UNIVERSIDAD AUTÓNOMA DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

Grado en INGENIERÍA INFORMÁTICA

TRABAJO FIN DE GRADO

CORRECCIÓN ORTOGRÁFICA PARA DISLEXIA CON REDES NEURONALES

Souad Mbarki
Tutor: Jordi Porta
Ponente: Gonzalo Martínez Muñoz

Septiembre 2019

CORRECCIÓN ORTOGRÁFICA PARA DISLEXIA CON REDES NEURONALES

AUTOR: Souad Mbarki
TUTOR: Jordi Porta

Dpto. de Ingeniería Informática.
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Septiembre de 2019

Resumen

Este Trabajo Fin de Grado tiene como objetivo diseñar un algoritmo basado en redes neuronales capaz de corregir palabras o secuencias multipalabra escritas con errores disléxicos. Dicho algoritmo estará desarrollado con la ayuda de la biblioteca de redes neuronales Keras, utilizando el lenguaje de programación Python. Para este fin, se analizarán dos bases de datos diferentes, la primera en español y la segunda en alemán, obtenidas de dos investigaciones sobre dislexia. A lo largo de este TFG se explicaran los métodos utilizados para la generación de dicho algoritmo, el cual estará basado en redes neuronales recurrentes, así como se explicará el procesamiento de los datos y se analizarán los resultados obtenidos.

Abstract

This End-of-Grade Project aims to design an algorithm based on neural networks that corrects words or multi-word sentences written with dyslexic errors. This algorithm will be developed with the help of Keras, a neural network library, using Python as programming language. For this, two different databases will be analyzed, the first in Spanish and the second in German, obtained from two investigations on dyslexia. Throughout this End-of-Grade Project, the methods used for the generation of the algorithm, based on recurrent neural networks, will be explained, as well as the data processing and the results obtained will be analyzed.

Palabras clave

Dislexia, LSTM, Dataset, RNN, red neuronal, Keras, dropout, tamaño de lote, modelo Seq2seq, sobreajuste, Búsqueda en rejilla.

Keywords

Dyslexia, LSTM, Dataset, RNN, neuronal network, Keras, dropout, batch size, Seq2seq model, overfitting, grid search.

Agradecimientos

En primer lugar, me gustaría agradecer a mis padres y hermanos su paciencia infinita y sus ganas siempre de hacerme reír. Gracias por haberme acompañado en esta etapa larga e intensa.

A todos los compañeros de batallas en las prácticas, proyectos y trabajos, por haber conseguido superar esta etapa con éxito y con ganas de comernos el mundo. En especial a Jimena, William, Poderoso y JuanFran por ser un apoyo y desahogo constante.

A Jordi Porta y Gonzalo Martínez Muñoz, por la oportunidad y ganas de ayudarme a realizar este trabajo de fin de grado.

Y por último, pero no menos importante, a Diego, siempre en la cafetería repartiendo cariño, café y sesiones de psicología gratis.

INDICE DE CONTENIDOS

1 Introducción 1
1.1 Motivación 1
1.2 Objetivos 1
1.3 Organización de la memoria 1
2 Estado de la cuestión 3
2.1 SeeWord 3
2.2 Firefixia 3
2.3 IDEAL eBook Reader 3
2.4 Dytective 3
2.5 Disanedu 3
2.6 Lixta 3
2.7 Deslixate 3
3 Materiales 5
3.1 Español 5
3.2 Alemán 6
3.3 Análisis de la base de datos en español 7
3.4 Análisis de la base de datos en alemán 10
4 Metodología y modelos 13
4.1 Redes neuronales 13
4.1.1 Clasificación de las redes neuronales 14
4.1.2 Redes neuronales recurrentes (RNN). 15
4.2 Representación de datos 18
4.3 Modelos Seq2Seq 19
4.4 Parámetros e hiperparámetros de los modelos. 21
5 Experimentos y resultados 23
5.1 ECHO 25
5.1.1 Echo Español 25
5.1.1 Echo Alemán 27
5.2 DYSLEXIA-ES 28
5.3 DYSLEXIA-DE 30
6 Conclusiones y trabajo futuro 33
6.1 Conclusiones 33
6.2 Trabajo futuro 33
Referencias 35
Glosario 37
Anexos I
A Ejemplo de fichero de salida. Español I
B Extracto de la base de datos echo español. V
C Extracto de la base de datos echo alemán. ..... - 1 -

INDICE DE FIGURAS

FIGURA 1. EJEMPLO DE TEXTO CORREGIDO DE UN ALUMNO CON DISLEXIA (15 AÑOS) DE [7]. 6

FIGURA 2. BASE DE DATOS DE ESPAÑOL DE [6]..... 6

FIGURA 3. BASE DE DATOS CON PALABRAS EN ALEMÁN [5]. 7

FIGURA 4. PORCENTAJE DE DISTRIBUCIÓN DE LAS PALABRAS DE LA BASE DE DATOS Y EL PORCENTAJE DE ERRORES. 7

FIGURA 5. NEURONA CON 4 ENTRADAS 13

FIGURA 6. EJEMPLO DE RED NEURONAL CON UNA CAPA OCULTA. 14

FIGURA 7. ESQUEMA DE UNA LSTM. 16

FIGURA 8. ESQUEMA DE GRU. 18

FIGURA 9. EJEMPLO DE CODIFICACIÓN ONE-HOT. 19

FIGURA 10. EJEMPLO DE LA ARQUITECTURA CODIFICADOR-DECODIFICADOR. 20

FIGURA 11. EJEMPLO DE LA ARQUITECTURA CODIFICADOR-DECODIFICADOR. 20

FIGURA 12. EJEMPLO DE LA ARQUITECTURA CODIFICADOR-DECODIFICADOR CON UN MECANISMO DE ATENCIÓN. 20

FIGURA 13. EJEMPLO DE SCRIPT DE BÚSQUEDA EN REJILLA. 23

FIGURA 14. MODELO 1. 24

FIGURA 15. MODELO2. 24

FIGURA 16.1. RESULTADOS CON BASE DE DATOS ECHO ESPAÑOL. 25

FIGURA 17. PORCENTAJE DE ACIERTOS EN ENTRENAMIENTO Y VALIDACIÓN. TAMAÑO DE LOTE 8 Y 250 NEURONAS. 26

FIGURA 18. PORCENTAJE DE ACIERTOS EN ENTRENAMIENTO Y VALIDACIÓN. TAMAÑO DE LOTE 8 Y 500 NEURONAS. 27

FIGURA 19.1. RESULTADOS CON BASE DE DATOS ECHO ALEMÁN. 27

FIGURA 20. PORCENTAJE DE ACIERTOS. ECHO ALEMÁN. TAMAÑO DE LOTE 8 Y 150 NEURONAS... 28

FIGURA 21.1: RESULTADOS CON BASE DE DATOS ESPAÑOL. 28

FIGURA 22. PORCENTAJE DE ACIERTOS. ESPAÑOL. TAMAÑO LOTE 8 Y 150 NEURONAS. 29

FIGURA 23. PORCENTAJE DE ACIERTOS. ESPAÑOL. TAMAÑO LOTE 8, 150 NEURONAS Y DROPOUT DEL 20%..... 30

FIGURA 24.1: RESULTADOS CON BASE DE DATOS ALEMÁN. 30

FIGURA 25. PORCENTAJE DE ACIERTOS. ESPAÑOL. TAMAÑO LOTE 8 Y 150 NEURONAS. 31

INDICE DE TABLAS

TABLA 1 . PORCENTAJE DE ERRORES 8

TABLA 2. TIPOS DE ERRORES DISLÉXICOS. 9

TABLA 3. FRECUENCIA DE PALABRAS REPETIDAS. 9

TABLA 4. REPETICIÓN DE PALABRAS EN ALEMÁN. 11

TABLA 5. FRECUENCIA DE LAS PALABRAS EN ALEMÁN. 11

TABLA 6. PORCENTAJE DE ERROR ALEMÁN. 12

TABLA 7. EJEMPLO DE ERRORES ESPAÑOL. 29

1 Introducción

1.1 Motivación

La dislexia, considerada la dificultad de aprendizaje más común [10], es un trastorno del aprendizaje, precisión y fluidez, de origen neurológico, que afecta a la comprensión lectora y la expresión escrita de las personas que la tienen.
Por otra parte, la dislexia no afecta a la inteligencia general de las personas que la tienen, pero una de las consecuencias derivadas de este trastorno es un vocabulario reducido debido a la escasa experiencia lectora [7].
La dislexia es categorizada como un trastorno del neurodesarrollo llamada "trastorno específico del aprendizaje" [1]. Este trastorno se puede clasificar en dos tipos:

- Adquirido: aparece a causa de una lesión cerebral concreta.
- Evolutivo: no hay una lesión cerebral concreta que la haya producido, es adquirida genéticamente. Es la más frecuente en el ámbito escolar.

El porcentaje de la población mundial que sufre este trastorno se estima sobre el 10 % [11]. Según la Academia Nacional de Ciencias Americana, entre el 10 y el 17.5 % de la población de Estados Unidos tiene dislexia [2]. En cuanto a España, se estima que el porcentaje de personas con dislexia está entre el 5 y el 10 % [29].

Por otro lado y según *Meng* [15], la tasa de errores de origen disléxico varía de forma considerable dependiendo del lenguaje. Las personas que tienen dislexia cometen más errores de tipo fonológico y léxico. A pesar de que la dislexia es considerada un problema de inversión de las letras en una palabra y números, tal y como se ha comentado anteriormente, sólo un 30 % de las personas que tienen dislexia revelan ese problema [7], lo que explica que gran parte de los errores reflejados en la base de datos, que se han usado y que se comentarán más tarde, son debidos a errores ortográficos, que niños de edades tempranas puedan cometer y no expresamente errores disléxicos.

1.2 Objetivos

A lo largo del desarrollo de este TFG se abordará la construcción de un modelo, aplicando algoritmos de redes neuronales, para generar hipótesis sobre palabras aisladas. Para la detección y corrección mencionadas, se ha optado por usar redes neuronales recurrentes (RNN) que son un tipo de redes neuronales con aprendizaje automático supervisado. El cual se va a desarrollar, posteriormente y con más detalle en el apartado de *Metodología y modelos*.

1.3 Organización de la memoria

En cuanto a la estructura de este documento, se ha organizado de la siguiente manera:

- Capítulo 1: Motivación. Objetivos. Organización de la memoria.
- Capítulo 2: Estado de la cuestión.
- Capítulo 3: Materiales. Descripción de los datos. Base de datos en español. Descripción de los datos. Base de datos en alemán.
- Capítulo 4: Metodología y modelos. Redes neuronales. Representación de datos. Preparación de los datos. Modelos Seq2Seq. Parámetros e hiperparámetros de los modelos.
- Capítulo 5: Experimentos y resultados. ECHO. DYSLEXIA-ES. DYSLEXIA-DE.
- Capítulo 6: Conclusiones y trabajo futuro.

2 Estado de la cuestión

Con el fin de desarrollar un algoritmo para la detección y corrección de las palabras aisladas con errores ortográficos producidos por personas con dislexia, se ha realizado una búsqueda de las aplicaciones y herramientas con una funcionalidad similar. La mayoría de aplicaciones encontradas para la dislexia tienen otros objetivos distintos que la corrección y ninguna está basada en redes neuronales. Las más usadas actualmente son [9, 28]:

2.1 SeeWord

Es un entorno de procesamiento de palabras, que dispone de asistencia tanto en la escritura de textos como en su lectura por los usuarios [26].

2.2 Firefixia

Es una extensión del navegador Mozilla Firefox, diseñada para para servir de ayuda y apoyo las personas con dislexia [27].

2.3 IDEAL eBook Reader

Es un lector de libros electrónicos que dispone de configuraciones para ajustar la vista de los libros a personas con dislexia. Se puede descargar en el siguiente enlace: https://play.google.com/store/apps/details?id=org.easyaccess.epubreader.

2.4 Dytective

Consiste en una serie de tests mediante los cuales y acorde a la edad del usuario es capaz de detectar si este tiene riesgo de tener dislexia. Esta aplicación está disponible para descargar en la página https://www.changedyslexia.org y se tanto para las tecnologías Android como iOS.

2.5 Disanedu

Es una aplicación interactiva para mejorar la competencia lectora, a través de su microsite, se trabaja la velocidad lectora, la comprensión de textos y el perfeccionamiento palabra por palabra. Permite la modificación de los textos para adaptarlos a cualquier necesidad e incluso idioma.

2.6 Lixta

Una aplicación española desarrollada por Encódigo en 2014 para facilitar la memorización de vocabulario o corregir las faltas de ortografía frecuentes.

2.7 Deslixate

Es una aplicación que permite obtener un pre-diagnóstico de la dislexia en niños de entre siete y 12 años.

3 Materiales

Los algoritmos basados en redes neuronales necesitan un conjunto de datos sobre el que realizan su entrenamiento con el fin de producir generalizaciones, encontrar patrones y poder detectar y corregir los errores aprendidos en otros conjuntos de datos. Para ello, se han utilizado dos bases de datos que serán expuestas en las siguientes secciones.

