamente, se les muestran los listados de los recursos ordenados según su perfil. Estos recursos, son los que ofrece la propia página más recursos derivados que los usuarios pueden crear mediante dos herramientas distintas: por un lado, pueden seleccionar y anotar partes de los recursos desde el propio sistema y esas secciones anotadas se guardan como nuevos recursos; y por otro, permiten a los usuarios juntar varios recursos en un colección ordenada y anotada, que llaman caminos. Con la herramienta de anotación se puede dividir un recurso en varios y con la de los caminos se agrupan recursos. Partiendo de esta base de recursos, se analiza la actividad de navegación de los usuarios por la página y la retroalimentación explícita de los usuarios sobre los recursos (comentarios, puntuaciones y tags) y con ello eligen el orden con el que se muestra al usuario los recursos en cada página.

(Mödritscher, 2010) propone un sistema de recomendación de *personal learning environments* (PLEs), entendiéndolo por un conjunto de herramientas, servicios y artefactos de aprendizaje recolectados de varias fuentes para ser utilizados por estudiantes. A estos artefactos y herramientas podríamos referirnos hoy en día como OER, porque recordemos que el término no estaba extendido en el 2010. El sistema que propone, utilizando una extensión del navegador web, permite a los usuarios crear una colección jerárquica de enlaces a herramientas, servicios y demás que los usuarios pueden catalogar por contextos (como trabajo, casa...) y por actividad. Repasa a continuación muchas técnicas posibles para la creación de recomendaciones una vez recolectada esa información, como PageRank, clustering o NLP, pero no llega a concretar ninguna de ellas.

Como ocurría con el primer ejemplo, (Shelton et al., 2010) presenta un sistema ya desarrollado e implementado completamente. Este sistema, Folksemantic, busca crear una comunidad de recomendación de OER y parte de la integración de dos proyectos anteriores: OCWFinder, que permite realizar búsquedas de recursos abiertos mediante tags y OER Recommender, que, dado un recurso, recomienda otros similares (Duffin y Muramatsu, 2008). Tras juntar estos dos sistemas, añadieron al sistema capacidades de red social. Permitieron a los usuarios crear cuentas para tener perfiles y comentar/puntuar otros, así como añadir sus propios recursos. Sobre esta información, ofrecen una plataforma de búsqueda de recursos basada en modelos vectoriales y que mejora los resultados analizando los patrones de comportamiento de los usuarios por la página. A diferencia de (Brusilovsky et al., 2010), esta propuesta sí trabaja con varios repositorios, pero no es muy explícita en cuánto a cuáles o a cómo afina la búsqueda. El sistema a día de hoy no puede ser encontrado y la página web tanto de Folksemantic, como de los dos sistemas previos, están caídas.

(Sicilia et al., 2010) presenta una evaluación del desempeño de algoritmos de filtrado colaborativo aplicándolo sobre un conjunto de OER extraídos de (MERLOT, n.d.), un repositorio de metadatos de OER. Antes de entrar a la evaluación en sí, hace un par de apuntes muy interesantes. Por un lado, destaca que, por las características del problema de recomendación en entornos educativos, parece importante aplicar algoritmos de filtrado colaborativo multiatributo. Mientras que en otros sectores todos los usuarios buscan un mismo fin, no es el caso en educación, donde existen muy distintos conjuntos de usuarios con intereses distintos. Un sistema de filtrado colaborativo que no haga esas distinciones (por ejemplo, por niveles educativos, por rol o por idioma) posiblemente realizará recomendaciones muy erráticas. Ya entrando en la evaluación, utiliza la versión del filtrado colaborativo basada en usuarios y prueba distintas métricas de similitud, utilizando como input las puntuaciones otorgadas por los usuarios para cada recurso. Como conclusión, destacan que los sesgos de las puntuaciones (por ejemplo, las puntuaciones son generalmente muy elevadas y escasas) hacen que el error absoluto sea relativamente alto.

Una aproximación que ya sale del filtrado colaborativo es la de (Santos y Boticario, 2010), que en su lugar deciden aplicar un sistema de recomendación basado en reglas. Justifican esta decisión

en el hecho de que los profesores tienen una clara preferencia por los sistemas de recomendación que pueden justificar de una forma clara las recomendaciones. ¿Y por qué dan tanta importancia a los profesores? Principalmente porque entienden que el proceso educativo no puede basarse en las preferencias de los estudiantes como los sistemas de recomendación en otros ámbitos, sino que tienen que tener un input directo de profesionales con experiencia en educación que permitan crear caminos pedagógicamente lógicos. Esta es una reflexión interesante, ya que destacan que es muy posible que los estudiantes tengan otros incentivos para elegir los recursos que los puramente didácticos. Así mismo, entienden que tener input constante de profesores es muy costoso y aclaran que esperan recolectar datos suficientes como para finalmente lograr algún tipo de abstracción que vaya reduciendo esa intervención.

Todas las propuestas anteriores tratan de una forma u otra de la recomendación de recursos, pero, salvo la de (Brusilovsky et al., 2010), no realizan ningún tipo de transformación en la representación de los recursos. (Wang y Sumiya, 2010) sí que lo hacen, siendo especialmente novedosos en ese sentido. En concreto, se centran en la recomendación de diapositivas. Postulan que cada vez es más habitual encontrar disponibles las diapositivas de cursos o charlas y que ofrecen una gran oportunidad para el aprendizaje. Lo que proponen es un buscador que compara el texto de entrada del usuario con un modelo jerárquico que crean con los conceptos que aparecen en cada diapositiva para cada fichero. Explican que como el texto de las diapositivas suele ser esquemático y jerárquico, analizando la indentación y tamaño de cada frase pueden hacer un árbol jerárquico que refleje de forma fidedigna contenido de la presentación.

II.2.2. Trabajos más recientes (2017-2018)

En lo que queda de capítulo repasaremos cinco publicaciones más recientes relacionadas con la temática de búsqueda, recomendación o cuestiones afines que pueden influir en este proceso. De los cinco, cuatro se refieren a propuestas más concretas y empezaremos con ellos. El último, presenta una arquitectura para la recomendación de OER, así que por su elevada similitud con el trabajo actual lo dejaremos para la siguiente sección.

Como decíamos, algunos de los artículos que vienen a continuación se centran más en cuestiones afines que en la propia recomendación. Uno de ellos es el trabajo de (Gasparetti, De Medio, Limongelli, Sciarrone, y Temperini, 2018), que propone utilizar técnicas de aprendizaje automático para determinar si un recurso tiene prerequisites previos. La cuestión de los prerequisites es importante porque es fácil imaginar que un recurso puede ser perfecto para un estudiante en todas las métricas, pero aun así no ser el adecuado porque requiere para su aprovechamiento de algún concepto del que el estudiante todavía no dispone. Un sistema de recomendación que incluya la detección automática de prerequisites será más robusto, ya que podrá recomendar un OER y otros OER que complementen al primero respecto a los prerequisites no cumplidos por el usuario. El procedimiento que proponen empieza aplicando técnicas de procesamiento de lenguaje natural para tokenizar los recursos en listados de términos. Después, los términos de ese listado los enlaza con sus artículos de Wikipedia que utiliza como una ontología débil, ya que se fija en la categoría en la que están asignados esos artículos en la propia Wikipedia. El listado de términos del recurso, el contenido de la página de la Wikipedia, las páginas que esta enlaza y las categorías que tengan asignadas son la entrada final del algoritmo de clasificación, que va recibiendo pares de OER y determina si el primero responde a un prerequisite del segundo.

El otro artículo en la línea de mejorar la información disponible sobre los recursos es el de

(Hanna, Abhari, y Ferworn, 2017). En él, se presenta un método para introducir análisis de emociones en textos al proceso de valoración de los OER. En concreto, lo que hace es crear una base de datos de recursos de MERLOT que disponen de comentarios y de puntuaciones por parte de los usuarios. En base a las puntuaciones, clasifica a los recursos según si son bien valorados o mal valorados. Partiendo de esa clasificación, analiza los comentarios para extraer qué palabras aparecen de forma desproporcionada en ambas categorías, suponiendo que se podrán usar posteriormente como indicadores de la calidad de un recurso sin puntuación, pero con comentarios. Como ya avisaban en (Sicilia et al., 2010), las puntuaciones de MERLOT tienen un sesgo muy importante, que también mencionan como limitación, junto al hecho de que los recursos peor valorados tienden a tener muy pocos comentarios, del orden de 1 o 2, lo que tampoco permite extraer mucha información. Además, las palabras que finalmente extrae como indicadores de calidad son palabras como *useful*, *great*, *bad* o *useless*, que ya sabíamos que son palabras con esa carga semántica.

(Christudas, Kirubakaran, y Thangaiyah, 2018) utilizan algoritmos genéticos para realizar recomendaciones de OER a alumnos. Para ello, definen y combinan tres métricas distintas:

- Nivel de compatibilidad del recurso respecto al estilo de aprendizaje del estudiante de acuerdo con el modelo de (Felder y Silverman, 1988).
- El nivel de complejidad del recurso educativo comparado con el nivel de conocimiento del estudiante.
- La afinidad entre el nivel de interactividad del recurso y del usuario. Para el usuario de nuevo utiliza el test de Felder y Silverman, mientras que para el recurso define una métrica de 0 a 4, donde 0 es para recursos no interactivos y el 4 para recursos que permiten interacción en tiempo real.

Una limitación de este estudio es que todos los valores de los recursos en cada una de las métricas tienen que ser definidos previamente por el equipo docente del curso. Esto nos lleva a un problema común a todos los sistemas adaptativos: el esfuerzo que supone para los autores del contenido introducir en el sistema los recursos a adaptar y las métricas que los definen a ojos del sistema. Métodos automáticos de extracción de características reducen esta carga de trabajo y son indispensables cuando hablamos de aplicaciones con gran número de recursos, como las que trabajan con grandes repositorios de OER.

Por último, (Hajri, Bourda, y Popineau, 2017) presentan un sistema de recomendación destinado específicamente para recomendar OER dentro de MOOCs. Construyen varios modelos distintos que interactúan para hacer la recomendación:

- Un modelo de usuario teniendo en cuenta qué conoce, su estilo de aprendizaje (utilizando, de nuevo, el modelo de (Felder y Silverman, 1988)) y los idiomas que habla.
- Un modelo del MOOC, proporcionado directamente por el profesor del curso durante la creación del curso, que recoge los objetivos didácticos de cada unidad del curso y los prerrequisitos, así como el nivel de dificultad (a elegir entre principiante, intermedio y avanzado).

Con estos modelos, partiendo de una base de recursos, va aplicando distintos filtros en cadena para ir seleccionando los recursos a recomendar para cada usuario. Primero eligen los recursos más

próximos a la búsqueda utilizando tf-idf (Salton, Fox, y Wu, 1983). A continuación, reordenan según los modelos de usuario y del MOOC. Los autores no especifican la fuente de los recursos ni tampoco cómo adaptan exactamente los recursos según el modelo de usuario. Parece que es una investigación en un estado temprano.

II.2.3. *Recommendation of Open Educational Resources. An approach based on Linked Open Data.*

Como adelantábamos antes, el trabajo de (Chicaiza, Piedra, Lopez-Vargas, y Tovar-Caro, 2017) ocupa un lugar especial en este capítulo dada su estrecha relación con el presente trabajo. Los autores del artículo también definen una arquitectura para la recomendación de OER. La mayor diferencia viene del hecho de que la arquitectura de Chizaiza está orientada a la recomendación a través del uso de *linked data*, o datos enlazados, mientras que nuestra arquitectura contempla varios métodos de recomendación.

Los autores empiezan definiendo una serie de “decisiones tecnológicas” que sirven para concretar la arquitectura. La principal, como decíamos, es que el filtrado de los recursos se basa en datos enlazados. Proponen un sistema de filtrado multidimensional pensado para trabajar en escenarios heterogéneos, que es la situación de los repositorios de OER. Esta decisión supone que todos los modelos deben basarse en vocabularios definidos de acuerdo a los principios de las tecnologías de datos enlazados. Así mismo, deciden crear una arquitectura lo suficientemente modular como para permitir la integración sencilla con otras herramientas, o la sustitución de cualquier componente por alguna de estas herramientas ya existentes.

La arquitectura que proponen trabaja con tres modelos: el que describe a los usuarios, el que representa a los OER y una representación esquemática intermedia que conecta a unos con otros. Y con estos tres modelos se divide el proceso en cuatro etapas:

Gestión de datos: Esta etapa comienza recopilando metadatos desde distintos repositorios y otras fuentes disponibles en la web y utiliza estos metadatos junto a vocabularios abiertos para genera los modelos de los OER, que finalmente almacena en un repositorio local.

Enriquecimiento: Se completan los modelos de los OER y de los usuarios utilizando datos enlazados públicos y fuentes sociales.

Explotación: La fase genérica donde se implementa un mecanismo de procesamiento de conocimiento con algún fin. En el caso que los autores proponen en el artículo se centran en el filtrado de los OER y para ello proponen utilizar lenguajes de búsqueda sobre datos enlazados como SPARQL.

Retroalimentación: Sección transversal a todo el proceso en la que se implementan distintos métodos de inferencia y derivación semántica sobre los datos de interacción del usuario con el sistema.

En el capítulo siguiente introducimos ya la arquitectura que proponemos en este trabajo y se discutirán las diferencias y parecidos con la aquí presentada.

Arquitectura propuesta

Hasta ahora, en capítulos anteriores hemos presentado la motivación general detrás del trabajo y hemos repasado las líneas abiertas principales en el área de OER, junto a investigaciones relevantes por su similitud con esta. En este capítulo, entramos ya con la arquitectura propuesta para la recomendación de OER. Primero, haremos una descripción general de la arquitectura que se propone. A continuación, discutiremos los retos que se derivan de tratar con OER para recomendación, ya que estos sirven para justificar las decisiones tomadas en la creación de la arquitectura. A continuación, entraremos ya a un repaso sistemático y exhaustivo por todas las etapas de la arquitectura. El capítulo lo cierra una discusión sobre las fortalezas y debilidades de la arquitectura.

III.1. Descripción general

En la introducción adelantábamos que el objetivo principal de este trabajo es detallar una arquitectura para la recomendación de OER. Decíamos que tenía que ser una arquitectura que fuera capaz que trabajar con OER de distintas fuentes (repositorios institucionales, públicos, páginas web generales...), con OER en distintos formatos (vídeo, audio, texto, simulaciones, explorables...) y que sirviera para hacer recomendaciones en distintos contextos y a muy distintos usuarios con distintos objetivos. Las recomendaciones tenían que ser, por tanto, personalizadas.

Vamos a ampliar un poco esta definición con algunos puntos más sobre esta definición. El primero, es que la arquitectura tiene que ser modular, para permitir la integración de componentes de distintos orígenes o la sustitución de componentes con la llegada de nuevas tecnologías.

El segundo, es que la arquitectura debe suponer lo menos posible sobre qué acompaña al OER. Algunos sistemas de recomendación de OER se basan en datos enlazados, comentarios de usuarios o validación de expertos, pero no tenemos certeza de que para todos los OER existan esta información asociada. La única certeza que tenemos es que al haber un OER hay contenido didáctico en ese OER. Toda información posterior con la que podamos trabajar es bienvenida y reforzará el sistema, pero debe ser accesoría.

El tercero tiene que ver con la diversidad de contextos, y más concretamente, con la diversidad de objetivos. No todos los usuarios buscan lo mismo de un recomendador de OER y puede

que sus necesidades cambien con el tiempo. Por ejemplo, un profesor podría querer encontrar recursos similares a uno dado, mientras que un alumno podría buscar recursos según temática. La arquitectura busca reconocer esta diversidad de objetivos y por eso contempla la posibilidad de que haya métodos distintos de explotar la información recolectada y derivada de analizar los OER. Idealmente, esos métodos coexistirán en un solo sistema, que se podrá ir construyendo método a método.

El cuarto y último punto, es que todo el proceso debe ser retroalimentado. Es imposible acertar a la primera y la arquitectura debe reflejar y contemplar sistemas de retroalimentación.

Partiendo de todo ello, llegamos a la propia arquitectura. En lo que queda de sección la introduciremos, reservando secciones posteriores para detallar y justificar los detalles más finos. La arquitectura que proponemos se compone de 5 etapas, que se recogen en la figura III.1 y en la siguiente lista:

Etapa 0: Recolección de OER desde distintas fuentes.

Etapa 1: Transformación de los OER a una representación común.

Etapa 2: Creación de los modelos de recurso y usuario.

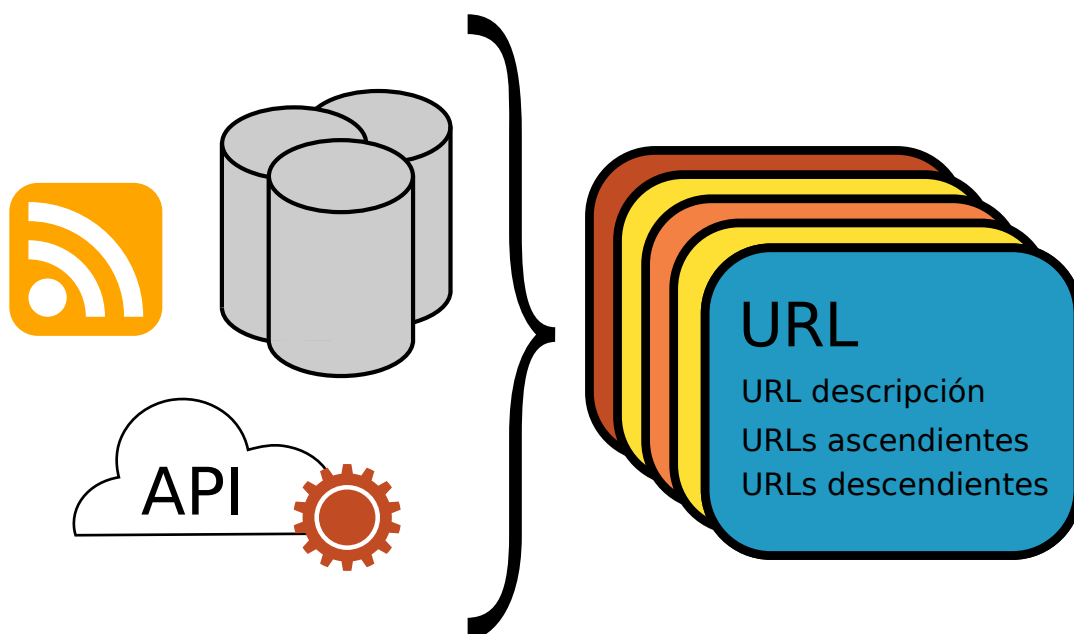
Etapa 3: Explotación de los datos.

Etapa 4: Retroalimentación del proceso.

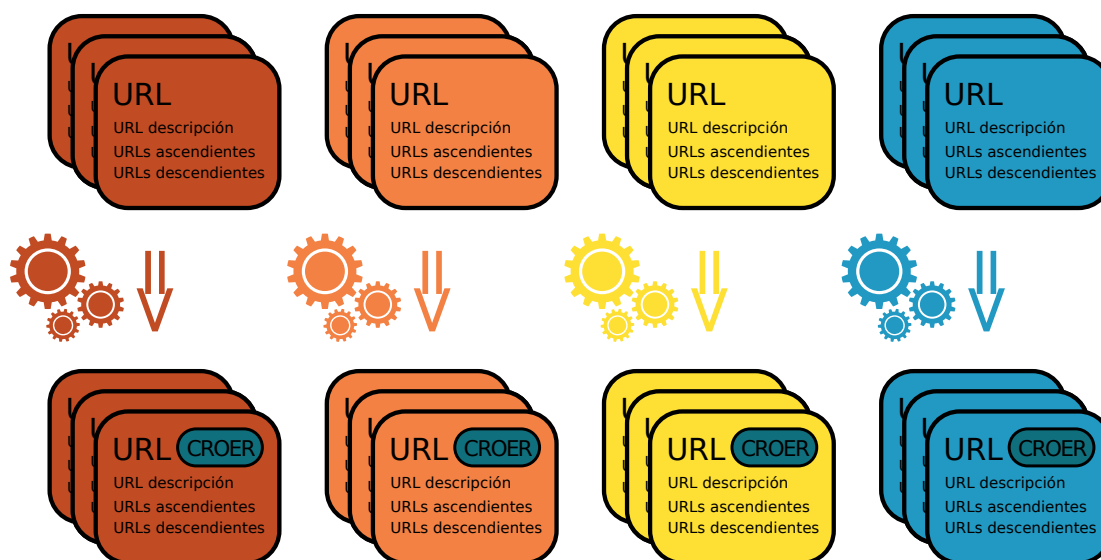
El proceso empieza en la **Etapa 0**, que es la etapa encargada de recopilar todos los OER disponibles en las fuentes que se estén considerando. En toda la arquitectura se identifican los OER de forma inequívoca por su URL y es este el único requisito que debe cumplir software que implemente esta etapa: debe devolver un listado de URL. Opcionalmente, el proceso puede asociar a esta URL más información. En concreto, puede especificarse una URL de descripción, que podría ser, por ejemplo, la URL de la página con la descripción en el repositorio o un enlace al fichero de datos enlazados. Así mismo, opcionalmente se pueden especificar OER ascendientes y descendentes, es decir, OER que incluyan al presente y que sean incluidos por el presente, respectivamente. Con estas dos últimas opciones logramos tratar de la misma forma a cursos que se componen de libros y estos a su vez de capítulos, por ejemplo.

Partiendo de este listado de URLs (con o sin información opcional asociada), pasamos a la **Etapa 1**. En esta etapa, todos los OER son transformados a representaciones comunes. Esta representación común la hemos llamado CROER (*Common Representation of an Open Educational Resource*) y responde a la idea de que los algoritmos de recomendación que vienen después necesariamente transformarán los OER a alguna representación propia (textual, datos enlazados...). Estas representaciones pueden ser compartidas por varios algoritmos de recomendación y para facilitar eso se distinguen los CROER. Además, esta etapa es fundamental para poder tratar todos los OER, sin importar su formato original, de la misma forma. Al lograr abstraer así el formato, logramos que los sistemas de recomendación que vienen en etapas posteriores se abstraigan de la representación del contenido y se centren principalmente en este. Cada formato original tendrá un transformador distinto y lógicamente la complejidad de estos variará enormemente según el formato original.

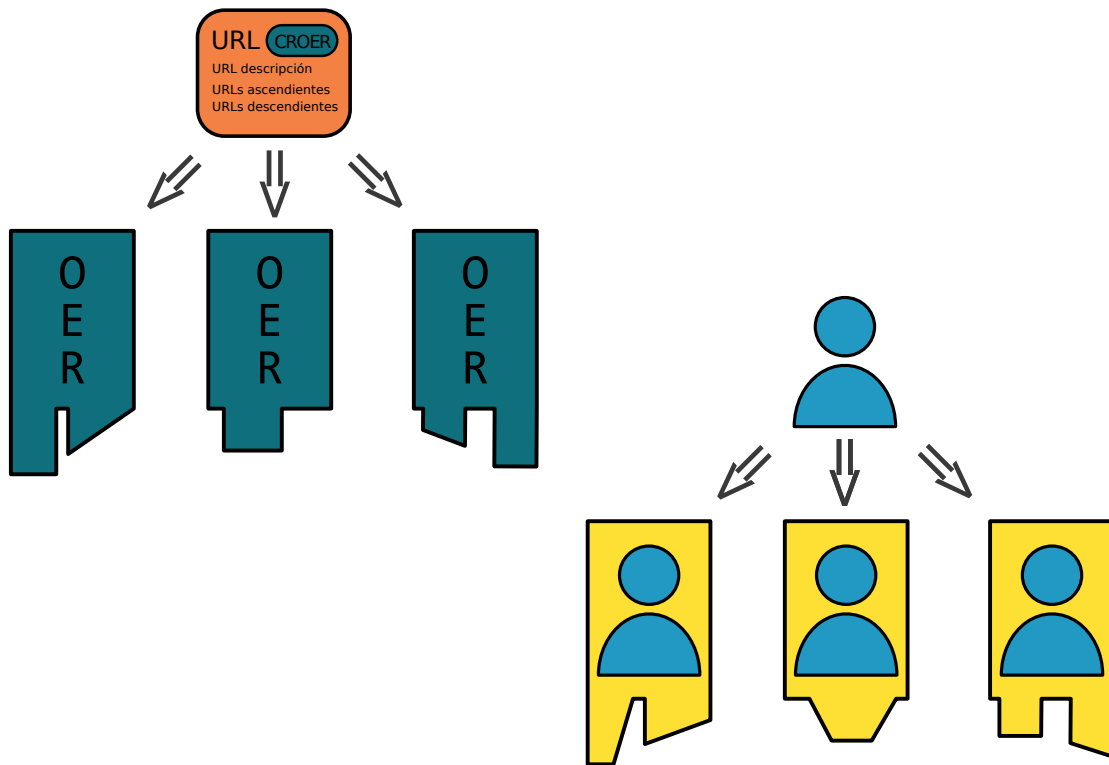
La siguiente etapa, la **Etapa 2**, es la etapa de creación de los modelos que serán posteriormente utilizados en la fase de explotación. La necesidad de esta etapa es que tendremos mucha información para cada entidad del sistema (usuarios y recursos) pero no todos los modelos utilizarán toda la información. De los usuarios se recogen sus preferencias, deseos, historia y demás, mientras que para cada recurso podrían existir varias CROER (en formato textual,



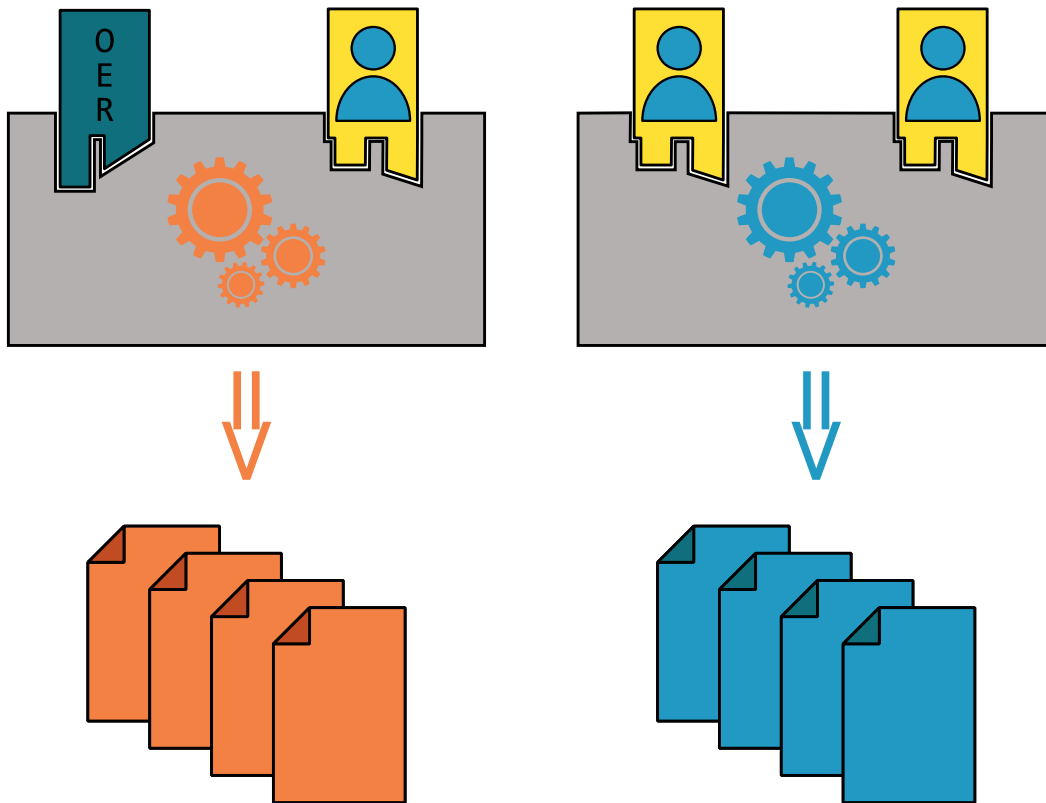
(a) Etapa 0: Recolección de OER desde distintas fuentes.