3.1 Español

El primer conjunto que se ha utilizado para el análisis es una base de datos en español, fue construida gracias al artículo de los investigadores Luz Rello, Ricardo Baeza-Yates y Joaquim Llisterri "*DysList: An Annotated Resource of Dyslexic Errors*" [7]. Cabe destacar que, para la creación de esta base de datos, los investigadores mencionados, se basaron en dos trabajos anteriores con textos en inglés y español:

- El primero, realizado en 2007, contiene 3134 palabras en inglés y 363 errores, que posteriormente fue ampliado a 21 524 palabras, los cuales contenían 2654 errores, con más de 800 errores de palabras reales. Fue compuesto por ejercicios y muestras de error realizados por estudiantes de secundaria. [16].
- El segundo, es un trabajo realizado gracias a 16 textos de 1057 palabras y 157 errores no repetidos, escritos por niños de edades entre 6 y 15 años que tienen dislexia [17].

En total, se compilaron 83 textos, pertenecientes a 54 ensayos y ejercicios escritos por niños con dislexia de edad entre 6 y 15 años. Estos textos y ensayos, fueron propuestos por los profesores de varios colegios a sus alumnos. Un ejemplo de estos textos manuscritos se muestra en la Figura 1.

Por otra parte, la base de datos arriba mencionada, está estructurada de la siguiente manera: Contiene un total de 73 columnas y 1171 palabras (filas), las dos primeras son identificadores de las palabras. Las dos siguientes filas, la tercera y la cuarta, como se puede observar en la Figura 2, son las palabras corregidas y con error, y las dos siguientes son la frecuencia con las que éstas se han observado. Por último, otra columna que tuvimos en cuenta para el desarrollo de este TFG es la columna 14 que representa la distancia de Levenshtein, es decir el mínimo número de cambios que hay que introducir en la palabra mal escrita para su corrección. El resto de columnas, no se usan en este TFG.

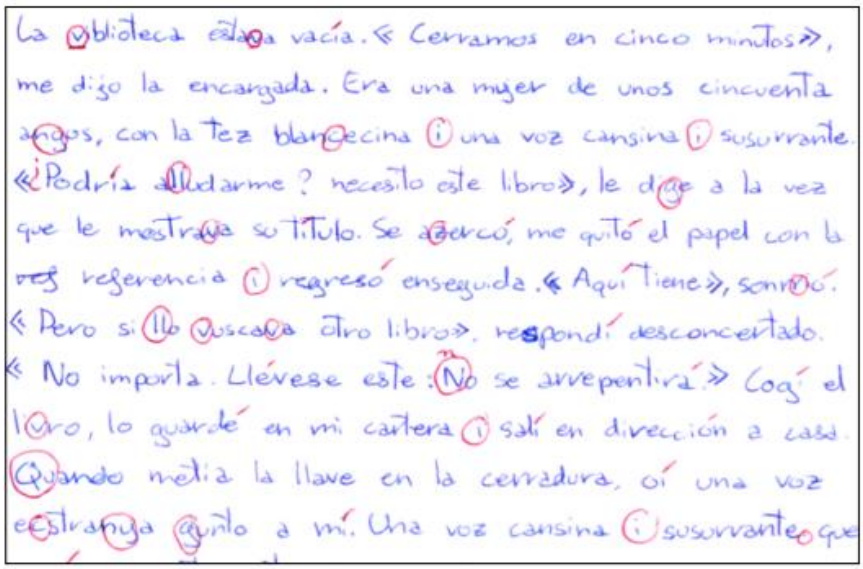
La biblioteca estaba vacía. «Cerramos en cinco minutos», me dijo la encargada. Era una mujer de unos cincuenta años, con la tez blanquecina y una voz cansina y susurrante. «¿Podría ayudarme? necesito este libro», le dije a la vez que le mostraba su título. Se acercó, me quitó el papel con la referencia y regresó enseguida. «Aquí tiene», sonrió. «Pero si no buscaba otro libro», respondí desconcertado. «No importa. Llévese este: No se arrepentirá.» Cogí el libro, lo guardé en mi cartera y salí en dirección a casa. Cuando metía la llave en la cerradura, oí una voz extraña cerca de mí. Una voz cansina y susurrante, que

Figura 1. Ejemplo de texto corregido de un alumno con dislexia (15 años) de [7].

A	B	C	D	E	F
ID	ID_JLL	Correct_Word	Error_Word	Correct_Word_F	Error_Word_ Fre
326	375	de~hierbabuena	dehiervabuena	8571	0
1075	1230	te~pasa	tepassa	130386	2
292	338	corazón~a~cien	corazónacies	185	0
719	841	me~regaló	merregaló	25161	0
1021	1174	se~enfado	senfado	1609	4
1063	1218	tal~vez	talvez	2306709	195580
188	213	voy~a	boia	5260183	1612
731	854	mi~tío	mitio	50245	303
359	416	de~vidrio	devidreo	714321	0

Figura 2. Base de datos de español de [6].

3.2 Alemán

Este segundo conjunto de datos de [5] contiene producidas palabras o secuencias de palabras en alemán. En total, el conjunto de datos inicial, se compone de 25 columnas y 1021 filas, que son el número total de palabras que contien la base de datos. La estructura general de la base de datos es muy parecida a la primera, ya que esta investigación, fue llevada a cabo más tarde por los mismos investigadores**,** Luz Rello, Maria Rauschenberger, Silke Fuchsel y Jorg Thomaschewski. La primera columna es el identificador que se le ha asignado a cada palabra, la segunda y la tercera, son las palabras correcta y con error respectivamente. El resto reflejan otros datos como el tipo de error, la distancia de Levenshtein. Las longitudes de las palabras y otros datos que no son usados para este TFG. Puede verse un fragmento en la Figura 3.
Esta base de datos ha sido construida gracias a la colección de 47 textos escritos por estudiantes de entre 8 y 17 años [8].

A	B	C	D	E	F
ID	Correct_Word	Error_Word	Damerau-Lenver	Correct_Word_F	Error_Word_ Fre
1	Bank	Bak	1	305.000.000	436
2	Bus	Pos	2	283.000.000	32.000.000
3	Bus	Pus	1	283.000.000	419
4	Bus	Bos	1	283.000.000	1.020.000
5	Nest	Nes	1	11.000.000	21.600.000
6	Lšwe	Lowe	1	8.980.000	458
7	Tisch	Tšsch	1	52.000.000	18.9
8	Bein	Pain	2	23.100.000	481
9	Bein	Pein	1	23.100.000	481

Figura 3. Base de datos con palabras en alemán [5].

3.3 Análisis de la base de datos en español

En este apartado se realizará un análisis cualitativo de la base de datos, como la frecuencia de las palabras y otras propiedades, que serán de utilidad a la hora de interpretar los resultados obtenidos en los experimentos que se han realizado durante la creación de este TFG.

En cuanto al análisis de datos, de los 83 textos arriba mencionados, los creadores de la base de datos extrajeron una lista con 887 palabras con diferentes errores, exceptuando errores de acentuación, ya que han considerado que los niños de edades tempranas (entre 6 y 15 años) aún cometen dichos errores, es decir, que son errores propios de niños de su edad. Además, se ha observado que 678 palabras están mal escritas con diferentes variantes (las palabras *sigilosamente* y *accesibilidad* son las que más variantes tienen). De la lista, también se extrajeron 894 pares de palabras mal escritas con un total de 1171 errores.

Por otra parte, la longitud de las palabras mal escritas difiere entre 1 y 20 letras, con una media de 7,47 letras por palabra. En la Figura 4, se muestra una gráfica con los valores de las longitudes de las palabras y el porcentaje de distribución de la posición donde el error aparece.

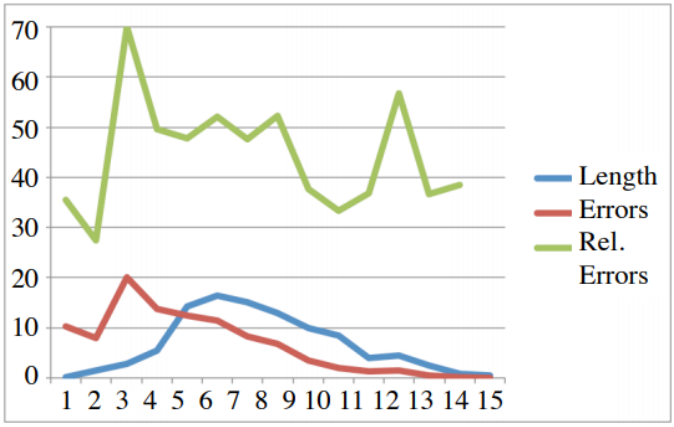

Figura 4. Porcentaje de distribución de las palabras de la base de datos y el porcentaje de errores.

Con el fin de una comprensión mejor de los errores que se presentan en las bases de datos utilizadas, se desarrollarán los tipos de errores de origen disléxico clasificados en uno de los trabajos que se han tenido en cuenta para construir esta base de datos [17]:

- Errores disléxicos basados en el número de diferencias de la palabra deseada:
    1. *Simple errors*. Difieren de la palabra correcta en una sola letra. Y pueden ser de debidos a:
        a) Sustitución.
        b) Inserción.
        c) Omisión.
        d) Trasposición.
    2. *Multi-errors*. Se diferencian en más de una letra de la palabra destino (*target word*).
    3. *Word boundary errors* (Errores de límite de palabra). Son errores que representan casos especiales de errores de omisión e inserción. Por ejemplo, omitir un espacio o insertarlo.

- Errores disléxicos basados en su correspondencia con palabras existentes:
    1. *Real-word errors*. Faltas de ortografía que generan otra palabra correcta. Por ejemplo, siendo bien la palabra correcta se ha escrito como ven, que es errónea, pero genera una palabra correcta en el lenguaje.
    2. *Non-word errors*. Faltas de ortografía que no resultan en otra palabra correcta.

- Errores disléxicos de primera letra:
    1. *First letter errors*. Por ejemplo *no (*know*), del inglés.

En la Tabla 1 se muestra el porcentaje de cada uno de los respectivos errores encontrados en el trabajo mencionado [17]:

Categoría	**Porcentaje**
Errores simples	67
Multi-errores	23
Errores de límite de palabra.	10
Errores de palabra real	21
Errores palabras incorrecta	79
Errores de primera letra	11

Tabla 1 . Porcentaje de errores

Por otra parte y analizando el artículo [8], se han extraído otros datos de gran importancia, relativos a la clasificación de los diferentes tipos de error que se han producido y sus porcentajes. A continuación, se van a desarrollar dichos errores:

- Sustitución: Sustituir un letra o más letras por otras.
- Inserción: Añadir una letra o más letras a la palabra mal escrita
- Omisión: Eliminar u omitir letras.
- Transposición: Cambiar el orden de las letras contiguas.

El porcentaje de estos errores en la base de datos es el siguiente:

Tipo de error	**Porcentaje**
Sustitución	58.84
Inserción	13.40
Omisión	26.30
Transposición	1.45

Tabla 2. Tipos de errores disléxicos.

Este conjunto de palabras o secuencias de palabras está compuesto, como ya se ha mencionado, por 1171 palabras, de las cuales 435 palabras no son repetidas, es decir, sólo aparecen una vez y sin variantes. El resto, son 243 palabras que derivan en 736 palabras escritas de forma errónea mínimo dos veces. La frecuencia con la que aparecen algunas de las palabras se muestra en la Tabla 3. Como se puede observar en la misma, las palabras "*accesibilidad*" y "*sigilosamente*" son las que más variantes tienen.

Correct_Word	COUNTA of	rosita	5	hiciste	4	perfecto	3
sigilosamente	12	payaso	5	hermano	4	pensaba	3
accesibilidad	12	palabras	5	hacer	4	palabra	3
proceso	9	mucho	5	hablar	4	paisaje	3
huevos	9	ilustrador	5	giraba	4	octubre	3
porcentajes	8	hace	5	genial	4	noviembre	3
conversación	8	gusta	5	galaxia	4	necesita	3
verdadera	7	guerra	5	frecuencias	4	miércoles	3
también	7	beber	5	extensión	4	literales	3
pequeño	7	aquí	5	envuelve	4	justo	3
habitación	7	año	5	dibujo	4	jueves	3
berenjena	7	alrededor	5	convencerlos	4	israelíes	3
arquitectura	7	agua	5	cerca	4	invariables	3
necesitaba	6	adjetivo	5	bien	4	imágenes	3
interior	6	significativo	4	ayuntamiento	4	hube	3
garbanzos	6	servicios	4	voy~a	3	hay	3
excepto	6	policía	4	verbos	3	gimnasio	3
alienígena	6	pequeña	4	veinte	3	felicidad	3
adverbios	6	peligro	4	significativos	3	extraña	3
voy	5	participantes	4	se~ha	3	estaba	3
variable	5	noche	4	se~enfado	3	escribe	3
señor	5	jugar	4	presente	3	era	3
salida	5	huele	4	perros	3	envía	3

Tabla 3. Frecuencia de palabras repetidas.

Haciendo un análisis más profundo de la base de datos, se ha observado que uno de los errores más comunes es confundir la letra "b" con la letra "d" en múltiples palabras, por ejemplo escribir "*deber*" en vez de "*beber*", "*derengena*" en lugar de "*berenjena*" y "*bonde*" en vez de "*donde*". Esto es debido a que las personas con esta dificultad tienden a confundir la "b" por la "d" o la "p" por la "b".

Otra característica común que se ha podido observar, es escribir juntas secuencias de palabras que de forma correcta se escriben separadas. En la Tabla 2, se muestran algunos ejemplos. En total se han encontrado 74 secuencias de palabras o multipalabras, de las cuales 29 se repiten al menos dos veces.

Por otra parte, se ha observado el caso contrario al explicado en el párrafo anterior, que consiste en escribir separadas secuencias de palabras que ortográficamente se escriben juntas, aunque este caso se ha dado con menos frecuencia que el anterior. En la Tabla 3, se muestran algunos ejemplos. En este caso, se ha encontrado 58 palabras escritas erróneamente como palabras separadas por un espacio.