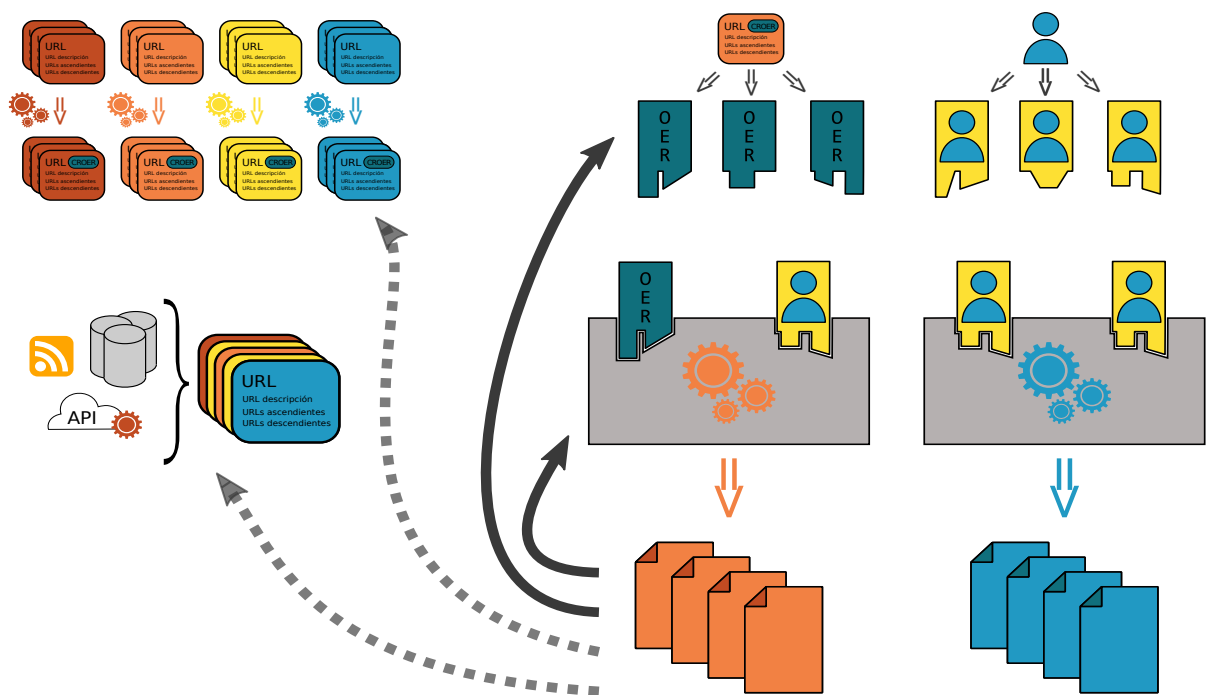
(b) Etapa 1: Transformación de los OER a una representación común.



(c) Etapa 2: Creación de los modelos de recurso y usuario.



(d) Etapa 3: Explotación de los datos.



(e) Etapa 4: Retroalimentación del proceso.

Figura III.1: Etapas de la arquitectura

datos enlazados...). En esta etapa se crean “vistas” de estos grandes conjuntos de datos para ser procesados por los sistemas de recomendación posteriores.

Estos modelos se combinan en la **Etapa 3** para producir las recomendaciones. Esta es la fase de explotación y en ella múltiples sistemas coexisten, respondiendo así a la multiplicidad de objetivos que mencionábamos antes. Algunos algoritmos de recomendación asociarán recursos entre sí, otros recursos con usuarios, o incluso podrían asociarse usuarios con usuarios. Estos algoritmos de recomendación podrían trabajar con los modelos de recurso y usuario de la etapa anterior o con los resultados procesados de otros modelos, permitiendo así que un algoritmo de recomendación construya sobre algoritmos previos, siguiendo una estrategia de divide y vencerás.

Por último, la **Etapa 4** es la destinada a recoger, analizar y propagar la información necesaria para introducir en el sistema retroalimentación. Esta información será propagada principalmente a las etapas 2 y 3, ya que estas son las etapas que más se prestan a implementar métodos automáticos para tener en cuenta dicha retroalimentación. Aun así, podrían existir algoritmos de recomendación que ignoraran esta retroalimentación. La propagación de la retroalimentación a las etapas 0 y 1 sería también posible, pero mucho más improbable que pudiera realizarse sin necesidad de intervención humana, pues previsiblemente serán etapas más estables para las que no merecería la pena realizar un sistema de actualización automático.

En resumen, partiendo de una serie de fuentes de OER, recopilamos las URL de estos, para a continuación transformar los OER a una serie de representaciones comunes que puedan ser aprovechadas por varios modelos posteriores. Desde estas representaciones comunes, creamos tantos modelos de recurso como sean necesarios para los recomendadores, que, junto a los modelos de usuario, los utilizan para crear las recomendaciones. Finalmente, se realiza una evaluación del proceso que genera una retroalimentación que se propaga a las etapas anteriores para ir mejorando el proceso.

III.2. Retos a superar

Tras la presentación de la arquitectura en la sección anterior con la que hemos mostrado una vista panorámica, ahora repasaremos los retos principales a los que se enfrenta la propuesta que justifican muchas de las decisiones tomadas en el diseño de la arquitectura.

III.2.1. Heterogeneidad de fuentes

Parece que casi existen tantos repositorios de OER como OER. Hay repositorios gestionados por instituciones educativas, por organizaciones no gubernamentales o repositorios gestionados por gobiernos. Existen algunos donde solo ciertas personas o colectivos pueden subir material, mientras que en otros cualquiera puede hacerlo. Los hay con gestores o bibliotecarios que revisan todo el contenido y los hay sin ellos. Los hay temáticos y los hay de muchos temas distintos. Los hay que son principalmente productores y los hay que son recopiladores. Hay tantos, que los repositorios de repositorios también son habituales.

La inmensa mayoría de ellos permiten acceder de forma automática a los recursos, y algunos incluso ofrecen API pública para acceder a ellos o sus servicios, como buscadores. Algunos pocos, irónicamente, recopilan recursos abiertos, pero no permiten el acceso de forma automática a ellos por medios informáticos (un ejemplo de esto sería OER Commons, que lo prohíbe en sus

condiciones de servicio (OER Commons, 2017)), por lo que no todos los repositorios sirven como fuente de datos.

Así mismo, la forma de presentar los OER es muy diversa. Lo más habitual es que haya algún tipo de listado, bien como resultado de una búsqueda, bien por una organización del contenido en categorías, del que cuelguen páginas específicas para cada recurso. Estas páginas tienden a tener un conjunto de campos en común (título, fecha, licencia...), pero otros muchos campos peculiares de cada página, o con definiciones propias que no permiten una comparación directa. Por ejemplo, es muy habitual que haya un campo que guíe hacia cuál es el público objetivo del recurso, pero algunos están expresados en años, en niveles educativos (que cambian de un país a otro) o en algún tipo de etiqueta que intente representar dificultad.

Lo que sí es común a todos los OER y repositorios, sin importar ninguna otra consideración, es la asignación de una URL para identificar el recurso. Como todos los recursos tienen necesariamente una URL asociada, consideramos esta como lo que designa al recurso en la etapa 0. Aunque esta designación no es inequívoca, ya que un solo recurso puede tener varias URL distintas, designaciones más complejas que resuelvan este problema resultan demasiado costosas.

III.2.2. Heterogeneidad de formatos

Una de las grandes oportunidades que ofrecen los OER es que para cada tema que se busque, se podrán encontrar recursos muy diversos tratando ese mismo punto. No solo ya desde distintos enfoques, sino en formatos completamente distintos. Podemos encontrar vídeos, libros, presentaciones, charlas grabadas, simulaciones software... Esta diversidad es maravillosa porque a pesar de que hay bastantes evidencias de que cada estudiante tiene unas preferencias o estilos de aprendizaje distintos, es muy raro que un docente tenga los recursos materiales y temporales como para crear un curso en el que cada recurso está en varios formatos.

Pero esta diversidad de formatos complica mucho la recomendación, ya que cada formato viene con sus peculiaridades. Por ejemplo, un recomendador solo de libros creará un modelo que incluya información sobre la editorial, la fecha de edición o el número de páginas, métricas que no tienen ningún sentido hablando de vídeos, por ejemplo. Y no es solo difícil a la hora de construir los modelos, sino que también es complicado a la hora de trabajar con los ficheros de datos originales. El software necesario para tratar o analizar ficheros de vídeo es muy distinto al necesario para hacer lo mismo con una página web.

Para solucionar esto se hace la propuesta de introducir la idea de representaciones comunes de los OER, lo que se ha llamado CROER (*Common Representation of an OER*). Este paso intermedio permite desacoplar la etapa de análisis del recurso para la extracción de características de la etapa de construcción del algoritmo de recomendación, logrando así romper una dependencia que dificulta el desarrollo de soluciones en esta área. Un recomendador de OER a día de hoy tiene que elegir qué formatos de recurso acepta y desarrollar analizadores para ellos, dejando fuera a todos los demás, presentes y futuros. Con esta aproximación, todo recomendador que trabaje con, por ejemplo, una representación textual del contenido del recurso puede dar soporte inmediatamente a todo formato que tenga ya algún sistema de traducción a texto implementado.

III.2.3. Heterogeneidad de estructura lógica

El concepto de qué es un recurso educativo es muy amplio y está normalmente muy condicionado por las características del modelo educativo en el que se desenvuelven las personas, así como sus circunstancias personales. Un ejemplo (un tanto extremo pero ilustrativo) es que para una persona que está aprendiendo a leer un catálogo de publicidad puede ser un gran recurso educativo porque asocia imágenes de productos que conoce con texto y así le permite saber si lo que ha leído tiene sentido, mientras para la mayoría de nosotros esos catálogos no son más que publicidad no deseada que viajan del buzón a la basura sin ninguna posibilidad de redención.

Volviendo a casos más normales, no es raro que los OER estén organizados en cursos, libros o recopilaciones, que son a su vez OER igual de válidos. Algunos usuarios solo necesitarán leer un capítulo, pero otros el libro entero y por eso tiene sentido esta estructura jerárquica variable. El problema es que no hay ningún tipo de estándar que regule de ninguna forma esto (y no parece que vaya a ser creado y masivamente respetado). Así mismo, al permitir los OER la modificación y redistribución de las modificaciones, es muy probable que los recursos sean fraccionados en otros recursos.

Mantener toda esta información disponible para el proceso de recomendación puede ser útil. Esta información se recoge de forma natural en la etapa 0, la de recolección de recursos, pero pasa prácticamente inalterada hasta la etapa 3. Además, como decíamos, al no existir una estructura clara, no es posible crear un modelo jerárquico estándar. Por todo ello, en la arquitectura se contempla que los recursos puedan tener listados de recursos ascendientes (es decir, de los que el recurso presente es parte) y recursos descendientes (es decir, de los que el recurso presente es el origen).

III.2.4. Diversidad de objetivos

Ya decíamos en la descripción general de la arquitectura que uno de los objetivos principales en el desarrollo de la misma era contemplar la diversidad de contextos, y más concretamente, con la de objetivos. No todos los usuarios buscan lo mismo de un recomendador de OER y puede que sus necesidades cambien con el tiempo.

Hacer un sistema que cumpla con todas las necesidades de un colectivo tan diverso (alumnos, profesores y de todo el mundo) a la primera es una empresa inabarcable, así que muchas investigaciones y desarrollos previos se han concentrado en pequeñas partes del problema global. Algunas han sido ampliadas posteriormente o integradas, pero son las menos. Tener 1.000 sistemas distintos para encontrar OER lejos de resolver el problema del descubrimiento lo agrava, pues además hay que encontrar el buscador adecuado.

Es por todo esto que en esta propuesta se contempla desde un principio en la etapa de explotación la multiplicidad de algoritmos de recomendación. Se entiende que una estrategia de divide y vencerás es la única viable, además se ser la que permite una mayor sinergia entre diferentes sistemas desarrollados de forma independiente, ya que con todos los desacoplamientos entre componentes que contempla la arquitectura se permite que cualquier esfuerzo previo se pueda integrar en un sistema cada vez mayor y más completo.

Es además esperable que esta integración de múltiples modelos en un solo sistema permita que surjan sinergias entre ellos que no surgirían, o con mucho más esfuerzo, que si fueran sistemas distintos. Si los sistemas de recomendación comparten información se ahorran esfuerzos

de recolección y se logran nuevas oportunidades.

III.3. Definición y justificación de cada etapa

Hasta ahora en el presente capítulo hemos presentado una visión general de la arquitectura y después un repaso a los principales retos identificados y cómo estos han condicionado decisiones de diseño de la arquitectura. En esta sección vamos a repasar una a una las distintas etapas de la arquitectura, entrando ya en profundidad en qué deben cumplir, cómo y qué sentido tienen.

III.3.1. Etapa 0: Recolección de OER desde distintas fuentes

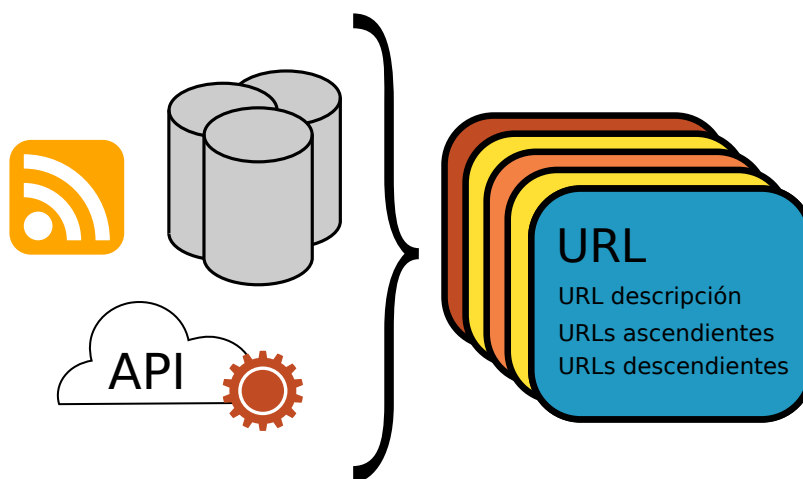


Figura III.2: **Etapa 0: Recolección de OER desde distintas fuentes.** En esta etapa la heterogeneidad es doble: hay múltiples fuentes y múltiples formatos posible, representados aquí con colores distintos. Para cada OER se guarda su URL y opcionalmente se puede acompañar de una URL adicional que enlace a una página de descripción y un listado de URLs de OER que sean ascendientes o descendientes de este.

Un sistema de recomendación no puede recomendar lo que no conoce. Así que la primera etapa es necesariamente la de salir al mundo a conocer lo que se quiere recomendar. La etapa 0 es la etapa encargada de ello.

Cuando hablábamos del reto que supone la heterogeneidad de fuentes, ya adelantábamos que existen tantos modelos de repositorios de OER que es imposible buscar un modelo común que los abstraiga a todos. La única asunción que podemos permitirnos sobre qué contendrá ese repositorio es que habrá un recurso y el recurso estará identificado con una URL. Es un poco tautológico, porque si no hay recurso no hay nada que recomendar y no nos interesa y si no tiene una URL no es accesible y por lo tanto no es abierto y tampoco nos interesa.

Un problema que surge es que no siempre es suficiente solo con la URL del propio recurso.

Muchas veces, como por ejemplo en ficheros de texto, ese recurso puede ser auto explicativo, pero en otras ocasiones como vídeo o audio, estos recursos pueden ir acompañados de una página de descripción del recurso sin la cual las posibilidades de entender el recurso (o conocer la licencia bajo la que se distribuye) disminuyen. Para estos casos hemos contemplado la posibilidad de que a cada URL de recurso se le acompañe la URL de descripción. Esta URL puede ser un HTML enfocado a ser visto por humanos o pueden ser datos enlazados dirigidos al procesamiento automático. Sea como sea, no se puede asumir que este campo contendrá información.

Por último, aparecen los listados de recursos ascendientes y descendientes, sobre lo que no tiene sentido añadir nada a lo explicado en la sección III.2.3.

III.3.2. Etapa 1: Transformación de los OER a una representación común

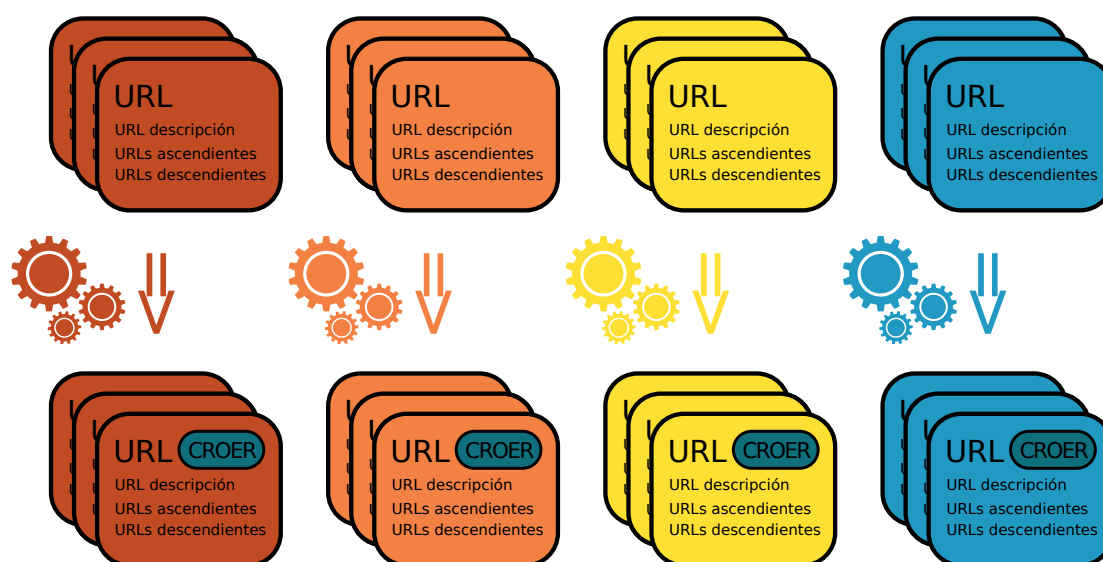


Figura III.3: **Etapa 1: Transformación de los OER a una representación común.** Para cada tipo de recurso (vídeos, texto, simulaciones...) el proceso es distinto, ya que se parte de formatos distintos. El resultado, el llamado CROER, sí es el mismo formato para todos, sin importar el de origen. Para cada recurso existirán varios CROER que recogerán todo lo que sea compartido entre varios algoritmos de recomendación posteriores.

La heterogeneidad de formatos discutida anteriormente en la sección III.2.2, junto con la diversidad de objetivos nos obliga a buscar una representación común de los recursos si queremos lograr un sistema con aspiraciones universales. El problema es que no todos los algoritmos de recomendación pueden trabajar con las mismas representaciones de partida. Existen algoritmos de recomendación basados en datos enlazados, en el contenido de los recursos, en filtrado

colaborativo, en modelos... y como estas representaciones son distintas, al final todo sistema de recomendación debe elegir cuál va a ser su aproximación desde un principio y se limita a algoritmos de recomendación basadas en ella.

Pero, ¿y si no es necesario elegir? Es decir, tenemos opciones de crear representaciones de los OER con datos enlazados o basadas en su contenido transformándolo a texto y posiblemente existan otras muchas. Si aceptamos que podemos tener múltiples sistemas de recomendación, necesariamente tenemos que aceptar que todas estas representaciones pueden convivir y para ello aceptamos que existen varias representaciones posibles para cada recurso. La pregunta ahora es que si en una etapa posterior creamos tantos modelos como sean necesarios para cada algoritmo de recomendación, ¿cuál es el sentido de esta representación común previa? ¿No recoge la diversidad de modelos posterior ya todo lo que necesitamos?

Lo cierto es que no. Es decir, los modelos de recurso que se crean en la etapa siguiente están fuertemente ligados a las necesidades de algoritmos de recomendación concretos. Los CROER que se crean en esta etapa no, y están pensados de forma independiente y abstracta. La regla general para discernir si algo debería ser un modelo o un CROER es si puede ser útil para más de un algoritmo de recomendación: si puede serlo, debería ser un CROER y por lo tanto compartido, porque los modelos de recurso son exclusivos de cada algoritmo de recomendación. Esta distinción es la que justifica este paso.

III.3.3. Etapa 2: Creación de los modelos de recurso y usuario

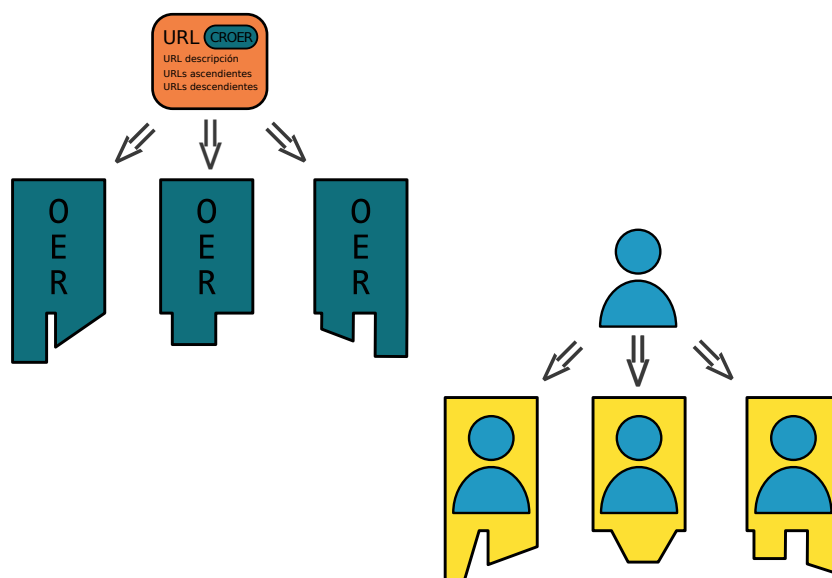


Figura III.4: **Etapa 2: Creación de los modelos de recurso y de usuario.** Para cada recurso o usuario, según el sistema de explotación que venga a continuación, es posible crear distintas representaciones o vistas, que pueden ser compartidas.

Aunque esta y la siguiente etapa son dos diferenciadas, tienen una fuerte dependencia mutua, ya que no se crearán modelos de recurso y de usuario si no para ser utilizados posteriormente por los algoritmos de recomendación de la siguiente etapa.

La idea es que los sistemas que implementen la arquitectura al final parten de una serie de representaciones originales de cada recurso (que son el resultado de la etapa anterior) y del cada usuario (que es toda la información recopilada por el sistema a lo largo de su interacción) y en lugar de trabajar sobre ese modelo primario directamente, cada algoritmo de recomendación posterior debería crear modelos derivados o vistas de este para manipularla ahí la información. Estas vistas permiten que los modelos almacenen información sin pisarse los unos a otros. Es importante en esta etapa prestar especial atención para evitar la duplicidad de información. Primero, porque si una algo que un algoritmo de recomendación crea es útil para otros debería ser un CROER, no un modelo propio. Y segundo, porque si hubiera duplicidades de información eso tiende a estados inconsistentes y si se logran evitar es a base de una duplicación de esfuerzos.

Que varios algoritmos de recomendación compartan un mismo modelo de usuario primigenio puede ser también una gran ventaja en cuanto a la usabilidad del sistema comparado con varios que repliquen la misma funcionalidad. Por un lado, si varios algoritmos de recomendación comparten como entrada algún dato que deben preguntar al usuario, al compartirse con una sola pregunta basta, lo que evita saturar al usuario. Incluso, si el sistema quiere evitar preguntar al usuario e intenta inferir, cuantos más datos tenga disponibles previos, habrá más posibilidades de encontrar relaciones que permitan la inferencia.

III.3.4. Etapa 3: Explotación de los datos

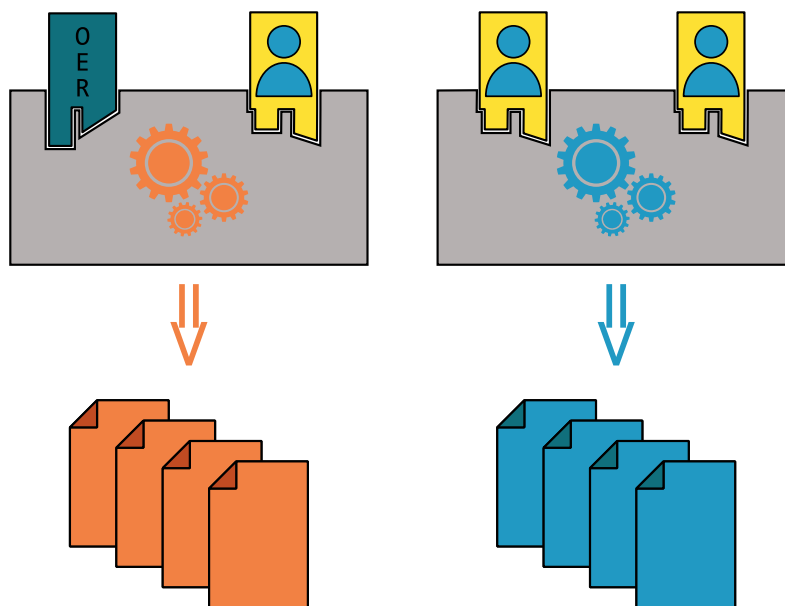


Figura III.5: **Etapa 3: Explotación de los datos.** Existen muchos métodos de recomendación posible, según las necesidades y contexto de los usuarios, y todo pueden convivir bajo la arquitectura. Distintos modelos pueden compartir datos de entrada y a su vez la salida de unos puede ser la entrada de otros, permitiendo construir modelos cada vez más complejos y por lo tanto útiles.

No hay que olvidar que el objetivo final de la arquitectura es hacer recomendaciones y esta es la etapa donde ese objetivo finalmente se cumple. Como ya hemos explicado, en esta etapa múltiples sistemas de recomendación conviven. Los datos de entrada pueden ser compartidos y

los de salida de unos ser la entrada de otros, tantas veces como sea necesario, creando así un ecosistema vivo que va creciendo hacia un sistema cada vez más completo y útil.