Otros dos errores muy comunes que se han observado analizando la base de datos son confundir la letra "b" con la letra "v", y la letra "y" con la "ll". Estos errores se pueden considerar ortográficos también y no sólo disléxicos. En el primer error común, sustituir la "b" por la "v", se encuentran 100 casos, mientras que en el caso contrario se han encontrado 46 errores.

3.4 Análisis de la base de datos en alemán

Para el segundo conjunto de datos, en alemán, se ha realizado un análisis parecido al anterior. Los investigadores han adoptado un criterio de anotación de errores diferente al aplicado en el conjunto de datos del español, esto se debe a que el alemán tiene una ortografía y estructura silábica diferente, tal como han comentado los investigadores en su artículo [8].

Comenzando el análisis con la frecuencia de palabras repetidas, se ha encontrado que en esta base de datos existen varias palabras repetidas, hay un total 198 palabras con al menos dos variantes erróneas, y otras 316 palabras que se han escrito erróneamente sólo una vez. La frecuencia de repeticiones se muestra en la Tabla 4, así como las palabras con más frecuencia que se reflejan en la Tabla 5.

Frecuencia de repetición	**Número de palabras**
2	14
3	121
4	30
5	10
6	7

7	4
10	1
11	1

Tabla 4. Repetición de palabras en alemán.

Correct_Word	COUNTA of Corr
richtig	11
sieht	10
sofort	7
Schiedsrichter	7
Gewinnern	7
fliegt	7
vielschichtig	6
verschiedene	6
verfallene	6
Sportplatz	6
Spa§	6
meinen	6
KrŸcken	6
Hšhlenwanderun	6
gehen	6
frŸhstŸcken	6
ZugbrŸcke	5
VerkŠuferin	5
TischtennisschlŠ	5
Stra§enbahn	5
Spinnennetz	5
rissen	5
Rei§verschluss	5
Proviant	5
nicht Šrgern	5
MitschŸler	5
kontrollieren	5
Geburtstagsgesc	5
Fu§ballmannsch	5
aus Versehen	5
Arbeitsmaterialie	5
Anstoss	5
alle	5
zu tun	4
Vogelfutter	4
Vieh	4
versteckten	4

Tabla 5. Frecuencia de las palabras en alemán.

En cuanto a los tipos de error que se han podido contemplar, a parte de los errores comentados en el apartado anterior (sustitución, inserción, omisión y otros tipos de errores), se han definido dos tipos de error más específicos para el alemán [8]:

- Capital letter errors.
- Non-capital letter.

El porcentaje de estos errores se refleja en la Tabla 6.

Tipo de error	**Porcentaje**
Omisión	25
Sustitución	19
Multierror	16
Adición	9
Letra mayúscula	9
Letra minúscula	6
División de palabra	2

Transposición	0.9
Palabras sin espacio	0.3

Tabla 6. Porcentaje de error alemán.

El análisis cualitativo que se ha llevado a cabo en los apartados anteriores, se ha realizado con el fin de poder dar una interpretación correcta los resultados que generará la aplicación de los algoritmos que se han seleccionado para resolver el problema expuesto en este TFG.

4 Metodología y modelos

Para la realización de este TFG se ha optado por usar redes neuronales recurrentes, debido a su adecuación para el procesamiento del lenguaje natural (PLN).

4.1 Redes neuronales

Las redes neuronales artificiales son un algoritmo de aprendizaje automático supervisado. Estos algoritmos son capaces de producir generalizaciones a partir de la observación y el análisis de una serie de ejemplos dados. Las redes neuronales artificiales, como su nombre indica, son un modelo computacional inspirado en el comportamiento de las redes neuronales y su mecanismo de computación en el cerebro humano, en el cual, las neuronas son las unidades de computación que reciben entradas que tienen asociado un peso y generan salidas. Según este modelo, cada neurona multiplica su entrada por el peso correspondiente, sumando los resultados, se aplica una función no lineal al resultado final y se pasa a la salida correspondiente a dicha neurona. Todo este mecanismo se explica con más profundidad en apartados posteriores. La Figura 5 muestra un ejemplo del esquema de una neurona.

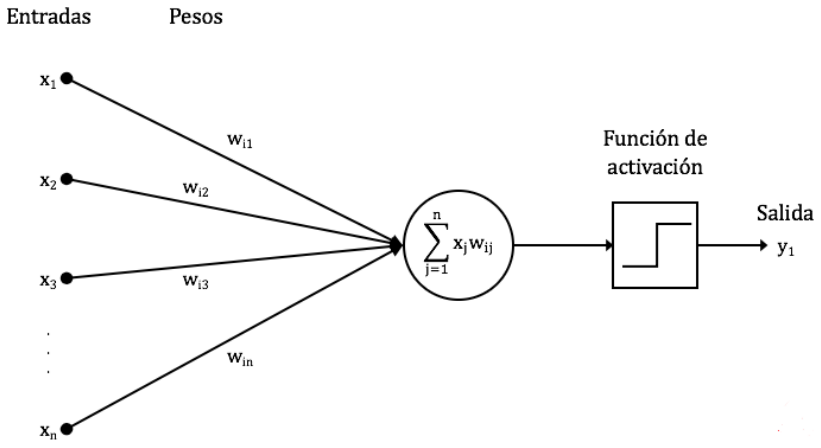

Figura 5. Neurona con 4 entradas

Las neuronas forman una red conectándose unas a otras y la salida que genera una de ellas puede ser la entrada de otra u otras neuronas. Además, una red neuronal puede estar formada por varias capas que pueden clasificarse de la siguiente manera:

- Capa de entrada: Compuesta por neuronas que reciben información o señales procedentes del entorno.
- Capas ocultas: Formadas por aquellas neuronas que no poseen conexión directa con el exterior, es decir, cuyas entradas provienen de capas anteriores y sus salidas se pasan como entrada a neuronas de capas posteriores (de salida).
- Capa de salida: Neuronas que proporcionan los valores de respuesta de la red neuronal.

En la Figura 6 se muestra un ejemplo de una red neuronal con una capa oculta. En dicha red, la salida de las neuronas que forman la capa de entrada son las entradas para las

neuronas de la capa oculta de la red, estas a su vez, genera una salida que puede ser la entrada de otra capa oculta.

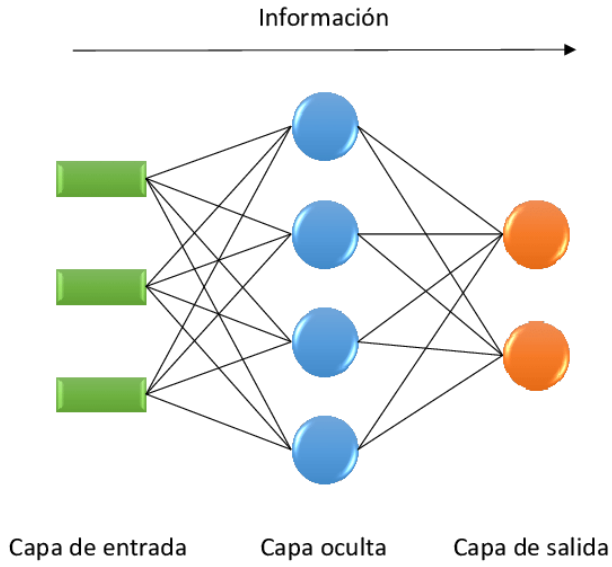

Figura 6. Ejemplo de red neuronal con una capa oculta.

4.1.1 Clasificación de las redes neuronales

Existen diferentes tipos de redes neuronales, por lo que pueden clasificarse de dos maneras, según la topología o estructura de red y según el método de aprendizaje

- Según su topología:
  La topología o estructura de una red consiste en la forma de organización de las capas que forman dicha red. Según esta clasificación se encuentran los siguientes tipos [21]:
  - **Perceptrón simple** (Red neuronal monocapa): consiste en una red neuronal artificial compuesta sólo por una capa de neurona de entrada y otra capa de salida. Es la red neuronal más simple.
  - **Perceptrón multicapa** (Red neuronal multicapa): consiste en una generalización del perceptrón, en vez de tener solo una capa de entrada, está dispone de un conjunto de capas intermedias o capas ocultas entre la capa de entrada y la de salida.
  - **Red neuronal convolucional** (CNN): En este tipo de redes, la red dispone de varias capas ocultas especializadas, es decir, las neuronas no están conectadas con todas las capas siguientes sino que, solo con un subgrupo de ellas, reduciendo el número de neuronas y la complejidad computacional del sistema.
  - **Red neuronal recurrente** (RNN): Este tipo de red no posee una estructura de capas, sino que permiten conexiones arbitrarias entre las neuronas, pudiendo crear ciclos, con lo que se consigue crear la temporalidad, permitiendo la retropropagación (*backpropagation*), donde la respuesta de la salida de la red puede ser la entrada de la misma. De esta manera se permite que la red tenga memoria.

- Según su método de aprendizaje:

- **Aprendizaje supervisado**: En este tipo de aprendizaje se controla el entrenamiento de la red por un supervisor, el cual determina la respuesta que se debe generar para cada entrada. El supervisor controla la salida y si esta no es correcta, modifica los pesos de las conexiones, con el fin de que la salida obtenida se aproxime a la deseada.
- **Aprendizaje no supervisado** o autosupervisado: En este tipo no hace falta supervisor para ajustar los pesos de la red.

4.1.2 Redes neuronales recurrentes (RNN).

El tipo de red neuronal artificial por el que se ha optado en este TFG son las redes neuronales recurrentes [16], estas funcionan como memorias asociativas, son una función que recibe como entrada una secuencia de datos de longitud arbitraria, de *n* vectores con d_{in} dimensiones ($x_{1:n}= x_1,...,x_n$) y devuelve un único vector con d_{out} dimensión (y_n). Este vector de salida, es usado para las predicciones futuras.

$$y_n = \mathrm{RNN}(x_{1:n})$$

$$x_i \in \mathbb{R}^{d_{in}} \quad y_n \in \mathbb{R}^{d_{out}}.$$

Existen diferentes tipos de redes neuronales recurrentes, dependiendo del tipo de operaciones que utilizan. En el siguiente párrafo se van a exponer los diferentes tipos de RNN y en profundidad las LSTM, que son las desarrolladas en los experimentos de este TFG.

- **CBOW como RNN** (continuous bag of words): Una CBOW está compuesta por palabras, donde cada una de estas es un vector. Esta arquitectura permite predecir la salida a partir de una ventana de palabras contexto (donde la predicción es independiente del orden de las palabras contexto) [19].

$$s_i = R_{\mathrm{CBOW}}(x_i, s_{i-1}) = s_{i-1} + x_i$$

$$y_i = O_{\mathrm{CBOW}}(s_i) = s_i$$

$$s_i, y_i \in \mathbb{R}^{d_s}, \; x_i \in \mathbb{R}^{d_s}.$$

- **RNN Simple** (SRNN): También conocida como la Red de Elman, es la red neuronal recurrente más simple que tiene en cuenta el orden de las palabras en una secuencia, es decir, su contexto. Tiene las siguiente forma:

$$s_i = R_{\mathrm{SRNN}}(x_i, s_{i-1}) = g(s_{i-1}W^s + x_i W^x + b)$$

$$y_i = O_{\mathrm{SRNN}}(s_i) = s_i$$

$$s_i, y_i \in \mathbb{R}^{d_s}, \; x_i \in \mathbb{R}^{d_x}, \; W^x \in \mathbb{R}^{d_x \times d_s}, \; W^s \in \mathbb{R}^{d_s \times d_s}, \; b \in \mathbb{R}^{d_s}.$$

Tanto el estado s_i como la entrada x_i son transformadas linealmente, sumado su resultado (añadiendo un sesgo b), y pasadas a una función de activación no lineal g (tanh). La salida y_i se obtiene mapeando el estado s_i. El problema de esta arquitectura es que cada estado s_i, es considerado como una memoria finita, de la que se lee y en la que escribe en cada paso

de la computación. No se puede controlar el acceso a memoria, este problema también es conocido como el problema de la desaparición del gradiente (vanishing gradient).

- **LSTM**: La arquitectura **Long Short-Term Memory**, usada para el problema propuesto en este TFG, fue introducida por *Hochreiter & Schmidhuber* (1997), y diseñadas explícitamente para resolver el problema de "desaparición del gradiente" o de acceso a memoria explicado en el párrafo anterior.
  Las LSTM dividen el vector de estado s_i en dos partes. Una primera parte es tratada como celdas de memoria, donde contiene tres puertas, las cuales controlan el modo en el que la información fluye dentro o fuera de la celda [20] y donde se preserva la memoria. De esta forma, la red es capaz de retener información de entradas anteriores en el tiempo y tener en cuenta dependencias temporales largas y el contexto.
  La segunda parte es la memoria en la que se trabaja en cada paso. Para cada entrada, una "puerta" es usada para decidir qué partes de la entrada deben ser escritas en la celda de memoria y que partes de la memoria en ese instante deben ser ignoradas. El esquema de una LSTM, se muestra en la siguiente figura (7), donde x_j es el valor de la secuencia en el instante j y h_{j-1} es la salida de la unidad LSTM en el paso anterior.

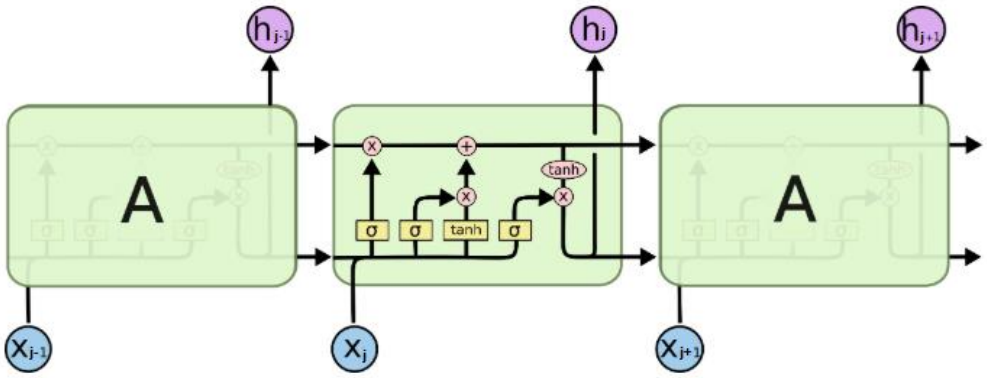

Figura 7. Esquema de una LSTM.

Matemáticamente la función LSTM se representaría de la siguiente manera:

$$
\begin{aligned}
s_j = R_{\text{LSTM}}(s_{j-1}, x_j) &= [c_j; h_j] \\
c_j &= f \odot c_{j-1} + i \odot z \\
h_j &= o \odot \tanh(c_j) \\
i &= \sigma(x_j W^{xi} + h_{j-1} W^{hi}) \\
f &= \sigma(x_j W^{xf} + h_{j-1} W^{hf}) \\
o &= \sigma(x_j W^{xo} + h_{j-1} W^{ho}) \\
z &= \tanh(x_j W^{xz} + h_{j-1} W^{hz})
\end{aligned}
$$

$$
y_j = O_{\text{LSTM}}(s_j) = h_j
$$

$$
s_j \in \mathbb{R}^{2 \cdot d_h}, \; x_i \in \mathbb{R}^{d_x}, \; c_j, h_j, i, f, o, z \in \mathbb{R}^{d_h}, \; W^{xo} \in \mathbb{R}^{d_x \times d_h}, \; W^{ho} \in \mathbb{R}^{d_h \times d_h}.
$$

Siendo:

cj: El nuevo estado.
hj: La salida de la unidad LSTM.
f: La puerta "del olvido" (*forget gate*), donde se decide la parte de memoria que será ignorada en el siguiente estado.
i: La puerta de entrada externa (*external input gate*).
o: La puerta de salida (*output gate*).

La memoria c_j se actualiza: la puerta de olvido f, controla la cantidad de la memoria anterior que se conserva y la puerta de entrada i, controla la cantidad de las actualizaciones propuestas a mantener. Finalmente, el valor de h_j (que también es la salida y_j) se determina en función del nuevo estado de la memoria c_j.
Este mecanismo de activación permite que los gradientes permanezcan altos en rangos de tiempo muy altos, por lo que evita el problema de desaparición del gradiente.

- **GRU**: Propuestas por *Kyunghyun Cho* [20], e igual que las LSTM las GRU (***Gated Recurrent Units***) se basan en un mecanismo de activación, pero con menos puertas, GRU sólo posee dos puertas, y sin dividir la memoria. Esto hace que GRU sea más simple que LSTM:

$$
\begin{aligned}
s_j = R_{\text{GRU}}(s_{j-1}, x_j) &= (1 - z) \odot s_{j-1} + z \odot \tilde{s_j} \\
z &= \sigma(x_j W^{xz} + s_{j-1} W^{sz}) \\
r &= \sigma(x_j W^{xr} + s_{j-1} W^{sr}) \\
\tilde{s_j} &= \tanh(x_j W^{xs} + (r \odot s_{j-1}) W^{sg})
\end{aligned}
$$

$$
y_j = O_{\text{GRU}}(s_j) = s_j
$$

$$
s_j, \tilde{s_j} \in \mathbb{R}^{d_s}, \; x_i \in \mathbb{R}^{d_x}, \; z, r \in \mathbb{R}^{d_s}, \; W^{xo} \in \mathbb{R}^{d_x \times d_s}, \; W^{so} \in \mathbb{R}^{d_s \times d_s}.
$$

Siendo:

z: La puerta de actualización (*update gate*).
r: La puerta de reajuste (*reset gate*).
h: La puerta de salida y a la vez el estado.

En esta arquitectura, la puerta de actualización, *z*, indica cuánto contenido de las celdas anteriores hay que mantener y la puerta de reajuste, *r*, indica cómo incorporar la nueva entrada al contenido anterior de la celda. La siguiente figura (14) muestra el esquema de la arquitectura GRU.

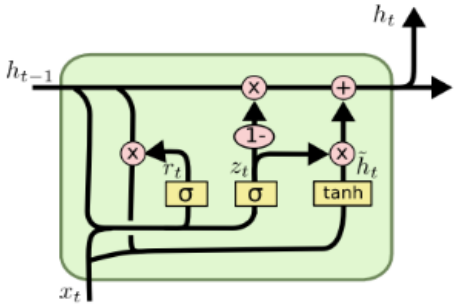

Figura 8. Esquema de GRU.

4.2 Representación de datos

Uno de los grandes retos a los que se enfrenta cuando se analizan lenguajes naturales es la forma de representar los datos, ya que muchos algoritmos de aprendizaje automático no permiten trabajar con datos categóricos. En este apartado se van a explicar los dos métodos más comunes para ese fin y el método que se ha usado para el desarrollo de este TFG. Las dos representaciones del lenguaje natural más usadas son la **codificación one-hot** (*one-hot encoding*) y la **codificación de vectores de incrustación profunda** de palabras (*dense embedding vector*).

La primera forma de representación, *embedding vector*, consiste en incrustar cada una de las características principales de un vocabulario (determinantes, sustantivos...) en un vector de d dimensiones. Para ello se entrena una red neuronal prealimentada de la siguiente manera:

1º: Se extraen las principales características lingüísticas de un vocabulario, *f1*,...,*fk* .
2º: Para cada una de esas características se devuelve un vector, *v.fi.*
3ª: Se combinan los vectores con alguna operación (concatenación, suma, o una combinación de ambos), formando un vector de entrada *x*.
4ª: Se introduce el vector *x* como entrada a un clasificador no-lineal.

La segunda representación, *one-hot*, utilizada como método de codificación de datos en este TFG, consiste en un método en el que las palabras son representadas como una secuencia de códigos numéricos que identifican cada carácter de la palabra según una determinada tabla de conversión. Esta técnica representa las palabras como vectores binarios, primero necesita que las palabras estén mapeados a valores enteros y luego representa cada variable entera (o letra) a un vector donde todos los valores están a cero menos la posición que corresponde a la letra en cuestión. De esta manera se asigna un identificador a cada a cada uno de los elementos del vocabulario. La tabla de conversión se obtiene enumerando los caracteres que se observan en los datos. Esta tabla de conversión reservará la posición 0 para representar un carácter especial de relleno (PAD). En la Figura 9 puede verse un ejemplo de codificación junto con su tabla de conversión.

Debido a que las redes neuronales necesitan tanto entradas como salidas de dimensión fija, las secuencias codificadas se redimensionan para que su longitud coincida con una determinada longitud máxima y las posiciones de esas secuencias que no representan

caracteres se rellenan con el carácter especial de relleno (PAD). En la Figura 9 puede verse como la secuencia final tiene longitud cinco y sus dos últimas posiciones representan el carácter especial.

PAD	0
‘a’	1
‘b’	2
‘o’	3
‘p’	4
‘s’	5

“paso”

→ [4, 1, 5, 3]
→ [4, 1, 5, 3, 0, 0]
→ [[0,0,0,0,1,0],[0,1,0,0,0,0],[0,0,0,0,0,1],[0,0,0,1,0,0],[1,0,0,0,0,0],[1,0,0,0,0,0]]

Figura 9. Ejemplo de codificación one-hot.

4.3 Modelos Seq2Seq

Seq2Seq, (secuencia a secuencia), es una familia de modelos basados en redes neuronales que transforman secuencias en secuencias. Estos modelos han demostrado su utilidad en traducción automática donde el orden de palabras y su contexto es importante para traducir una oración. Los modelos Seq2Seq, también llamados modelos codificador-decodificador, están formados por dos elementos básicos, el primero es el codificador (*encoder*) y el segundo es el decodificador (*decoder*). El codificador mapea la secuencia de entrada (también llamada *context vector*) a un espacio dimensional, representado por un vector mediante una función de codificación, y se lo pasa al decodificador que lo convierte en una secuencia. Los codificadores y decodificadores más simples usan una celda LSTM (*Long Short Term Memory*), explicada en el primer apartado de esta sección. En las Figuras 10 y 11 se muestran ejemplos de la arquitectura *codificador-decodificador*.

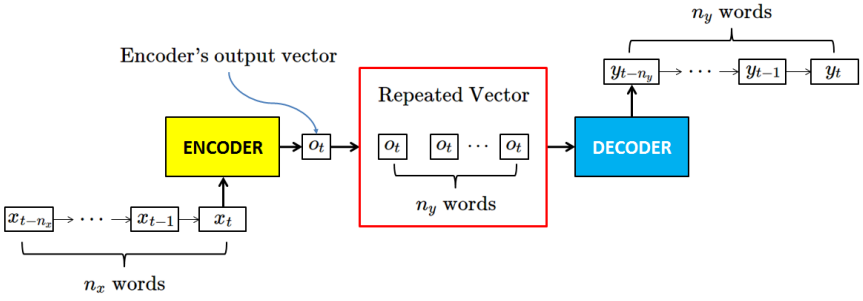

Figura 10. Ejemplo de la arquitectura codificador-decodificador.

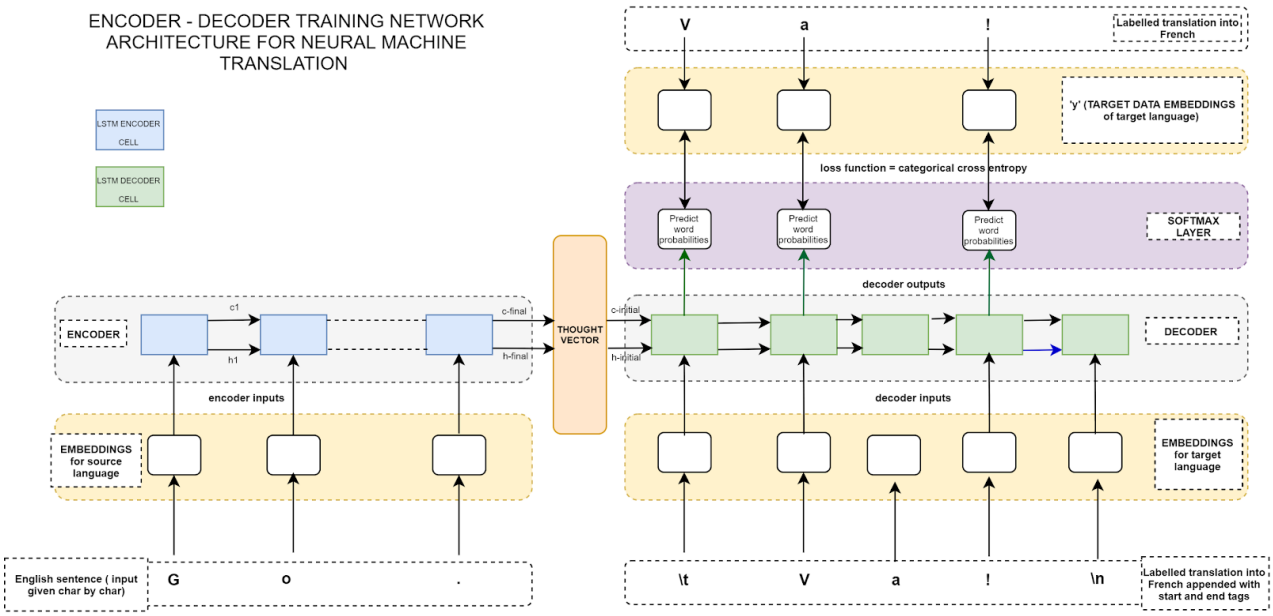

Figura 11. Ejemplo de la arquitectura codificador-decodificador.

El problema que presenta esta arquitectura es el bajo rendimiento en secuencias de entrada o salida largas, esto se debe a que esta arquitectura obliga a todas las secuencias de entrada a estar codificadas en un vector de longitud única. Para remediar este problema se puede utilizar un **mecanismo de atención**, el cual permite al decodificador centrarse en una parte de contexto que genera el codificador, en vez de usar el contexto entero de toda la oración como, para ello el codificador provee más contexto al decodificador. En la Figura 12 se muestra un ejemplo de la arquitectura con un mecanismo de atención.