La cantidad de algoritmos de recomendación que aquí tienen cabida es inmensa. Hay necesidades que se cubrirán asociando recursos por afinidad temática, por diversidad de formato o porque forman una progresión. Otros algoritmos podrían asociar los recursos con usuarios o incluso a los usuarios entre sí. La idea fundamental es que haya tantos algoritmos de recomendación como pueda ser útil para alguien y que comportan tanta información como sea posible. Ejemplos de posibles algoritmos de recomendación se verán extensamente en el siguiente capítulo.

III.3.5. Etapa 4: Retroalimentación del proceso

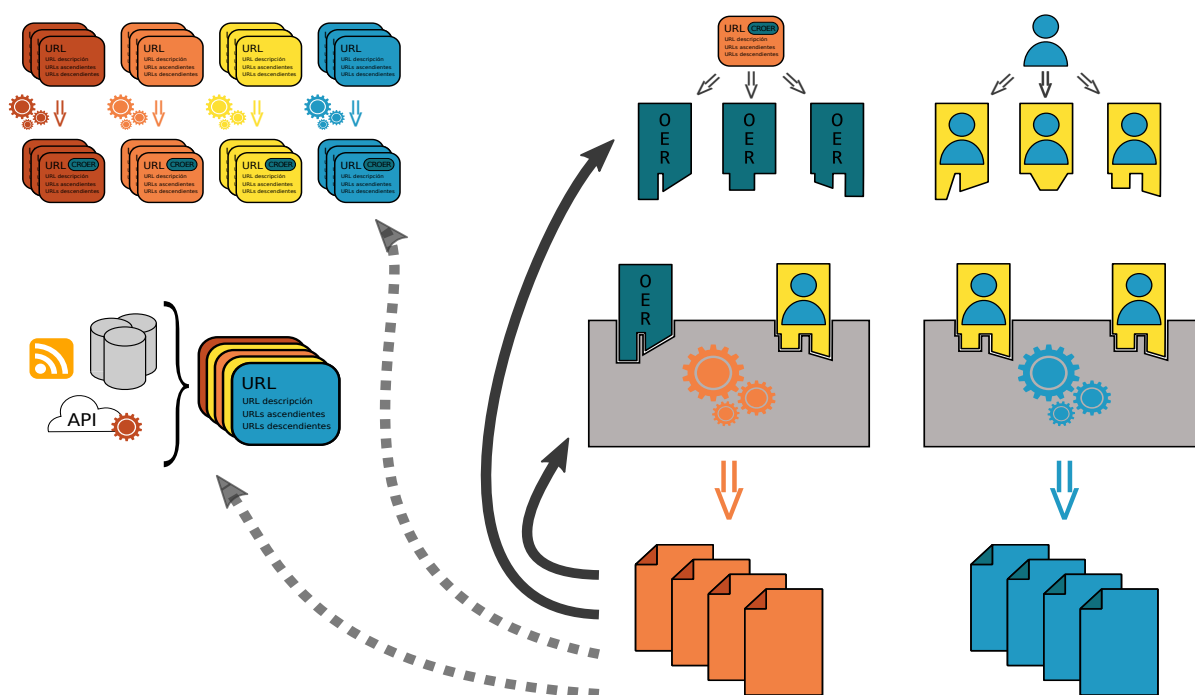


Figura III.6: **Etapa 4: Retroalimentación del proceso.** Partiendo de las recomendaciones finales y de cómo responden los usuarios a ellas (de forma explícita o implícita) se pueden crear datos que permitan afinar el proceso. Las etapas que naturalmente recibirán más retroalimentación son las etapas 2 y 3, mientras que la 0 y 1 son etapas más estables para las que crear sistemas que automáticamente integren la retroalimentación sería demasiado costoso.

Una vez realizadas las recomendaciones muchos modelos pueden aprovecharse de retroalimentación dada por los usuarios y es importante tener esto en cuenta para permitir una mejora paulatina del proceso global. La idea es muy sencilla: una vez un sistema de recomendación ha creado una recomendación, algún otro modelo de análisis de respuesta deberá entrar a analizar la retroalimentación explícita o implícita que genere el usuario. Puede ser tan sencillo como ofre-

cer al usuario botones de “me gusta”/“no me gusta” para cada recomendación y sencillamente incorporar esa información al modelo de cada recurso o puede ser más complejo, como analizar los patrones de actividad del usuario en el sistema recogidos en registros implementados para ello.

Sea como fuere el método para obtener la retroalimentación, todas las etapas previas pueden beneficiarse de las conclusiones que se deriven de esta retroalimentación, pero hay una distinción entre las dos primeras y las dos últimas. Mientras que para las etapas 2 y 3 de creación de modelos y explotación tendrá mucho sentido dedicar esfuerzos a que la incorporación de esta retroalimentación en el proceso pueda ser automática no ocurrirá lo mismo con las etapas 0 y 1, que serán etapas mucho más estables y que necesiten menos cambios, por lo que parece más razonable que estos cambios se vayan a producir solo tras una intervención humana, aunque siempre tras analizar la retroalimentación.

III.4. Discusión: fortalezas y debilidades de la propuesta

En este capítulo hemos presentado la arquitectura, así como algunos retos a los que se intenta dar respuesta con una serie de decisiones concretas dentro del diseño de la arquitectura. Una vez presentada, y antes de pasar a ejemplos y propuestas para la implementación de la arquitectura que veremos en el capítulo IV, vamos a discutir ahora la arquitectura propuesta, intentando destacar sus principales fortalezas, pero también los sacrificios que estas suponen.

Empecemos con las fortalezas:

- Con esta arquitectura es posible tratar todos los formatos de recursos y todas las fuentes de la misma forma, aligerando así el trabajo de los sistemas de recomendación.
- La transformación de los recursos a representaciones comunes facilita la introducción de nuevos algoritmos de recomendación, ya que pueden aprovechar esfuerzos previos con mayor facilidad y se reducen esfuerzos en el mantenimiento.
- Se reconoce desde un principio la idoneidad de elegir una aproximación de divide y vencerás. Crear un sistema de recomendación de OER que responda a todas las necesidades y objetivos a la primera es irrealizable. Con esta arquitectura altamente modular se permite ir construyendo sobre modelos previos.
- Se incluye una etapa separada de creación de modelos partiendo de los CROER para recalcar lo importante que es que los algoritmos de recomendación compartan todo estos CROER en la medida de lo posible para eliminar así esfuerzos duplicados, garantizar que no se dan estados incongruentes y abonar el terreno para que surjan sinergias.
- La importancia de la retroalimentación en el proceso está contemplada desde el principio, logrando así que toda implementación de la arquitectura deba diseñarse con ella en mente y no como un añadido adicional que degrade sus posibilidades.

Pasando al área de las debilidades, una de las principales debilidades de la arquitectura propuesta no tiene tanto que ver con la propia arquitectura sino con el hecho de intentar solucionar un problema en un ámbito donde ya existen muchos intentos de solución con una nueva propuesta. La viñeta de la figura III.7 recoge este sentimiento muy bien. Si uno de los

problemas para el descubrimiento de recursos es que hay inabarcables fuentes, ¿de verdad va a mejorar la situación crear una fuente nueva? Es cierto que esta propuesta justo lo que busca es agrupar cuantos más esfuerzos previos mejor, pero seguro que no es la primera con esas intenciones.



Figura III.7: *Standards*, viñeta sacada de (xkcd, n.d.)

Entrando ya en la propuesta en sí, la idea de desacoplar la etapa de creación de representaciones comunes y la de creación de modelos permitirá facilitar que los algoritmos de recomendación compartan datos, pero a la vez puede que se vuelva todo muy caótico. La idea sería que los CROER, la parte común, se alteraran muy poco y fueran exclusivamente acumulativos (nunca eliminando contenidos de ellos) para evitar que algoritmos de recomendación se quedaran huérfanos, pero puede ser difícil predecir a la hora de introducir un nuevo algoritmo de recomendación si la información que necesita podría ser útil para otros modelos posteriormente y debería ir al CROER o si es algo más particular y debería estar en el modelo de recurso.

Otro potencial punto de fallo viene de la convivencia de múltiples sistemas de recomendación. Si existen varios, es posible que algunos entren en conflicto, principalmente porque lleguen a conclusiones contradictorias que quieran almacenar. ¿Cómo el sistema puede resolver esos conflictos? Ahora mismo no hay definido ningún mecanismo y se deja a la implementación porque se entiende que debería ser situaciones relativamente sencillas de resolver con un mecanismo de ponderación (que se implemente como un nuevo algoritmo de recomendación sobre los anteriores) que se afine con retroalimentación. Pero si esta situación se diera muy a menudo o no fuera tan sencilla de solucionar, podría llegar a tener sentido presentar una revisión de la arquitectura que contemplara una nueva etapa para ello.

Evaluación de la viabilidad de la arquitectura

En el capítulo anterior hemos presentado una arquitectura para recomendación de OER dividida en 5 etapas. En este capítulo vamos a detallar algunas propuestas sobre cómo podría ser la implementación de estas etapas. En la figura III.1 se puede encontrar una representación esquemática de la arquitectura, que podría ser útil para la lectura de este capítulo.

IV.1. Etapa 0: Recolección de OER desde distintas fuentes

Decían en 1998 los fundadores de Google que *“Crawling is the most fragile application since it involves interacting with hundreds of thousands of Web servers and various name servers which are all beyond the control of the system.”* (Brin y Page, 1998) y esta primera etapa va principalmente de crawling.

Idealmente, todo repositorio de OER ofrecería una API pública que ofreciera una manera sencilla y rápida de recolectar los OER. Aún más ideal sería que esa API fuera idéntica en todos los repositorios. Desgraciadamente, no todos los servicios tienen dicha API pública y de tenerla cada servicio tiene sus características y limitaciones propias. Siendo así, no queda otra alternativa que realizar crawling, es decir, un programa que se dedique a recorrer la web de una forma ordenada para guardar lo que resulte interesante.

Realizar crawling tiene una serie de importantes retos y es una línea de investigación en sí misma, por lo que aquel sistema que vaya a implementar la arquitectura necesariamente deberá investigarlo. Algunos de los retos que más puede afectar al crawling de OER son los siguientes:

Servidores con limitaciones: Un crawler tiene como objetivo visitar el mayor número de páginas web en el menor tiempo posible, por lo que intentará maximizar el número de peticiones que realiza. Un servidor web, sin embargo, está más cómodo cuantas menos peticiones recibe y es muy posible que esto lleve a conflictos entre los crawlers y los servidores. Los servidores web intentan garantizar el acceso al contenido a todos los usuarios y un crawler descontrolado puede acaparar demasiados recursos, llegando a ralentizar el servidor o incluso causar un inintencionado ataque de denegación de servicio. Siendo así, es habitual que los servidores tengan restricciones en el número de peticiones por segundo que permiten hacer por usuario. Algunos servidores las anuncian, otros no, pero el número

es altamente variable, por lo que planificar es complicado. Una aproximación bastante expandida es utilizar varias colas con prioridad orientadas a garantizar que el mismo servidor no recibe demasiadas peticiones seguidas, pero que aun así garantiza que el crawler es eficiente (Najork y Heydon, 2002).

Trampas para crawler: Muchos servidores web tienen trampas para crawlers, es decir, situaciones que pueden ralentizar o incluso atrapar a un software de crawler. El ejemplo más claro es una página infinita, como un calendario, en la cual un crawler podría quedarse atrapado intentando llegar a un final que no existe antes de visitar otras páginas. Algunos servidores web incluyen estas trampas de forma malintencionada, pero la gran mayoría son sencillamente consecuencia de necesidades del servicio. Un método para evitarlas es asignar a cada dominio una métrica de presupuesto temporal o de recursos y salir de él una vez se agote (Olston y Najork, 2010).

Orden de visita: Como el número de páginas en la web crece a un ritmo frenético y ya existen en una cantidad inabarcable de páginas, todo crawler tiene que renunciar al sueño de visitarlas todas. Siendo así, el orden en el que se visiten es relevante, ya que no tienes la certeza de que eventualmente llegarás a lo que buscas. Además, el crawler de muchas fuentes no es algo instantáneo, por lo que los recursos irán apareciendo a lo largo del tiempo. Por todo esto, nos gustaría que el crawler visitara primero las páginas más relevantes, para obtener primero lo que más nos interesa. Existen muchas maneras de hacer esto, pero de entre todas ellas destaca el algoritmo de PageRank (Cho, Garcia-Molina, y Page, 1998), que es el introducido por Google y que asigna una métrica de importancia a las páginas según la topología de la red que se forma por los hipervínculos entre páginas (entendiendo que las páginas muy referenciadas son las mejores valoradas).

Pero la búsqueda de recursos en repositorios no es la única opción disponible y existen otras dos aproximaciones a considerar. Por un lado, el sistema puede permitir a los usuarios que suban al servicio sus propios OER o links a los OER. Si solo se hace esto, se elimina por completo la carga de recopilar los OER del sistema y se traslada a los usuarios, pero para ello hace falta una gran masa de usuarios productores desde el principio, lo cual es difícil de conseguir si no hay nada para atraerles al sistema.

La otra opción es la búsqueda de recursos en redes sociales, especialmente, dentro de comunidades dedicadas a la enseñanza, ya sea dentro de redes sociales centradas en la enseñanza o genéricas. Es fácil imaginar que en los foros del CMS de cualquier institución educativa se deben compartir muchos OER, pero solo la institución dueña del CMS puede acceder a esa información. En redes sociales genéricas cualquiera puede acceder, pero puede ser igualmente muy costoso encontrar los OER entre todo el contenido. La ventaja es que todo OER encontrado ha sido al menos avalado por una persona, lo que ya es un filtro a la calidad (y si se tienen en cuenta el número de personas que lo comparten, eso ya es una potente métrica de calidad).