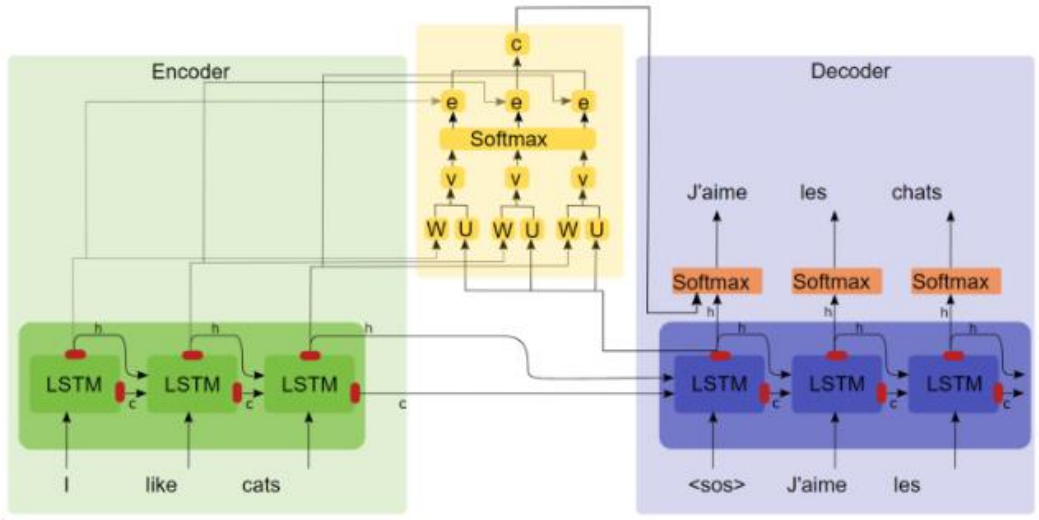

Figura 12. Ejemplo de la arquitectura codificador-decodificador con un mecanismo de atención.

4.4 Parámetros e hiperparámetros de los modelos.

El rendimiento y la capacidad computacional de los algoritmos basados en redes neuronales están relacionados en gran parte con la correcta selección de los parámetros para el entrenamiento de la red.

Para entender mejor el funcionamiento de las redes neuronales y del modelo Seq2seq, se deben definir los parámetros e hiperparámetros que éste utiliza. Los parámetros se usan para aproximar la función que se esté buscando, por ejemplo los pesos, y los hiperparámetros o parámetros de entrenamiento se usan con el fin de parametrizar la instanciación del modelo, es decir, se escogen con el fin de entrenar un modelo.

Los hiperparámetros más comunes usados para entrenamiento de los algoritmos basados en redes neuronales son los siguientes:

- *Epochs* (épocas): Consiste en el número de iteraciones que se repite el algoritmo sobre los datos de entrenamiento. Es definido, generalmente, como "una pasada por el conjunto de datos" [13]. Este hiperparámetro, es usado para separar el entrenamiento en distintas fases lo que nos permite llevar un registro de los logs.
- *Sample* (muestra): Cada elemento del dataset, cada fila que contiene la base de datos.
- *Batch size* (tamaño de lote): Es un subconjunto de samples o elementos a trabajar antes de actualizar los parámetros del modelo interno. Cada batch entrena la red en un orden sucesivo, teniendo en cuenta los pesos actualizados provenientes del lote anterior. Por ejemplo si tenemos un conjunto de datos de 100 muestras y un tamaño de lote de 20, el algoritmo entrena la red con las 20 primeras muestras del conjunto y actualiza los parámetros internos de la red. Después toma las 20 segundas muestras y vuelve a entrenar la red y así sucesivamente. Este hiperparámetro es muy útil, ya que nos permite evaluar de forma paralela todos los elementos dentro del batch. Como podemos deducir, cuanto más grande es el batch, mejores resultado da, pero también hace que el algoritmo tarde más tiempo entrenando, por ello, es recomendable encontrar el tamaño óptimo del batch que de un resultado aceptable, en un tiempo razonable.
- *Dropout*: Es uno de los hiperparámetros más importantes. Es una técnica de regularización simple usada en algoritmos de redes neuronales, donde se busca evitar el sobreajuste de la red y en la que unas neuronas seleccionadas al azar son ignoradas durante el entrenamiento. El sobreajuste o *overfitting*, consiste en sobreentrenar el algoritmo, lo que genera que dicho algoritmo sólo se ajustará a aprender los casos particulares que se le ha enseñado y no es capaz de reconocer patrones en nuevos datos de entrada.
- Número de neuronas en la capa oculta: Este parámetro define el número de neuronas que habrá en cada una de las capas ocultas que construyen la red. Encontrar el número óptimo de neuronas es bastante importante, ya que si éste es demasiado pequeño la red puede no entrenar de forma satisfactoria, y si es bastante grande, el tiempo de computación del algoritmo puede ser bastante alto.
- Optimizador: *Adam*, *SGD*, *RMSprop*, *Adagrad*, *Adadelta* y varios más son ejemplos de optimizadores, lo cuales se utilizan para hacer más preciso el modelo que estamos utilizando para entrenamiento el red.
- *Learning rate*: (tasa de aprendizaje). Es una función decreciente en el tiempo que indica la velocidad a la que estamos ajustando lo pesos de la red, tiene un rango entre 0 y 1, y afecta a la velocidad a la que el algoritmo alcanza las ponderaciones óptimas.

5 Experimentos y resultados

Con el fin de conseguir un algoritmo que genere una solución aceptable al problema planteado, se ha experimentado con varias herramientas e hiperparámetros ya explicados en apartados anteriores.

Como primera opción, se usó la herramienta ***openNMT***, la cual consiste en un sistema de código abierto de la traducción automática neuronal y aprendizaje de secuencias neuronales [22]. OpenNMT no ha generado resultados satisfactorios, por lo que ha sido descartado como herramienta de investigación para el problema planteado.

La segunda opción que se ha contemplado ha sido la herramienta ***Keras***, que consiste en una biblioteca de redes neuronales, también de código abierto, escrita en lenguaje Python [23]. Esta segunda opción ha dado mejores resultados que el *openNMT*. En los apartados siguientes se van a exponer y explicar los experimentos realizados y los resultados obtenidos con cada una de las bases de datos, explicadas en el apartado *Materiales*.

Por otra parte, para la búsqueda de los parámetros que optimizan los resultados se ha utilizado el método Grid Search (búsqueda en rejilla), el cual consiste en encontrar la optimización de los hiperparámetros, es decir, la mejor combinación de hiperparámetros para un modelo dado, en este caso LSTM (*Long Short Term Memory*), y un conjunto de datos de prueba. Los hiperparámetros que se han tenido en cuenta para la realización de pruebas son el número de neuronas en la oculta de la red, *epochs*, *batch size* y *dropout*. En este proyecto se ha usado una adaptación de este método, ya que en general es usado con diferentes modelos, sin embargo en este caso se ha mantenido un único modelo, LSTM, y se han variado los hiperparámetros.

Para este fin se han desarrollado varios script, que mediante el método Grid Search, han ido entrenando la red, ejecutando el modelo y generando los porcentajes de aciertos y las gráficas de aprendizaje. En la Figura 13 se muestra un ejemplo.

```
DATABASE = DysList_resource.data echoDatabase.data   German_Annotation_V027_final.data
BATCHSIZE = 8 16
DROPOUT = 0.3
HIDDENDIM = 200 300 400 500
all:
        for batch in $(BATCHSIZE); do \
            for data in $(DATABASE); do \
                for hidden in $(HIDDENDIM); do \
                    for dropout in $(DROPOUT); do \
                        for epoch in 100; do \
                            python3.6 test3-german.py $$data $$epoch $$batch $$hidden $$dropout $
                        done \
                    done \
                done \
            done \
        done

clean:
    rm -f test3-*.hdf5 *~
```

Figura 13. Ejemplo de script de búsqueda en rejilla.

Todos los experimentos con las distintas bases de datos se han realizado usando dos modelos seq2seq y redes LSTM, pero cambiando tanto el número de capas ocultas que

forman la red, como el uso o no de un mecanismo de atención, el cual se ha definido en apartados anteriores desarrollados a lo largo de este TFG.

- El primer modelo que se ha usado para entrenar la red posee tres capas ocultas y no dispone de mecanismo de atención. En la Figura 14 se muestra un ejemplo de dicho modelo.
- El segundo modelo dispone de dos capas ocultas y un mecanismo de atención.

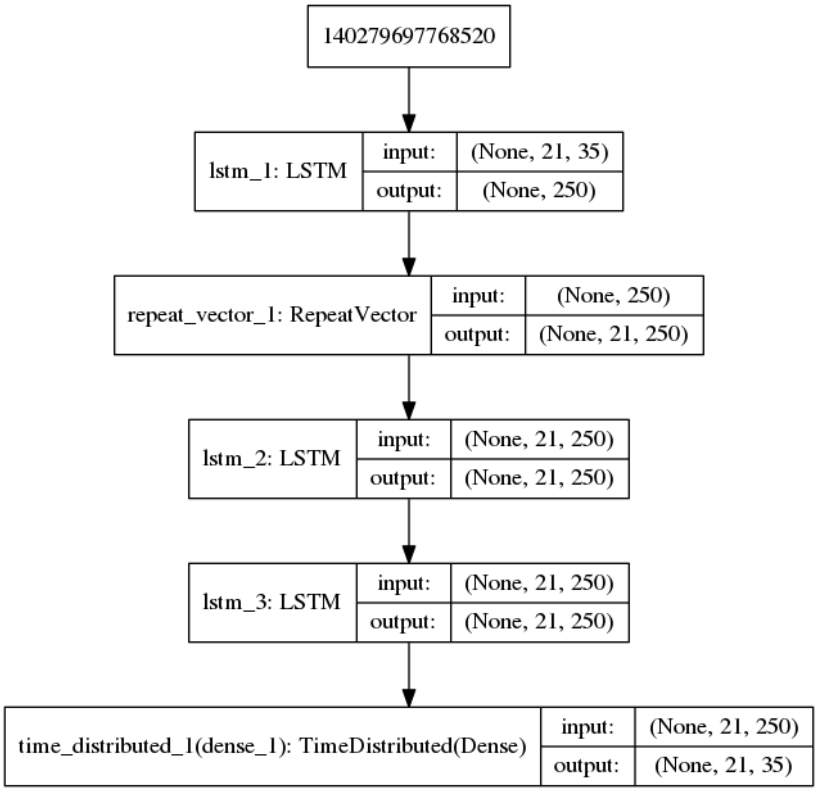

Figura 14. Modelo 1.

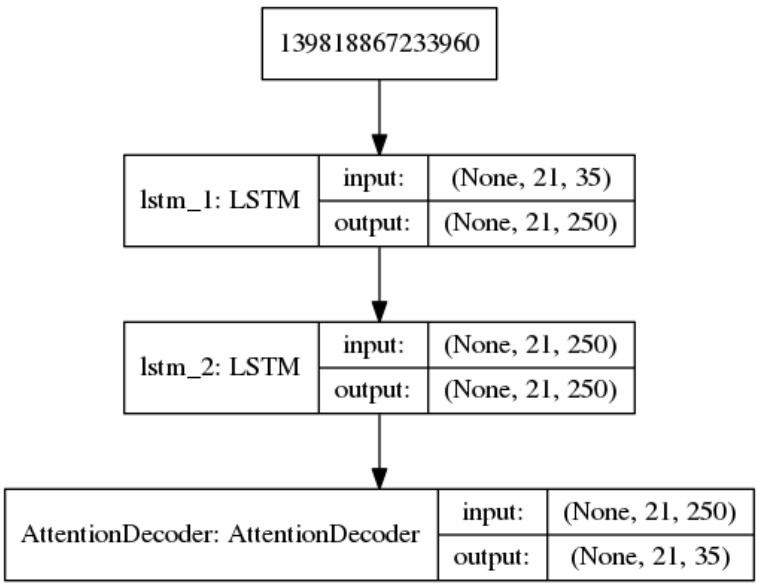

Figura 15. Modelo2.

En los siguientes apartados se van a exponer los resultados que han generado los diferentes conjuntos de datos combinando los hiperparámetros.

5.1 ECHO

Con el fin de obtener una visión general de los conjuntos de datos y medir la cota superior que los algoritmos son capaces de alcanzar, se han creado dos bases de datos, a partir de las dos explicadas anteriormente. La primera contiene las mismas secuencias de palabras que la base de datos en español, y la segunda, las mismas de la base de datos del alemán. Con este experimento se ha conseguido medir la capacidad computacional de los distintos modelos Seq2Seq para reproducir los datos de entrada en la salida, de manera que la precisión de estos modelos debería servir para estimar la cota máxima o el techo de los resultados que se obtendrían para otros problemas de secuencias más complejos. Es decir, con este experimento se pretende definir el alcance del algoritmo Seq2Seq que se ha utilizado, sus parámetros e hiperparámetros. En los siguientes apartados se procederá a explicar por separado cada uno de los dos experimentos.

5.1.1 Echo Español

En el Anexo B se muestra un extracto de la base de datos usada para este caso. Después de entrenar el algoritmo usando el segundo modelo, explicado la primera sección de este capítulo mismo, y diferentes hiperparámetros, se ha llegado un porcentaje de acierto máximo del 99 %, esto no significa otra cosa que la cota superior que dicho algoritmo puede alcanzar aplicado sobre otros conjuntos de datos. En las siguientes figuras se muestran los resultados con varios experimentos. En las cuales, las primeras filas son el batch size (color amarillo), la segunda son el número de neuronas en la capa oculta (color verde), la tercera el dropout (color azul) y la última son el porcentaje de aciertos (color naranja) con cada uno de esos parámetros.

Figura 16.1. Resultados con base de datos Echo Español.

Figura 16.2. Resultados con base de datos Echo Español.

Como se puede observar en las diferentes figuras anteriores, el máximo porcentaje de aciertos que la el algoritmo fue capaz de detectar ha sido del 99 %. Lo que indica que el algoritmo ha sido capaz de encontrar un modelo que detecta un 1 % de palabras correctas como erróneas. Haciendo un análisis más cualitativo de los datos sobre uno de los ficheros se han encontrado los siguientes tipos de error.

• Cambiar “u” por “g”	Cambiar “ó” por “i”	Cambiar “d” por “a”
• Cambiar “u” por “t”	Cambiar “ó” por “ú”	Cambiar “d” por “o”
• Cambiar “u” por “i”	Cambiar “ó” por “t”	Cambiar “d” por “e”
• Eliminar “d”	Eliminar “r”	Eliminar “m”

Observando los tipos de error se ha interpretado que el algoritmo en algunos de los casos no ha sido capaz de generalizar de forma totalmente correcta. En la Figura 16 se muestra una de las gráficas, en la que se representa los porcentajes de aciertos en entrenamiento validación frente al número de épocas, donde se ha entrenado con un tamaño de lote de 2 y 250 neuronas en la capa oculta. En la cual se observa que tanto el porcentaje de entrenamiento como de validación supera el 99 %. Además las dos curvas, entrenamiento y validación, son bastante alineadas, lo indica que no hay *overfitting* ni *underfitting.*
Por otra parte se ha observado que usando un número de neuronas más alto, a partir de 500, el algoritmo genera resultados insatisfechos. De esto se muestra un ejemplo en la Figura 17 y es debido a que la red subajusta (*underfitting*), no es capaz de generalizar el conocimiento que adquiere.

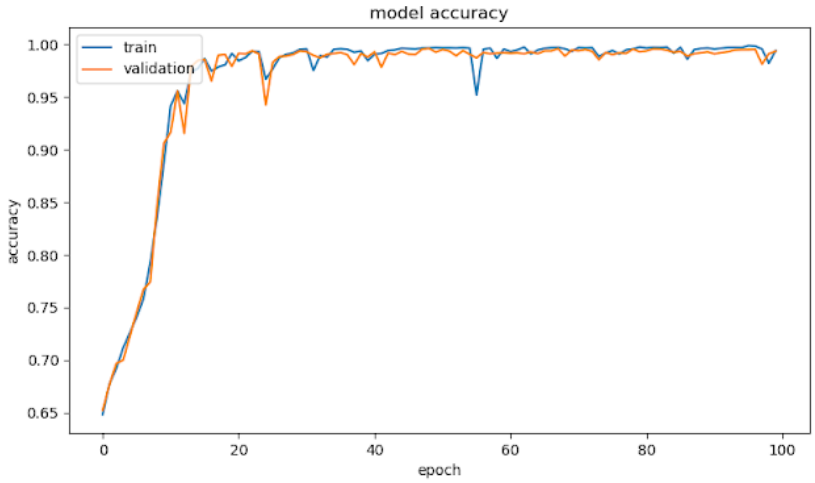

Figura 17. Porcentaje de aciertos en entrenamiento y validación. Tamaño de lote 8 y 250 neuronas.

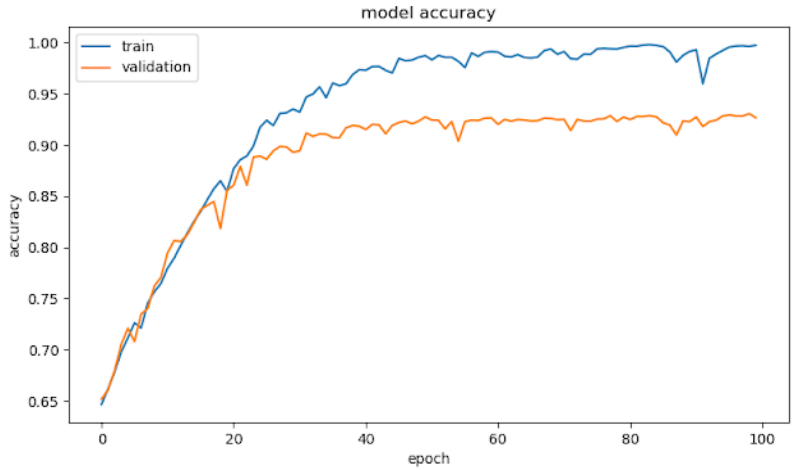

Figura 18. Porcentaje de aciertos en entrenamiento y validación. Tamaño de lote 8 y 500 neuronas.

5.1.1 Echo Alemán

En el Anexo C se muestra la base de datos usada para este caso. Al igual que en el caso anterior, se realizaron varias pruebas con el segundo modelo. En las siguientes figuras se muestran algunos ejemplos de las curvas de aprendizaje que ha generado el modelo con esta base de datos.

Analizando el conjunto total de pruebas se ha observado que variando los diferentes hiperparámetros, en la mayoría de los casos el porcentaje de aciertos ha superado el 99 %. Lo que significa que el modelo podido reproducir satisfactoriamente los datos de entrada en la salida.

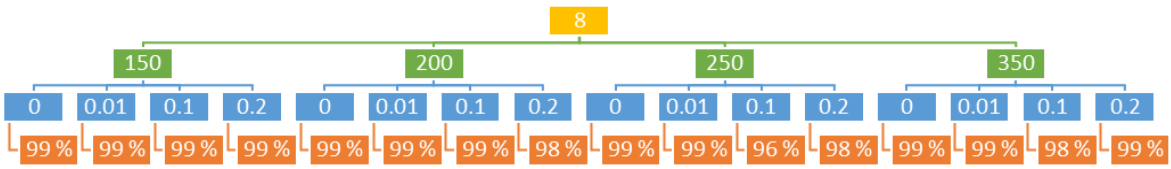

Figura 19.1. Resultados con base de datos Echo Alemán.

Figura 19.2. Resultados con base de datos Echo Alemán.

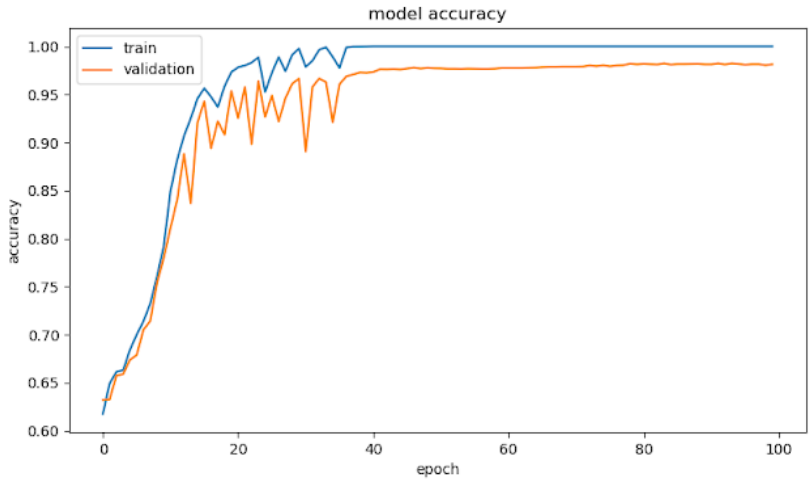

Figura 20. Porcentaje de aciertos. Echo alemán. Tamaño de lote 8 y 150 neuronas.

5.2 DYSLEXIA-ES

Después de realizar los experimentos con la base de datos, echo, comentada en el apartado anterior, interpretar los resultados y ajustar los valores de los parámetros hasta conseguir unos resultados aceptables, pasamos a probar los modelos con la base de datos de dislexia en español [6], explicada detalladamente en el apartado de *Materiales*.

En este caso el máximo porcentaje de aciertos que se ha logrado ronda el 90%. Analizando uno de los ficheros de salida que ha generado el algoritmo (En el *Anexo A* se añade un ejemplo de estos ficheros), con el fin de encontrar patrones de error. Se ha podido contemplar que la mayor parte de los errores se han generado debido a las repeticiones de palabras, es decir, las mismas palabras con diferentes errores se corrigen de forma errónea la mayoría de las veces que aparecen, como cambiar la letra "b" por la letra "v", añadir u omitir espacios, un ejemplo de este caso se muestra en la Tabla 7. El resto de errores son bastante incoherentes seguramente debido a la escasez de ejemplos con lo que la red ha sido entrenada. En las siguientes figuras se muestra algunos resultados y gráficas que muestran las curvas de aprendizaje.

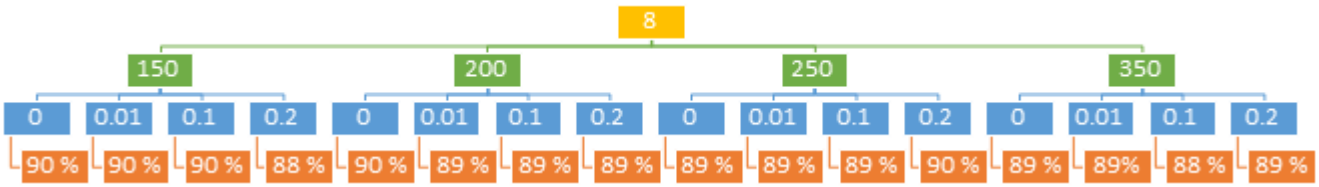

Figura 21.1: Resultados con base de datos Español.

Figura 21.2: Resultados con base de datos Español.

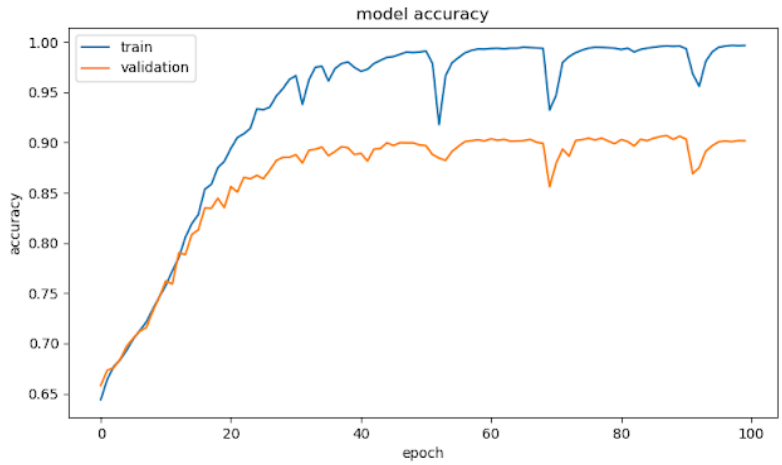

Figura 22. Porcentaje de aciertos. Español. Tamaño lote 8 y 150 neuronas.

queraba	que~a~ba	(quedara)
seguió	se~vió	(siguió)
bos~los	voslos	(vos~los)

Tabla 7. Ejemplo de errores español.

Por otra parte se ha observado que usando un *dropout* grande (10 o 20 %) la curva de aprendizaje crece más lentamente, sin embargo las dos curvas crecen más juntamente, ya que un *dropout* grande reduce el *overfitting*, lo que hace que las neuronas cercanas aprendan patrones que se relacionan y estas relaciones pueden llegar a formar un patrón muy específico con los datos de entrenamiento. Esto último genera que la red aprenda unos casos específicos y nos es capaz de producir generalizaciones. En la Figura 20, se muestra un ejemplo de este caso.

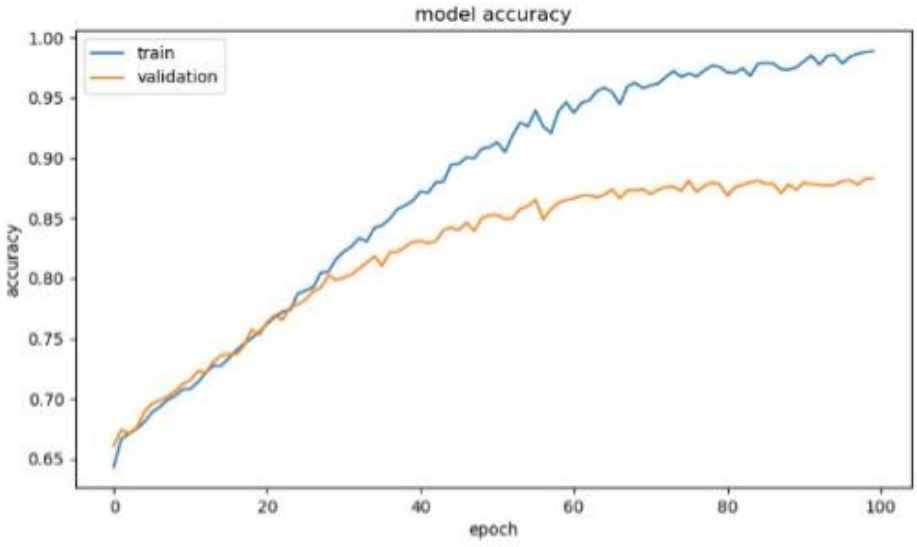

Figura 23. Porcentaje de aciertos. Español. Tamaño lote 8, 150 neuronas y dropout del 20%

5.3 DYSLEXIA-DE

Con la base de datos en palabras en alemán [5], se ha empleado el mismo proceso que con el resto de conjuntos de datos arriba explicados.
Durante el entrenamiento esta base de datos ha producido mejores resultados que la base de datos de español, un 3 % más de aciertos, el 93 % es el máximo porcentaje que se ha podido alcanzar. En las figuras siguientes se mostrarán los resultados más relevantes. De los errores que se han podido observar en este conjunto se ha podido deducir que son debidos en gran parte a los pocos ejemplos con los que ha entrenado la red y como consecuencia no ha podido establecer patrones generales para la corrección de los diferentes errores.

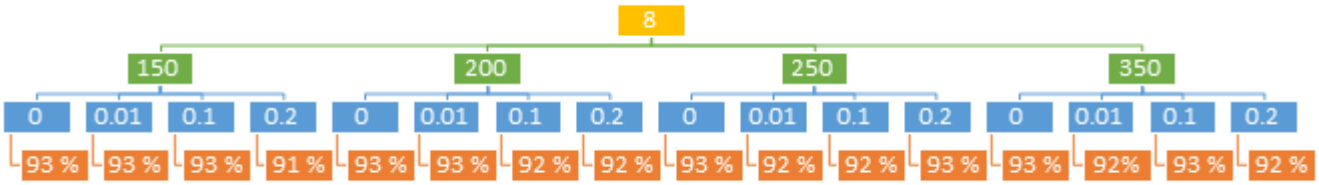

Figura 24.1: Resultados con base de datos Alemán.