Sea cuál sea el método de recolección, existen una serie de problemas comunes a todos ellos:

Contenido redundante: Al existir tantas fuentes de OER, cada una más enfocada a un tipo de público, centrarse solo en una no parece una opción para un sistema genérico. Y un problema que surgirá en cuanto haya dos fuentes (y se hace más grave con cuantas más fuentes haya) es el problema del contenido redundante. Es muy posible encontrar el mismo recurso en dos fuentes y no es trivial identificarlos como tal. Lo que es aún más complicado de tratar es cuando esas dos fuentes tienen dos versiones distintas del mismo

OER (por ejemplo, porque una de las fuentes tenga una errata corregida pero no la otra). Si muchos OER que son iguales son identificados como distintos por venir desde distintas fuentes podría ser problemático para los sistemas de recomendación, que los penalizarán injustamente respecto a los recursos que provengan solo de una fuente.

Mantenimiento de los OER: Es común que se piense que la información es algo estático, pero está siempre en evolución y como todo sistema que evoluciona necesita mantenimiento. Los OER son información con un fin y licencia concretos, pero información, al fin y al cabo. Un sistema que implemente la arquitectura deberá decidir si mantiene una copia de los recursos que descubre o si delega esa responsabilidad en las fuentes. La primera aproximación garantiza que siempre habrá una copia disponible, pero supone más coste en almacenamiento y no evita que cada cierto tiempo se deba comprobar con la fuente si el OER ha evolucionado. Si se delega en la fuente como almacenaje del OER, es posible que cuando se vaya a recomendar ya no esté disponible, por lo que también hay que comprobar con cierta frecuencia si esto es así.

IV.2. Etapa 1: Transformación de los OER a una representación común

En la sección III.3.2 justificábamos que, dado que existen múltiples familias de métodos de recomendación y que como cada familia entendía las entidades a recomendar de una forma, esto nos llevaba a que existían varias representaciones posibles para cada OER, en función de la familia. En esta sección vamos a repasar estas familias.

Una de las más prometedoras a nivel teórico es la recomendación basada en datos enlazados, es decir, en una visión que busca que los servidores web no solo sirvan información entendible por humanos, sino que además se publiquen datos estructurados y enlazados entre sí en un formato entendible por máquinas que permitan búsquedas semánticas. Estas búsquedas semánticas serían búsqueda en las que los buscadores comprenden las relaciones entre los objetos (ya que están recogidas en los datos enlazados) y por lo tanto son capaces de descubrir información implícita. Por ejemplo, si le preguntamos a Google “¿cuál fue el objetivo de la misión Apollo 7?” nos devolverá resultados que mencionen las misiones Apollo y la palabra objetivo, pero sin entender la relación entre ambas. De hecho, la búsqueda en Google España a día 2018-06-16 devuelve primero la página de Wikipedia del “Programa Apollo” (donde se repite mucho Apollo y objetivo), y después a la página de la misión “Apollo 17” en la ni aparece la palabra objetivo ni acierta en el número de misión. Una búsqueda semántica debería ser capaz de dar la respuesta correctamente que, para los curiosos, fue la primera misión tripulada con éxito del programa: se probó el rendimiento del precursor del cohete Saturno V, así como los trajes espaciales que se usarían en misiones posteriores y también contó con la primera retransmisión de TV en directo desde el espacio realizada desde una nave estadounidense.

Este método de recomendación basada en datos enlazados es en el que se centran (Chicaiza et al., 2017) y ya se ha introducido en la sección II.2.3. El problema que surge con esta aproximación es que es necesario que todos los repositorios cumplan con los estándares de datos enlazados pero muchos repositorios no ofrecen estos datos y los que lo hacen no tienen por qué ser siempre completos o consistentes dentro del mismo repositorio (Romero Pelaez y Alarcon, 2017) y menos aún entre diferentes repositorios.

Otra aproximación es la recomendación basada en contenido. La idea aquí es que el software

pueda interpretar el contenido de los recursos y en función de eso hacer las recomendaciones. La ventaja que tiene es que podemos asegurar que todos los repositorios y los recursos en ellos necesariamente tienen contenido. El problema es que el contenido es muy diverso: vídeos, textos, audios. . . Si transformamos ese contenido a una representación común ya es posible trabajar con fuentes heterogéneas. Una de las opciones más habituales es que esa transformación sea a texto, por dos motivos:

- El texto es una representación del lenguaje humano y el lenguaje humano es lo suficientemente amplio como para describir vídeos, imágenes o cualquier otra entidad de conocimiento.
- No existe ninguna otra representación tan amplia que sea tan eficiente a la hora de ser tratada a nivel informático como el propio texto, lo cual facilita el desarrollo y mantenimiento.

Sabemos que un ser humano puede expresar de forma escrita un vídeo, pero el problema viene cuando intentamos automatizar ese proceso. A día de hoy ya existen técnicas para generar automáticamente descripciones textuales de imágenes (Karpathy y Fei-Fei, 2017) y el área de transcripción de audio a texto está bastante estudiada, por lo que es cuestión de tiempo tener software que sea capaz de describir con palabras el contenido de vídeos, pero tendrá que pasar aún un tiempo hasta que estas técnicas se puedan aplicar de forma masiva a un coste razonable. Hasta entonces, podemos conformarnos con lo que suelen hacer los buscadores web a día de hoy: no conocemos qué hay en un vídeo o una imagen, pero casi siempre aparecen acompañados de texto y podemos mirar qué dice ese texto. Este método supone que, si hay texto y vídeo/imagen en una página, existe una relación entre ambos. Es decir, que si debajo de la misma foto aparece muchas veces las palabras “Presidente Obama” posiblemente se trate de una imagen del Presidente Obama.

Por último, otra representación muy común es la que ignora el contenido de los recursos a recomendar y se centra en estadísticas de dicho recurso, en general mirando el comportamiento de usuarios con ese recurso. El ejemplo más claro es los modelos utilizados para filtrado colaborativo, que al final buscan descubrir patrones en qué usuarios han interactuado con cada recurso para encontrar usuarios similares y recomendar lo que ha gustado a esos usuarios. Estos modelos de recurso también tienen potencial para ser compartidos entre varios algoritmos de recomendación y por lo tanto también tiene sentido tratarlos como CROER. Por ejemplo, el filtrado colaborativo se puede hacer en base a usuarios o en base a recursos, pero en ambos casos se parte de una representación en la que se registra qué usuario ha interactuado con qué recurso.

IV.3. Etapas 2 - 3: Creación de los modelos de recurso y usuario - Explotación de los datos

En este capítulo tratamos estas dos etapas en un mismo apartado porque la decisión de qué modelos de recurso y usuario se construyen desde los CROER de la etapa anterior está fuertemente ligado a qué algoritmos de recomendación se usarán en la etapa de explotación de datos.

Recordemos que lo más novedoso de esta arquitectura es que pone especial atención en que puedan convivir múltiples sistemas de recomendación de distintas familias en el mismo sistema

final, buscando así solucionar el problema abstracto de la recomendación de OER mediante una estrategia de divide y vencerás y reconociendo que cada algoritmo de recomendación funcionará mejor para una tarea distinta.

La cantidad de modelos que se han probado en distintos sistemas es abrumadora (Drachler, Verbert, Santos, y Manouselis, 2015), por lo que repasar todos ellos sería excesivo. Vamos a repasar en este capítulo algunos de los más utilizados, con una breve explicación de cada uno.

Como ya hemos introducido varias veces, una de las familias más populares es la conocida como métodos basados en contenidos (Lops, de Gemmis, y Semeraro, 2011), que no consideran al recurso como una caja negra, sino que intentan inferir la recomendación en base a lo que contenga. En esta familia brilla como método estrella el conocido como tf-idf (Salton et al., 1983). Este método parte de una representación textual de los recursos y calcula dos métricas (tf e idf) que combina para realizar un ranking de recursos en función a una cadena de búsqueda. La primera métrica es tf, *Term Frequency*, y se define para cada término y cada recurso, y es proporcional al número de veces que aparece cada término en cada documento, porque supone que si un documento repite más veces una palabra de las palabras que aparecen en la búsqueda será más relevante. La otra métrica es idf, *Inverse Document Frequency* que es una métrica que simboliza cómo de discriminatoria es cada palabra, ya que computa para cada término el número de documentos en los que aparece. Cuando un usuario busca, se calcula el peso de cada palabra de la cadena de búsqueda que pondera al número de veces que aparece en cada documento, de tal manera que se devuelven primero los documentos que contienen más veces las palabras menos frecuentes de la búsqueda. Este método es ideal para búsquedas y hacer un filtrado del conjunto de recursos para que concuerden con una temática expresada por el usuario como una búsqueda.

Otros sistemas dentro de la familia de basados en contenido serían los sistemas basados en reglas, en los que, en función de atributos del documento y una serie de reglas, se van haciendo recomendaciones. La ventaja más evidente de este sistema es que se pueden explicar de forma clara a los usuarios el proceso de razonamiento del sistema, lo cual es especialmente valorado por profesores (Santos y Boticario, 2010).

Por último, en los basados en contenido vamos a ver aquellos basados en la adaptación y modelado de usuario. Estos métodos tienen un fuerte modelo de características del usuario y en base a él realizan la adaptación. Uno de los más utilizados es algún modelo de estilos de aprendizaje, como el de (Felder y Silverman, 1988), que asigna a los usuarios métricas en cómo responden a ciertos estímulos, para que el sistema priorice esos estímulos para esos usuarios (Paredes y Rodríguez, 2004; Alshammari, 2016).

Los sistemas basados en datos enlazados y ontologías son catalogados por algunos como otro caso de sistemas de recomendación basados en contenido (Lops et al., 2011) mientras que para otros son su propia categoría (Chicaiza et al., 2017). Sea como fuere, ya se han introducido en la sección anterior.

Por último, la otra gran familia de métodos de recomendación es la de filtrado colaborativo, que se diferencia de las basadas en contenido porque tiende a centrarse más en cómo interaccionan los usuarios con el recurso que con el contenido en sí. Ofrecen la ventaja de que son métodos más independientes del contexto, pero esa podría ser también una de sus principales limitaciones. La familia de filtrado colaborativo se subdivide en dos grandes grupos (Koren y Bell, 2011): sistemas basados en vecinos y factorización de matrices.

Los sistemas basados en factorización de matrices son una progresión respecto a los basados en vecinos, ya que logran mayor precisión en las recomendaciones e integrar nuevas métricas, como

factores temporales, niveles de confianza y retroalimentación implícita (Koren, Bell, y Volinsky, 2009). Parten de una representación vectorial para usuarios y recursos en la que existe para cada entidad un vector con tantas dimensiones como factores se tengan en cuenta. Así, el producto escalar de estos dos vectores se puede entender como el interés del usuario en el recurso en función de todos los factores tenidos en cuenta. Un sistema de recomendación se construye sobre esto calculando vectores de pesos para cada factor que minimicen el error de las predicciones respecto a lo observado, usando para ello un algoritmo de descenso estocástico del gradiente o el método conocido como *alternating least squares* (ALS), ambos métodos importados desde otras disciplinas.

Pero recordemos que la arquitectura parte de la idea de que lo más potente posiblemente será la integración de todos estos sistemas en la consecución de una sola tarea. Para ello, al final harán falta métodos de recomendación que trabajen sobre los anteriores ponderando las distintas salidas, ya sea utilizando métodos de votación, de ponderación o métodos específicamente diseñados para ello.

IV.4. Etapa 4: Retroalimentación del proceso

La última etapa es la de retroalimentación. Algunos algoritmos de recomendación, como los de filtrado colaborativo, se construyen directamente sobre la información recolectada en esta etapa, por lo que, de nuevo, es importante tener en cuenta el resto de etapas a la hora de planificar esta. De todas las etapas, las etapas 2 y 3 son las que tienen más potencial de incorporar mecanismos automáticos que incorporen la retroalimentación en el sistema.

Dentro de la retroalimentación tenemos dos categorías principales respecto a si el usuario es consciente de que está suministrando retroalimentación: retroalimentación explícita e implícita.

En la explícita tenemos los sistemas donde el usuario directamente ofrece su opinión sobre la recomendación. Lo más habitual es que sea a través de un sistema de puntuación de recurso o mediante comentarios. El primer método es en el que se basan principalmente los métodos de filtrado colaborativo y aunque potencialmente recoge menos información que los comentarios, es más sencillo de tratar y directo de interpretar.

El debate que ha surgido en los últimos años es sobre qué tipo de escala utilizar. Tradicionalmente los sistemas comerciales ofrecían escalas de 1 a 5 estrellas (algunos, como Amazon, siguen funcionando así) pero otros muchos han migrado a un sistema binario de pulgar arriba/pulgar abajo (aunque en el fondo es ternario, porque no valorar un recurso que se ha visto es prácticamente una categoría en sí mismo). Esta transición hacia el sistema binario se ha visto en gran medida producida por el patrón típico de respuesta en el sistema de 5 estrellas, en el que una inmensa mayoría de las valoraciones eran de 5 estrellas, unas cuantas de 1 y muy pocas de números intermedios (YouTube, 2009).

En cuanto a retroalimentación implícita, tenemos todo sistema que intente deducir la satisfacción del usuario sin preguntar directamente. Para esto dos de las técnicas más habituales son la de estudiar los patrones de navegación por la página o el método de Test A/B. En el primer caso, lo que se hace es estudiar los logs del servidor web para agrupar las peticiones por usuarios y de ahí deducir cómo de efectiva ha sido una recomendación. Se ha utilizado ya en entornos de sistemas online de aprendizaje (Keskin, Sahin, Ozgur, y Yurdugul, 2016) y en sistemas genéricos se ha utilizado para introducir la retroalimentación en tiempo real (Wu, Alvino, Smola, y Basilico, 2016). En el caso del Test A/B, se asignan aleatoriamente usuarios a uno de dos

posibles algoritmos de recomendación sin que ellos lo sepan y se compara la retroalimentación de los usuarios ante ambos algoritmos. Este es el caso ideal para hacer pruebas que determinen si un algoritmo de recomendación da resultados mejores que otro con evidencia estadística, pero genera duda su utilización en sistemas educativos, ya que podría suponer una desventaja competitiva en la evaluación final para los usuarios asignados al modelo que da peores resultados.

IV.5. Ejemplo de aplicación de la arquitectura

Como parte de los estudios de master que concluye con este trabajo, en dos asignaturas distintas había que presentar un proyecto final. Para estas asignaturas se desarrolló un sistema de recomendación de OER personalizado según estilos de aprendizaje. Este sistema se detalla aquí como un ejemplo de cómo sería la aplicación de la arquitectura a un desarrollo normal, y porque estos trabajos sirvieron para refinar la arquitectura. Es importante aclarar que la programación de este prototipo no fue obra exclusiva del autor ya que contó con la colaboración de Sara Pérez Soler, su compañera de prácticas. El trabajo de implementación del ejemplo se dividió entre medias, pero la definición de la arquitectura es un trabajo exclusivo del autor.

El sistema que se implementó consistió en un sistema web de recomendación de recursos educativos en base al perfil de estilo de aprendizaje del usuario (tomando el modelo de (Felder y Silverman, 1988)). El sistema permite a personas registradas hacer una búsqueda por palabras y el sistema devuelve un listado de recursos en los que priman aquellos bien en formato PDF o bien en formato vídeo en función de si la persona tiene una preferencia verbal o visual, respectivamente. En la figura IV.1 se puede observar la pantalla de resultados para dos usuarios distintos, uno con una preferencia extrema hacia lo visual y el otro con una práctica indiferencia.

Para este prototipo se utilizaron dos fuentes de datos distintas: la página del MIT OpenCourseWare y la página de YouTube. Aunque todos los recursos del MIT OCW sí son recursos educativos distribuidos bajo licencias abiertas, no es el mismo caso para YouTube, donde la inmensa mayoría de sus vídeos están publicados bajo licencias no abiertas (aunque la plataforma ofrece la opción de publicar bajo una licencia Creative Commons, pero es raro que esto ocurra). La elección de YouTube como fuente de datos vino motivada más por las facilidades que da para recolectar su contenido que por las licencias.

En la página MIT OCW los recursos están ordenados por asignaturas y los hay en múltiples formatos, como vídeos, PDFs, páginas web, software... Para esta prueba no se recopilaban todos, sino solo los PDFs, ya que el objetivo principal era probar la adaptación según los estilos de aprendizaje y para ello se decidió asociar PDFs a estilos verbales y los vídeos de YouTube a los visuales, partiendo de la asunción de que los PDFs serían principalmente texto escrito (que no tiene por qué ser cierto, ya que también es muy habitual los PDFs con diagramas).

Para la página del MIT OCW se creó un programa que se dedicó a visitar todas las páginas del mismo dominio partiendo desde la raíz, saltando de hiperenlace a hiperenlace y guardando solo las URLs de PDFs. No se registraron páginas de descripción, ni recursos ascendientes o descendientes. Para la página de YouTube se hizo similar, pero empezando desde páginas de canales seleccionados manualmente por contener vídeos educativos de calidad. Se iban visitando todas las páginas enlazadas, pero de nuevo solo se registraban las URLs de los vídeos. Partimos de la asunción de que los vídeos que se registrarían serían todos vídeos relacionados con aquellos de los canales originales, pero no fue así. No tuvimos en cuenta que en la columna de vídeos recomendados de YouTube que aparece a la derecha de cada vídeo YouTube no intenta mostrar

Posición pre adaptación	Posición	URL	Tipo
4	1	https://www.youtube.com/watch?v=FZGugFqdr60	video
5	2	https://www.youtube.com/watch?v=rtAIC5J1U40	video
6	3	https://www.youtube.com/watch?v=dXdoim96v5A	video
8	4	https://www.youtube.com/watch?v=Nws5-kp8Blk	video
9	5	https://www.youtube.com/watch?v=AwUirxi9eBg	video
10	6	https://www.youtube.com/watch?v=LXOaCkbt4ll	video
11	7	https://www.youtube.com/watch?v=QzWW-CBugZo	video
13	8	https://www.youtube.com/watch?v=qTB3AEEUju8	video
14	9	https://www.youtube.com/watch?v=dHWFpkGsxOs	video
15	10	https://www.youtube.com/watch?v=ObnosznZvHY	video
17	11	https://www.youtube.com/watch?v=7FBzL7Jlifw	video
18	12	https://www.youtube.com/watch?v=X7rCxs1ppyY	video

(a) Resultados para un usuario con una puntuación 11/11 en visual.

Posición pre adaptación	Posición	URL	Tipo
1	1	https://ocw.mit.edu/courses/comparative-media-studies-writing/cms-608-game-design-fall-2010/assignments/MITCMS_608F10_p2g1_cards.pdf	pdf
2	2	http://ocw.mit.edu/courses/comparative-media-studies-writing/cms-608-game-design-fall-2010/assignments/MITCMS_608F10_p2g1_cards.pdf	pdf
3	3	http://ocw.mit.edu/courses/aeronautics-and-astronautics/16-01-unified-engineering-i-ii-iii-iv-fall-2005-spring-2006/comps-programming/mud6.pdf	pdf
4	4	https://ocw.mit.edu/courses/aeronautics-and-astronautics/16-01-unified-engineering-i-ii-iii-iv-fall-2005-spring-2006/comps-programming/mud6.pdf	pdf
5	5	http://ocw.mit.edu/courses/comparative-media-studies-writing/cms-608-game-design-fall-2010/assignments/MITCMS_608F10_p2g1_rules.pdf	pdf
6	6	https://ocw.mit.edu/courses/comparative-media-studies-writing/cms-608-game-design-fall-2010/assignments/MITCMS_608F10_p2g1_rules.pdf	pdf
7	7	https://ocw.mit.edu/resources/res-ll-005-d4m-signal-processing-on-databases-fall-2012/lecture-notes-and-class-videos/MITRES_LL-005F12_Lec0.pdf	pdf
11	8	https://www.youtube.com/watch?v=QzWW-CBugZo	video
8	9	https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-826-principles-of-computer-systems-spring-2002/assignments/s4.pdf	pdf

(b) Resultados para un usuario con una puntuación 1/11 en visual.

Figura IV.1: Ejemplos de resultados para dos usuarios, cada uno con unas preferencias distintas. Se puede observar que el usuario que es puramente visual solo recibe vídeos, mientras que el que tiene una libera preferencia ve mayoritariamente PDFs y algún vídeo (ya que, para esta búsqueda, sin adaptar, los primeros resultados eran mayoritariamente PDFs).

vídeos relacionados temáticamente o según calidad, sino que intenta maximizar el tiempo de estancia en la plataforma para así aumentar sus ingresos por publicidad. Siendo así, YouTube recomienda algunos vídeos por temática, pero principalmente recomienda vídeos populares, por lo que el software rápidamente empezó a incorporar los últimos adelantos de esperadas películas o recopilaciones de vídeos de gatos haciendo cosas divertidas.

En total, durante la etapa 0, se recopilaron 57.414 vídeos y 98.634 PDFs. Para la etapa 1 se creó un CROER basado en texto y se utilizaron dos aproximaciones distintas, una para cada tipo de formato. Para los PDFs sencillamente la transformación consistió en extraer el texto que había dentro del PDF ignorando el resto del contenido. La principal limitación de esto es que algunos quedan vacíos ya que son imágenes escaneadas sin texto incrustado. Para los vídeos se creó un software que utilizando la API pública de Google (Google, n.d.), descargaba el título de los vídeos, su descripción y los 100 comentarios mejor valorados, que concatenaba para crear la versión textual del vídeo. Esta aproximación asume que en los comentarios se discute el contenido del vídeo, lo que ocurre en gran medida en los vídeos educativos (donde la gente hace preguntas o comentarios serios) pero no tanto en vídeos de entretenimiento, donde la temática es libre.

Partiendo de esta CROER textual, se creó un modelo de recurso destinado a la explotación posterior por un esquema vectorial del tipo tf-idf (Salton et al., 1983). Para ello, partiendo de la CROER se tokenizó el texto eliminando todos los blancos (espacios, saltos de línea...) y eliminando todas las palabras de menos de 1 carácter, pero no se realizó ningún tipo de procesamiento de lenguaje natural adicional, como la agrupación de los términos por raíces semánticas. Además de este modelo de recurso se creó un modelo de usuario que sencillamente contenía la puntuación del usuario en la escala visual/verbal de Felder-Silverman, ignorando el resto de escalas del modelo.

La etapa de explotación se realizó con dos modelos distintos. El primero, como ya hemos dicho, el modelo tf-idf, que calcula el peso de cada término de cada documento dentro de este y dentro de todos los demás para asignarle un peso a su capacidad de discernir un documento de otro y con esos pesos y la consulta del usuario calcula que recursos son los que contienen más los términos de la consulta. Como resultado de este modelo se devuelve una lista ordenada de recursos según su afinidad con la consulta del usuario y es justo esta lista junto con la puntuación del usuario en la escala visual/verbal la entrada del siguiente modelo, que reasigna las puntuaciones de la lista según si el recurso es afín a la preferencial del usuario o no. Para ello, se tiene en cuenta si el usuario tiene una preferencia extrema o más neutra, restando o sumando una parte porcentual mayor o menor del valor original, respectivamente. Una vez recalculadas las puntuaciones, se reordena la lista de recursos y esta es la que se muestra al usuario.

Finalmente, decir que no hubo una etapa de retroalimentación, ya que los únicos usuarios del sistema fuimos los desarrolladores y no se generaron datos suficientes como para aplicar ningún tipo de retroalimentación automática, aunque sí sirvió el sistema para hacer una positiva primera implementación de la arquitectura, que reafirma el acierto de trabajar con CROER y la multiplicidad de algoritmos de recomendación cooperando por un objetivo final.

Conclusiones y trabajo futuro

La recomendación de recursos educativos abiertos plantea una serie de retos. Por un lado, existen muchos perfiles de usuarios distintos que pueden estar interesados. De esa multiplicidad de perfiles, llegamos a multiplicidad de objetivos. Se han creado muchos sistemas enfocados a cada uno de esos objetivos, pero el panorama actual de “un problema, una propuesta” dificulta que se aprovechen las sinergias que pueden surgir en un entorno en el que se comparta más información entre algoritmos de recomendación.

Por otro lado, los OER son en sí mismos muy diversos. Existen muchas fuentes de OER, estructuradas de distintas formas, que ofrecen distintas facilidades de acceso y que se enfocan al final a distintos perfiles de usuario. Así mismo, los propios OER se presentan en muy distintos formatos (vídeos, texto, audio, software. . .) y esto en sí mismo supone un reto.

Crear un sistema de recomendación desde cero que intente dar solución a todas estas cuestiones parece una empresa destinada al fracaso. Sin embargo, una estrategia de *divide y vencerás*, en la que se vayan integrando progresivamente múltiples soluciones previas e independientes bajo un mismo sistema parece tener potencial.

A lo largo de este trabajo se ha presentado una propuesta de arquitectura para la creación de un sistema de recomendación de OER multipropósito, basado en el principio de *divide y vencerás*, que pueda trabajar con múltiples fuentes, formatos y algoritmos de recomendación, para así lograr un sistema que aproveche todas las sinergias.

La arquitectura presentada se divide en cinco etapas:

Etapla 0: Recolección de OER desde distintas fuentes. En esta etapa se recorren distintos repositorios y páginas web con OER para agregarlos en un repositorio local desde el que se realizarán las recomendaciones.

Etapla 1: Transformación de los OER a una representación común. Cada algoritmo de recomendación representa a los recursos a recomendar de una forma distinta, dependiendo del tipo de algoritmo que sea. Muchos de estos algoritmos pueden compartir esa representación y al hacerlo estamos facilitando la introducción de nuevos algoritmos de la misma familia, así como simplificando el mantenimiento de todos los existentes.

Etapla 2: Creación de los modelos de recurso y usuario. De las representaciones comu-

nes de la etapa anterior es necesario hacer adaptaciones a modelos concreto de recursos y de usuario para que los algoritmos de recomendación los utilicen para su recomendación. Aunque fusionar esta etapa y la anterior sería posible, al separarlas se pone especial hincapié en las bondades de compartir modelos entre los distintos algoritmos de recomendación.

Etapa 3: Explotación de los datos. En esta etapa es donde, construyendo sobre el trabajo de las etapas anteriores, se realiza finalmente la recomendación. Al ser una arquitectura multipropósito y basada en el principio de divide y vencerás, en esta etapa pueden (y deben) existir múltiples algoritmos de recomendación. Algunos de ellos se construyen sobre otros algoritmos previos, incorporando la salida de estos algoritmos, lo que permite ir acumulando soluciones para hacer más completo el proceso de recomendación.

Etapa 4: Retroalimentación del proceso. Por último, en todo sistema de recomendación es fundamental incorporar procesos de retroalimentación, implícita o explícita, que permitan ir afinando las recomendaciones.