Figura 24.2: Resultados con base de datos Alemán.

En la siguiente gráfica se muestra la curva de aprendizaje de uno de los ejemplos con un batch size de 8 y 200 neuronas en la capa oculta.

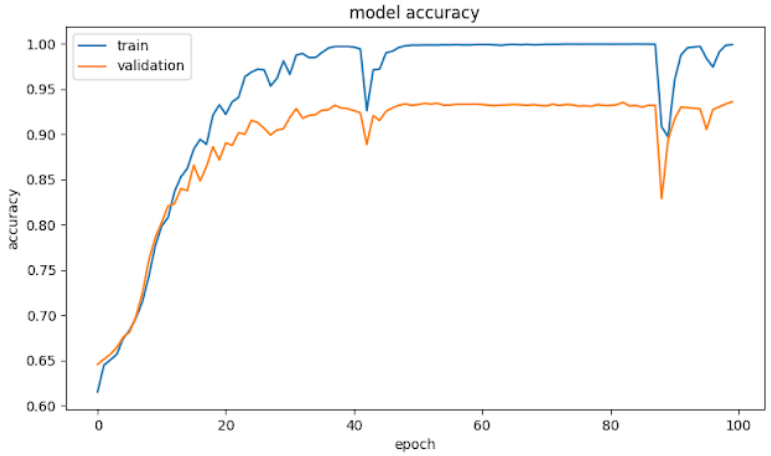

Figura 25. Porcentaje de aciertos. Español. Tamaño lote 8 y 150 neuronas.

Haciendo un análisis cualitativo de los resultados, con la dificultad de no tener competencias en el idioma, se ha podido concluir que gran parte de los errores producidos por el algoritmo son debidos, al igual que las demás bases de datos, a los pocos datos con los que ha entrenado la red. Esto tiene como consecuencia que el algoritmo no es capaz de aprender y generalizar todos los errores que poseen los conjuntos de datos.

6 Conclusiones y trabajo futuro

6.1 Conclusiones

Tras realizar este proyecto, he podido profundizar mi aprendizaje sobre los algoritmos basados en redes neuronales, el procesamiento de lenguaje natural y la dislexia, como elementos principales que forman este TFG. También puedo concluir que se ha alcanzado, en gran medida el objetivo propuesto en este proyecto.
El cual consistía en encontrar un algoritmo capaz de detectar y corregir errores disléxicos.
A pesar de que el modelo no ha sido capaz de alcanzar el 100 % de aciertos, sí ha alcanzado un 90 % o más en la gran parte de los casos. Esto como he podido deducir es debido a falta de datos que se han usado para el desarrollo de este TFG.

En cuanto a los objetivos que se han podido alcanzar han sido:

- Un modelo que alcanza un 99 % en la base de datos echo de español.
- Un modelo que alcanza un 99 % en la base de datos echo de alemán.
- Un modelo que alcanza un 90 % en la de datos de español.
- Un modelo que alcanza un 93 % en la de datos de alemán.

Por otra parte y desde el punto de vista de los conocimientos que he podido adquirir:

- Lenguaje Python.
- Uso de la herramienta *Keras*.
- El análisis y la interpretación de los resultados generados a partir de los algoritmos de redes neuronales que se han empleado.
- Conocimiento sobre las redes neuronales artificiales y sus diferencias.

6.2 Trabajo futuro

Como he mencionado los capítulos anteriores para el desarrollo de este TFG, me he basado en las redes neuronales recurrentes, porque han tenido gran éxito en el procesamiento del lenguaje natural, sin embargo para futuros trabajos se puede probar con otro tipo de redes neuronales como pueden ser las convolucionales. Aunque este tipo de redes construye más patrones espaciales, frente a los secuenciales que son los que tienen en cuenta las redes neuronales recurrentes. Otra forma de atacar este problema es probando con otro tipo de tecnología, que no sea *Keras.*
Por otro lado, se podría usar la búsqueda aleatoria de hiperparámetros, sustituyendo la búsqueda en rejilla (*Grid Search*), que es la que he empleado en este TFG, ya que según *Yoshua Bengio* [31], las ventajas prácticas de la búsqueda aleatoria frente a la búsqueda en rejilla son simplicidad conceptual, facilidad de implementación, paralelismo trivial.

Referencias

[1] https://integratek.es/que-es-la-dislexia/
[2] Interagency Commission on Learning Disabilities. Learning Disabilities: A Report to the U.S. Congress. Government Printing Office, Washington DC, 1987.
[3] J. E. Jiménez, R. Guzmán, C. Rodríguez & C. Artiles. Prevalencia de las dificultades específicas de aprendizaje: La dislexia en español. Anales de Psicología, 25(1):78–85, 2009.
[4] M. S. Carrillo, J. Alegría, P. Miranda & S. Pérez. Evaluación de la dislexia en la escuela primaria: Prevalencia en español. Escritos de Psicología, 4(2):35–44, 2011.
[5]https://docs.google.com/spreadsheets/d/1d09itNlk12XxBQOr9bsmUWZgdlxzS4U_6qwlbVn4-60/edit#gid=355627579
[6]https://docs.google.com/spreadsheets/d/1lyScHph5z3859C8GVnhQq84JF_KqsjrZyM122MTqQac/edit#gid=711366020
[7]https://www.researchgate.net/publication/265972336_DysList_An_Annotated_Resource_of_Dyslexic_Errors
[8] https://pdfs.semanticscholar.org/ccac/d06d0569c7150c23a99b570cabb3c74518c0.pdf
https://github.com/euclidjda/seq2seq-char-rnn
https://github.com/farizrahman4u/seq2seq
[9] https://www.educaciontrespuntocero.com/recursos/recursos-dislexia-alumnos/15797.html
[10] https://www.understood.org/es-mx/learning-attention-issues/child-learning-disabilities/dyslexia/dyslexia-fact-sheet
[11] International Dyslexia Association; EDA, Asociación Europea de Dislexia
[12] https://blog.insightdatascience.com/how-to-solve-90-of-nlp-problems-a-step-by-step-guide-fda605278e4e
[13] https://keras.io/getting-started/faq/#what-does-sample-batch-epoch-mean
[14] https://d-nb.info/1175204927/34
[15] Meng, H., Smith, S., Hager, K., Held, M., Liu, J., Olson, R., Pennington, B., DeFries, J., Gelernter, J., O'Reilly-Pol, T., Somlo, S., Skudlarski, P., Shaywitz, S., Shaywitz, B., Marchione, K., Wang, Y., Murugan, P., LoTurco, J., Grier, P., and Gruen, J. (2005). DCDC2 is associated with reading disability and modulates neuronal development in the brain. Proceedings of the National Academy of Sciences, 102:17053–17058, November.
[16] Computer Correction of Real-word Spelling Errors in Dyslexic Text. Ph.D. thesis, Birkbeck College, London University.
[17] Rello, L., Baeza-Yates, R., Saggion, H., and Pedler, J. (2012a). A first approach to the creation of a Spanish corpus of dyslexic texts. In LREC Workshop Natural Language Processing for Improving Textual Accessibility (NLP4ITA), pages 22–27, Istanbul, Turkey, May
[18] https://www.researchgate.net/figure/Figura-III4-Capas-de-una-Red-Neuronal-Capa-de-entrada-neuronas-que-reciben-datos-o_fig3_315762548

[19] https://stxlearning.com/2018/02/25/ejemplo-de-representaciones-distribuidas-en-procesamiento-de-lenguaje-word2vec/

[20] https://www.ibm.com/developerworks/ssa/library/cc-machine-learning-deep-learning-architectures/index.html

[21] http://www.diegocalvo.es/clasificacion-de-redes-neuronales-artificiales/
[22] http://opennmt.net/
[23] https://es.wikipedia.org/wiki/Keras
[24] https://www.aprendemachinelearning.com/que-es-overfitting-y-underfitting-y-como-solucionarlo/
[25] https://towardsdatascience.com/light-on-math-ml-attention-with-keras-dc8dbc1fad39
[26] https://onlinelibrary.wiley.com/doi/abs/10.1111/1467-8535.00331
[27] Santana, V.F., Oliveira, R., Almeida, L., Ito, M.: Firefixia: An accessibility web browser customization toolbar for people with dyslexia. In: Proc. W4A '13. Rio de Janeiro, Brazil (2013)
[28] https://blog.changedyslexia.org/wp-content/uploads/2017/03/wsrua2013.pdf
[29] https://www.disfam.org/tag/dislexia/
[30] https://vincentblog.xyz/posts/dropout-y-batch-normalization
[31] Random Search for Hyper-Parameter Optimization. James Bergstra, Yoshua Bengio. 2012.

Glosario

Rnn: Red neuronal recurrente.
LSTM: Long short term memory.
GRU: Gated Recurrent Units.
CNN: Red neuronal Convolucional.
CBOW: continuous bag of words.
PLN: procesamiento del lenguaje natural.

Anexos

A Ejemplo de fichero de salida. Español

ajugar	agugar	(a~jugar)	beneciciario	beneficiario	(beneficiario)
absorver	hacosrber	(absorber)	berengena	berenjena	(berenjena)
absorviendo	aaoosebendo	(absorbiendo)	derengena	derenjena	(berenjena)
accessiblidad	accesibilidad	(accesibilidad)	derengena	derenjena	(berenjena)
acsevilidad	accesibilidad	(accesibilidad)	berengenas	berenjena	(berenjenas)
aczecibilidad	accesibilidad	(accesibilidad)	viblioteca	bivlizacis	(biblioteca)
acetino	adjetivo	(adjetivo)	bienm	viene	(bien)
adjetibo	adjetivi	(adjetivo)	blancecina	blanciaa	(blanquecina)
admin	aabie	(admiten)	buro	buroo	(burro)
admin	aabie	(admiten)	buso	vusoo	(busco)
abverbios	adverbios	(adverbios)	cavallro	caballero	(caballero)
adbervios	adverbios	(adverbios)	calavacines	caladaiinn	(calabacines)
adbervios	adverbios	(adverbios)	caejón	sacinó	(callejón)
agraderá	agradecerá	(agradecerá)	cayejón	calaeóó	(callejón)
abua	haba	(agua)	canció	sanció	(canción)
ergue	egua	(agua)	contar	contra	(cantar)
alfavetización	al~ftabicnó	(alfabetización)	cantarímo	cantaríamos	(cantaríamos)
ilegenes	alienígena	(alienígena)	captel	suttel	(cartel)
ilegenes	alienígena	(alienígena)	catrastrófica	cartastócicnc	(catastrófica)
alviado	alivado	(aliviado)	cérlas	células	(células)
al~readedor	alrededor	(alrededor)	derca	derca	(cerca)
alrrederdor	alrededor	(alrededor)	rerca	rerca	(cerca)
antropómimo	antropamimo	(antropónimos)	zerca	cerca	(cerca)
antropómimo	antropamimo	(antropónimos)	cerreza	cerrasa	(cereza)
aho	hay	(año)	ciervatillo	cicrrballo	(cervatillo)
ajo	agoo	(año)	zincomil	hicsoimi	(cinco~mil)
anyo	hño	(año)	zincomil	hicsoimi	(cinco~mil)
anyos	añoo	(años)	quine	qui~e	(cine)
añyos	año	(años)	cisculante	sigslaeten	(circulante)
aprobechar	acrorecuar	(aprovechar)	cyrculante	curcantent	(circulante)
ají	aquí	(aquí)	classe	cllass	(clase)
arcite	arquitectura	(arquitectura)	cubo	gugvo	(cobo)
arcite	arquitectura	(arquitectura)	cojen	cogeee	(cogen)
arsestaron	arsestaro	(arrestaron)	cojió	cogii	(cogió)
ací	hací	(así)	comert	comerta	(comer)
antún	antin	(atún)	composicion	composion	(composicional)
aiuntamiento	ayuntamiento	(ayuntamiento)	composicion	composion	(composicional)
ayunamyento	ayuntamiento	(ayuntamiento)	comi	comm	(con~mi)
vajita	bajita	(bajita)	comi	comm	(con~mi)
vanco	bacio	(banco)	conesión	consióó	(conexión)