Además de la definición de la arquitectura, se han repasados las distintas soluciones existentes a día de hoy que podría implementarse en cada una de las etapas, recogiendo las soluciones que más recorrido han tenido en la literatura científica, aprovechando el conocimiento desarrollado en las áreas de recuperación de información, minería web, sistemas adaptativos y modelado de usuarios, ya que cada una de ellas puede hacer grandes aportaciones al sistema integrado final.

Pero este trabajo no es sino el primer paso en una investigación mucho más larga, que se desarrollará a lo largo de estudios de doctorado. El siguiente paso es la creación y evaluación de una implementación de la arquitectura. Para esto se plantea seguir dos estrategias: diseño iterativo e incremental y diseño centrado en el usuario.

Los primeros esfuerzos de siguientes trabajos irán destinados a conocer los distintos tipos de usuario que podrían estar interesados en la recomendación de OER, cuáles son sus necesidades, objetivos y expectativas. Los OER pueden ser relevantes en toda etapa educativa, desde la infancia hasta la edad adulta, así como en contextos de enseñanza formal o informal, guiada o independiente, por lo que un esfuerzo importante deberá dedicarse a caracterizar bien este contexto para guiar la investigación hacia resolver problemas reales.

Las distintas propuestas de implementación detalladas en el trabajo deberán efectivamente implementarse e ir evaluándose de forma progresiva para conocer la eficacia de cada una de ellas en los distintos contextos. Así mismo, la integración progresiva de todas estas estrategias de recomendación requerirá esfuerzos propios. Este sistema deberá dedicar una especial atención a la creación de la interfaz y experiencia de usuario, atendiendo a que resulte adecuada para todos los perfiles de usuarios, siempre manteniendo un equilibrio entre funcionalidad y complejidad aceptable.

Una vez el sistema empiece a coger suficiente forma, esta implementación de la arquitectura deberá ser evaluada para comprobar su efectividad y limitaciones en entornos reales. Al preverse un diseño centrado en el usuario estas pruebas en entornos reales se realizarán lo más pronto posible para incorporar cuanto antes al proceso las conclusiones de las mismas.

Dada la aproximación altamente integradora de la propuesta, es deseable que estos trabajos futuros no se limiten a integrar métodos de recomendación, sino que además construyan nuevos y mejorados, aprovechándose de las sinergias que pueden surgir en un sistema que integre tan distintas aproximaciones.

Idealmente, si las pruebas con usuarios funcionan y se logra ampliar a comunidades de usuarios mayores, así mismo se prevé la generación de conjuntos de datos para fomentar la investigación en sistemas de recomendación en contextos educativos, ya que la gran mayoría de los conjuntos de datos públicos en los que se basan las investigaciones están centrados en películas o canciones, siendo ambos ámbitos similares entre sí pero distintos al educativo y estas pruebas con conjuntos temáticamente limitados podrían estar ocultando problemas en los algoritmos de recomendación.

Referencias

- Alshammari, M. (2016). *Adaptation based on learning style and knowledge level in e-learning systems* (Tesis Doctoral, University of Birmingham). Accedido el 2018/07/19 desde <http://etheses.bham.ac.uk/6702/>
- Brin, S., y Page, L. (1998, apr). The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1-7), 107–117. doi: 10.1016/S0169-7552(98)00110-X
- Brusilovsky, P., Cassel, L. N., Delcambre, L. M., Fox, E. A., Furuta, R., Garcia, D. D., ... Yudelson, M. (2010, jan). Social navigation for educational digital libraries. *Procedia Computer Science*, 1(2), 2889–2897. doi: 10.1016/S1877-0509(10)00330-3
- Caudill, J. (2011). Using OpenCourseWare to Enhance On-Campus Educational Programs. En *Proceedings of TCC Worldwide Online Conference 2011* (pp. 43–47). TCC Hawaii. Accedido el 2018/06/19 desde <https://www.learntechlib.org/p/43742>
- Chicaiza, J., Piedra, N., Lopez-Vargas, J., y Tovar-Caro, E. (2017, apr). Recommendation of open educational resources. An approach based on linked open data. En *2017 IEEE Global Engineering Education Conference (EDUCON)* (pp. 1316–1321). IEEE. doi: 10.1109/EDUCON.2017.7943018
- Cho, J., Garcia-Molina, H., y Page, L. (1998, apr). Efficient crawling through URL ordering. *Computer Networks and ISDN Systems*, 30(1-7), 161–172. doi: 10.1016/S0169-7552(98)00108-1
- Christudas, B. C. L., Kirubakaran, E., y Thangaiah, P. R. J. (2018, feb). An evolutionary approach for personalization of content delivery in e-learning systems based on learner behavior forcing compatibility of learning materials. *Telematics and Informatics*, 35(3), 520–533. doi: 10.1016/j.tele.2017.02.004
- CK-12 Foundation. (n.d.). *CK-12 Foundation*. Accedido el 2018/06/19 desde <https://www.ck12.org/student/>
- Drachler, H., Verbert, K., Santos, O. C., y Manouselis, N. (2015). Panorama of Recommender Systems to Support Learning. En *Recommender Systems Handbook* (pp. 421–451). Boston, MA: Springer US. doi: 10.1007/978-1-4899-7637-6_12
- Duffin, J., y Muramatsu, B. (2008). OER recommender: Linking NSDL pathways and OpenCourseWare repositories. *Proceedings of the ACM International Conference on Digital Libraries*(2007), 449 –. doi: 10.1145/1378889.1378994
- Felder, R. M., y Silverman, L. K. (1988). Learning and Teaching Styles in Engineering Education. *Engineering Education*, 78(7), 674–681.
- FlatWorld. (n.d.). *FlatWorld*. Accedido el 2018/06/19 desde <https://catalog.flatworldknowledge.com/>
- Fundación Wikimedia. (n.d.). *Wikipedia*. Accedido el 2018/06/19 desde <https://www.wikipedia.org/>

- Gasparetti, F., De Medio, C., Limongelli, C., Sciarrone, F., y Temperini, M. (2018, may). Prerequisites between learning objects: Automatic extraction based on a machine learning approach. *Telematics and Informatics*, 35(3), 595–610. doi: 10.1016/j.tele.2017.05.007
- Google. (n.d.). *YouTube Data API | Google Developers*. Accedido el 2018/06/19 desde <https://developers.google.com/youtube/v3/>
- Hajri, H., Bourda, Y., y Popineau, F. (2017, jul). MORS: A System for Recommending OERs in a MOOC. En *Proceedings - IEEE 17th International Conference on Advanced Learning Technologies, ICALT 2017* (pp. 50–52). IEEE. doi: 10.1109/ICALT.2017.89
- Hanna, D., Abhari, A., y Ferworn, A. (2017). Comparing quantitative and comment-based ratings for recommending open educational resources. En *Proceedings of the 20th Communications & Networking Symposium* (p. 112). Society for Computer Simulation International. Accedido el 2018/06/19 desde <https://dl.acm.org/citation.cfm?id=3107983>
- Karpathy, A., y Fei-Fei, L. (2017, apr). Deep Visual-Semantic Alignments for Generating Image Descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4), 664–676. doi: 10.1109/TPAMI.2016.2598339
- Keskin, S., Sahin, M., Ozgur, A., y Yurdugul, H. (2016, oct). Online Learners' Navigational Patterns Based on Data Mining in Terms of Learning Achievement. En *International Conference on Cognition and Exploratory Learning in the Digital Age (CELDA)*. Mannheim: International Association for the Development of the Information Society. Accedido el 2018/06/17 desde <https://eric.ed.gov/?id=ED571417>
- Koren, Y., y Bell, R. (2011). Advances in Collaborative Filtering. En *Recommender Systems Handbook* (pp. 145–186). Boston, MA: Springer US. doi: 10.1007/978-0-387-85820-3_5
- Koren, Y., Bell, R., y Volinsky, C. (2009, aug). Matrix Factorization Techniques for Recommender Systems. *Computer*, 42(8), 30–37. doi: 10.1109/MC.2009.263
- Lops, P., de Gemmis, M., y Semeraro, G. (2011). Content-based Recommender Systems: State of the Art and Trends. En *Recommender Systems Handbook* (pp. 73–105). Boston, MA: Springer US. doi: 10.1007/978-0-387-85820-3_3
- Manouselis, N., Drachsler, H., Verbert, K., y Santos, O. C. (2010). Workshop on recommender systems for technology enhanced learning. En *Proceedings of the fourth ACM conference on Recommender systems - RecSys '10* (p. 377). New York, New York, USA: ACM Press. doi: 10.1145/1864708.1864797
- Manouselis, N., Drachsler, H., Vuorikari, R., Hummel, H., y Koper, R. (2011). Recommender Systems in Technology Enhanced Learning. En *Recommender Systems Handbook* (pp. 387–415). Boston, MA: Springer US. doi: 10.1007/978-0-387-85820-3_12
- MERLOT. (n.d.). *MERLOT*. Accedido el 2018/06/19 desde <https://www.merlot.org/merlot/index.htm>
- Mishra, S. (2017, sep). Open educational resources: removing barriers from within. *Distance Education*, 38(3), 369–380. doi: 10.1080/01587919.2017.1369350
- MIT. (n.d.). *MIT OpenCourseWare | Free Online Course Materials*. Accedido el 2018/06/19 desde <https://ocw.mit.edu/index.htm>
- Mödritscher, F. (2010, jan). Towards a recommender strategy for personal learning environments. *Procedia Computer Science*, 1(2), 2775–2782. doi: 10.1016/J.PROCS.2010.08.002
- Möller, E., Stallman, R., Lessig, L., Hill, B. M., y Beesley, A. (n.d.). *Definition of Free Cultural Works*. Accedido el 2018/06/08 desde <https://freedomdefined.org/Definition>
- Najork, M., y Heydon, A. (2002). High-Performance Web Crawling. En (pp. 25–45). Springer, Boston, MA. doi: 10.1007/978-1-4615-0005-6_2
- OER Commons. (n.d.). *OER Commons*. Accedido el 2018/06/19 desde <https://www.oercommons.org/>

-
- OER Commons. (2017). *OER Commons Platform Terms of Use*. Accedido el 2018/06/19 desde <https://www.oercommons.org/terms>
- Olston, C., y Najork, M. (2010). Web Crawling. *Foundations and Trends in Information Retrieval*, 4(3), 175–246. doi: 10.1561/15000000017
- Open Knowledge International. (n.d.). *Open Definition 2.1 - Defining Open in Open Data, Open Content and Open Knowledge*. Accedido el 2018/06/08 desde <https://opendefinition.org/od/2.1/en/>
- Paredes, P., y Rodriguez, P. (2004, feb). A Mixed Approach to Modelling Learning Styles in Adaptive Educational Hypermedia. *Advanced Technology for Learning*, 1(4). doi: 10.2316/Journal.208.2004.4.208-0823
- Romero Pelaez, A., y Alarcon, P. P. (2017). Metadata quality assessment metrics into OCW repositories. En *ICETC2017* (pp. 253–257). New York, New York, USA: ACM Press. doi: 10.1145/3175536.3175579
- Salton, G., Fox, E. A., y Wu, H. (1983, nov). Extended Boolean Information Retrieval. *Communications of the ACM*, 26(11), 1022–1036. doi: 10.1145/182.358466
- Santos, O. C., y Boticario, J. G. (2010, jan). Modeling recommendations for the educational domain. *Procedia Computer Science*, 1(2), 2793–2800. doi: 10.1016/j.procs.2010.08.004
- Shelton, B. E., Duffin, J., Wang, Y., y Ball, J. (2010, jan). Linking open course wares and open education resources: creating an effective search and recommendation system. *Procedia Computer Science*, 1(2), 2865–2870. doi: 10.1016/J.PROCS.2010.08.012
- Sicilia, M.-Á., García-Barriocanal, E., Sánchez-Alonso, S., y Cechinel, C. (2010, jan). Exploring user-based recommender results in large learning object repositories: the case of MERLOT. *Procedia Computer Science*, 1(2), 2859–2864. doi: 10.1016/J.PROCS.2010.08.011
- Stallman, R. M. (2002). What is free software. *Free Society: Selected Essays of*, 23.
- UNESCO. (2012). 2012 Paris OER Declaration. *World Open Educational Resources (OER) Congress*. Accedido el 2018/06/06 desde http://www.unesco.org/new/fileadmin/MULTIMEDIA/HQ/CI/CI/pdf/Events/English_Paris_OER_Declaration.pdf
- Wang, Y., y Sumiya, K. (2010, jan). Semantic ranking of lecture slides based on conceptual relationship and presentational structure. *Procedia Computer Science*, 1(2), 2801–2810. doi: 10.1016/J.PROCS.2010.08.005
- Wiley, D. (2010, jul). Openness as Catalyst for an Educational Reformation. *All Faculty Publications*. Accedido el 2018/06/08 desde <https://scholarsarchive.byu.edu/facpub/95>
- Wiley, D., Bliss, T. J., y McEwen, M. (2014). Open Educational Resources: A Review of the Literature. *Handbook of Research on Educational Communications and Technology*, 781–789. doi: 10.1007/978-1-4614-3185-5_63
- Wu, C.-Y., Alvino, C. V., Smola, A. J., y Basilico, J. (2016). Using Navigation to Improve Recommendations in Real-Time. En *Proceedings of the 10th ACM Conference on Recommender Systems - RecSys '16* (pp. 341–348). New York, New York, USA: ACM Press. doi: 10.1145/2959100.2959174
- xkcd. (n.d.). *xkcd: Standards*. Accedido el 2018/06/19 desde <https://xkcd.com/927/>
- YouTube. (2009). *Official YouTube Blog: Five Stars Dominate Ratings*. Accedido el 2018/06/17 desde <https://youtube.googleblog.com/2009/09/five-stars-dominate-ratings.html>