confianca	conficaa	(confianza)	ellosedebe	ellos~des	(ello~se~debe)
conjelados	conjllado	(congelados)	ellosedebe	ellos~des	(ello~se~debe)
contimigo	conmigo	(conmigo)	enbarcación	envaacacón	(embarcación)
consige	consij	(consigue)	empecado	empecao	(empezado)
cantemplór	contempló	(contempló)	energí	engrgí	(energía)
convenos	convencerlos	(convencerlos)	ensenyo	enseño	(enseño)
convenos	convencerlos	(convencerlos)	etonces	etrocces	(entonces)
ceonverración	conversación	(conversación)	em~bía	envía	(envía)
convesación	convarcación	(conversación)	embueve	envuelve	(envuelve)
creao	crealo	(creado)	erera	era	(era)
cundo	gunddd	(cuando)	hermita	herminia	(ermita)
corpo	corpoo	(cuerpo)	eacrive	eacibbe	(escribe)
curpo	gurpo	(cuerpo)	eacrive	eacibbe	(escribe)
cueeva	cueveaa	(cueva)	escrib	escrib	(escribe)
deagua	dejgaa	(de~agua)	escibrimos	escribimos	(escribimos)
dela	de~ll	(de~al)	essos	estos	(esos)
decartón	de~citón	(de~cartón)	especiale	espeesale	(especiales)
degent	de~gente	(de~gente)	explendor	expre~vero	(esplendor)
dehiervabuena	de~iibavenaa	(de~hierbabuena)	es~ta	esta	(esta)
dehiervabuena	de~iibavenaa	(de~hierbabuena)	extrés	estroa	(estrés)
demar	de~mar	(de~mar)	extricta	estricaa	(estricta)
depollo	de~polol	(de~pollo)	extrictamente	extricaanente	(estrictamente)
devilidat	debilidad	(debilidad)	exepecpto	excepto	(excepto)
dejir	dejii	(decir)	exepecpto	excepto	(excepto)
degar	de~ar	(dejar)	experto	experto	(excepto)
determaatología	dermatología	(dermatología)	experto	experto	(excepto)
determaatología	dermatología	(dermatología)	escitación	escrición	(excitación)
desalloro	desarrollo	(desarrollo)	esplice	esplice	(explica)
desenancias	desinencias	(desinencias)	esplice	esplice	(explica)
desovediente	descrbiente	(desobediente)	extranción	extensión	(extensión)
divugo	dibujo	(dibujo)	extranción	extensión	(extensión)
divujo	dibujo	(dibujo)	extrangero	extenngero	(extranjero)
dixo	dicío	(dicho)	felicida	felicidad	(felicidad)
diziembre	diceliprr	(diciembre)	feliridad	felicidad	(felicidad)
ditado	dittddo	(dictado)	filántropoco	filántropo	(filántropo)
diffícil	diffcil	(difícil)	freguencias	freguencias	(frecuencias)
digo	dijoo	(dijo)	frequentias	frecuentias	(frecuencias)
dirigir~se	dirfigies	(dirigirse)	frequentias	frecuentias	(frecuencias)
discurción	discusión	(discusión)	frición	frición	(fricción)
disutiendoyal	discutiendo~y~al	(discutiendo~y~al)	fución	fucinó	(fusión)
dotor	dottro	(doctor)	fututuro	futuro	(futuro)
bonde	voden	(donde)	galaccia	galaxia	(galaxia)
equalizador	ecualizaro	(ecualizador)	carvarzos	carbanzos	(garbanzos)
egemplo	ejemplo	(ejemplo)	gravanzos	garbanzos	(garbanzos)

enial	jenial	(genial)	imganes	iiágnnes	(imágenes)
guente	jue~te	(gente)	incosistencia	incosstencaa	(inconsistencia)
gimansio	gimnasio	(gimnasio)	imdica	imdicc	(indica)
gimansio	gimnasio	(gimnasio)	intostrial	industrial	(industrial)
girava	giraba	(giraba)	imovilizó	imoiiiza	(inmovilizó)
jiraba	jiraba	(giraba)	inoblidable	inolvidable	(inolvidable)
guardavan	gurdabín	(guardaban)	interiorios	interior	(interior)
guerrerra	guerrs0a	(guerra)	iterios	interior	(interior)
guerrerra	guerrs0a	(guerra)	interogarle	interrogar	(interrogarle)
gusan~no	gusano	(gusano)	introcción	introducción	(introducción)
cues	gusta	(gusta)	inbaririables	invariables	(invariables)
cues	gusta	(gusta)	inbaririables	invariables	(invariables)
a	el	(ha)	irraelies	israelíes	(israelíes)
aver	haber	(haber)	jomón	jomin	(jamón)
abia	habia	(había)	gersei	geraco	(jersey)
abia	habia	(había)	gersei	geraco	(jersey)
havían	había	(habían)	góvenes	gágena	(jóvenes)
avitación	aabitación	(habitación)	juebes	juevos	(jueves)
avitación	aabitación	(habitación)	juves	juevo	(jueves)
havitacion	habitación	(habitación)	guga	jugar	(jugar)
ablar	hablar	(hablar)	gusta	justa	(justa)
abra	hablar	(hablar)	lafabrica	la~fábrica	(la~fábrica)
ace	hacer	(hace)	lavajillas	lavavajillas	(lavavajillas)
are	hrre	(hace)	lentegas	lentgaa	(lentejas)
are	hrre	(hace)	livro	libroo	(libro)
acenos	accemos	(hacemos)	llama	y~an	(llaman)
acer	hacer	(hacer)	llegavan	lleganan	(llegaban)
azer	hacer	(hacer)	luses	luses	(luces)
aré	hará	(hará)	mandavan	madaable	(mandaban)
arán	haráá	(harán)	manorita	manonta	(manita)
ai	hay	(hay)	manorita	manonta	(manita)
alla	aya	(haya)	mastes	mattet	(martes)
er~mano	hermano	(hermano)	maqulino	macunioo	(masculino)
iziste	hicistu	(hiciste)	maqulino	macunioo	(masculino)
iziste	hicistu	(hiciste)	mallonesa	mañontas	(mayonesa)
ombre	oocre	(hombre)	mallores	mañor	(mayores)
honrrado	honrado	(honrado)	mecuestacantar	me~cuesta~cantar	(me~cuesta~cantar)
huso	husso	(hueso)	mesé	me~ha	(me~sé)
cuebos	cugvos	(huevos)	medias	mediados	(mediados)
guabes	huevos	(huevos)	mentaita	mentata	(mentita)
ivan	iban	(iban)	mesesdel	me~rel0e	(meses~del)
ivan	iban	(iban)	mes~ter	me~tear	(mester)
ilistnador	ilustrador	(ilustrador)	mejicana	me~icisa	(mexicana)
imganes	iiágnnes	(imágenes)	mejicano	me~icino	(mexicano)

momemto	momentt	(momento)	persuación	persacóón	(persuasión)
mutchas	muchas	(muchas)	porla	por~la	(por~la)
mugo	mugho	(mucho)	porlatarde	por~la~tarde	(por~la~tarde)
mungo	mucho	(mucho)	porzentages	porcentajes	(porcentajes)
mungo	mucho	(mundo)	porpue	prrpee	(porque)
natulal	naturale	(natural)	posiblidades	posilidases	(posibilidades)
nezesita	necesita	(necesita)	prosencia	procenta	(presencia)
necestiva	necesitaba	(necesitaba)	pesete	persete	(presente)
necizitaba	necesitaba	(necesitaba)	pesete	persete	(presente)
nezesitaba	necesitaba	(necesitaba)	presnte	presttte	(presente)
nosabía	nosaba	(no~sabía)	porceso	proceso	(proceso)
noce	nocho	(noche)	proseso	proceso	(proceso)
notge	noche	(noche)	prozeso	proceso	(proceso)
noxe	noche	(noche)	troceso	troceso	(proceso)
nomal	nomaa	(normal)	prororrumpir	prorrumpir	(prorrumpir)
no~ovembre	noviembre	(noviembre)	protectopas	protectares	(protectoras)
nuve	nubve	(nube)	pueso	pueso	(puso)
osea	hosar	(o~sea)	gue	jue	(que)
obserbó	voserbó	(observó)	quenova	que~no~va	(que~no~va)
observór	voseróóó	(observó)	queraba	que~a~ba	(quedara)
octuber	octubre	(octubre)	queraba	que~a~ba	(quedara)
octuvre	octubre	(octubre)	quegaba	que~avaaa	(quejaba)
officina	hifffina	(oficina)	querso	que~to	(queso)
otono	ootno	(otoño)	quizo	quiso	(quiso)
pa~rde	padre	(padre)	requerda	reque~ee	(recuerda)
pájinas	páinas	(páginas)	resutlados	resultados	(resultados)
pasisague	paisaje	(paisaje)	rosafarida	rosita	(rosita)
pala	palabra	(palabra)	jabado	sábado	(sábado)
pantolones	pantolose	(pantalones)	save	sabe	(sabe)
porecía	peqfíca	(parecía)	sabias	sabilas	(savias)
parq	parque	(parque)	senfado	se~enfado	(se~enfado)
paritcipnates	participantes	(participantes)	senfado	se~enfado	(se~enfado)
paritcipnates	participantes	(participantes)	sa	se~ha	(se~ha)
paritcipnates	participantes	(participantes)	sa	se~ha	(se~ha)
patir	pairi	(partir)	senundo	senenendo	(segundo)
passado	passa0o	(pasado)	señyales	señalss	(señales)
passado	passa0o	(pasado)	cicilosamente	sigilosamente	(sigilosamente)
paiaso	paiaso	(payaso)	ligisolamente	sigilosamente	(sigilosamente)
periglo	peligro	(peligro)	ligisolamente	sigilosamente	(sigilosamente)
pemsaba	persaba	(pensaba)	sigilosarmente	sigilosamente	(sigilosamente)
paceño	pequeño	(pequeño)	signifcancia	significacióa	(significancia)
pequenyo	pequeñoto	(pequeño)	signfiticativo	significativo	(significativo)
percepto	perfecto	(perfecto)	signimicatibos	significativos	(significativos)
porsonas	por~enas	(personas)	seguió	se~vió	(siguió)

B Extracto de la base de datos echo español.

trimestre	trimestre	
habitación	habitación	
canción	canción	
agua	agua	
había	había	
haber	haber	
envuelve	envuelve	
pequeño	pequeño	
alienígena	alienígena	
adverbios	adverbios	
cine	cine	
natural	natural	
palabras	palabras	
introducción		introducción
naturaleza	naturaleza	
jueves	jueves	
actividades		actividades
líquido	líquido	
paisaje	paisaje	
a~la~ley	a~la~ley	
cuidados	cuidados	
madera	madera	
que	que	
línea	línea	
genial	genial	
del~año	del~año	
entrar	entrar	
entonces	entonces	
adrede	adrede	
fuegos~artificiales		fuegos~artificiales
nivel	nivel	
mediados	mediados	
voy	voy	
la~fábrica	la~fábrica	
naturales	naturales	
hacer	hacer	
pagar	pagar	
garbanzos	garbanzos	
industrial	industrial	
baobab	baobab	
trabajos	trabajos	
guerra	guerra	
adverbios	adverbios	
células	células	
policía	policía	
a~jugar	a~jugar	
utilización	utilización	
si~se~tratase		si~se~tratase
me~cuesta	me~cuesta	
significativo		significativo
creía	creía	
huele	huele	
presente	presente	
comprobó	comprobó	
llegaban	llegaban	
llaman	llaman	
accesibilidad		accesibilidad
batir	batir	
aceptado	aceptado	
vez	vez	
hibridación		hibridación
alienígena	alienígena	
necesita	necesita	
vegetación	vegetación	
proceso	proceso	
año	año	
verdadera	verdadera	
hiciste	hiciste	
extranjero	extranjero	
verbos	verbos	
hecha	hecha	
dijeron	dijeron	

ha~roto	ha~roto	
voy	voy	
tarjetas	tarjetas	
perros	perros	
paisaje	paisaje	
ecualizador		ecualizador
cantar	cantar	
accesibilidad		accesibilidad
nivel	nivel	
veintitrés	veintitrés	
tengo	tengo	
habitación	habitación	
nauseabundas		nauseabundas
actrices	actrices	
jugar	jugar	
alrededor	alrededor	
historia	historia	
arboles	arboles	
sílaba	sílaba	
señales	señales	
domingo	domingo	
extensión	extensión	
humano	humano	
recibir	recibir	
extraña	extraña	
descontado		descontado
asteroide	asteroide	
participantes		participantes
clase	clase	
plantita	plantita	
ermita	ermita	
necesitaba	necesitaba	
verdadera	verdadera	
observó	observó	
habían	habían	
giraba	giraba	
congelados		congelados
había	había	
boli	boli	
donde	donde	
significancia		significancia
esta	esta	
filántropo	filántropo	
páginas	páginas	
pacífico	pacífico	
yendo	yendo	
giraba	giraba	
nadie	nadie	
de~gente	de~gente	
ilustrador	ilustrador	
voy~a	voy~a	
subir	subir	
entusiasmo		entusiasmo
mucho	mucho	
inolvidable		inolvidable
señor	señor	
arrastraba	arrastraba	
tecnológicas		tecnológicas
pequeño	pequeño	
entonces	entonces	
sufijo	sufijo	
contento	contento	
noche	noche	
preso	preso	
por~lo~tanto		por~lo~tanto
presente	presente	
martes	martes	
alfabetización		alfabetización
admiten	admiten	
en~el	en~el	
imágenes	imágenes	
verdadera	verdadera	
sigilosamente		sigilosamente
noviembre	noviembre	
relaciona	relaciona	

adjetivo	adjetivo	
verdad	verdad	
memoria	memoria	
extraño	extraño	
cuaderno	cuaderno	
perrito	perrito	
dermatología		dermatología
conversación		conversación
significativos		significativos
hicieron	hicieron	
personas	personas	
empezado	empezado	
variable	variable	
habitación	habitación	
grupos	grupos	
así	así	
de~pollo	de~pollo	
vaciar	vaciar	
sigilosamente		sigilosamente
células	células	
envía	envía	
sigilosamente		sigilosamente
células	células	
se~bañan	se~bañan	
momento	momento	
de~la~ley	de~la~ley	
hará	hará	
no	no	
giraba	giraba	
explica	explica	
preguntó	preguntó	
estrictamente		estrictamente
beneficiario		beneficiario
prehistóricos		prehistóricos
porcentajes		porcentajes
sigilosamente		sigilosamente
voy	voy	
origen	origen	
convencerlos		convencerlos
perfecto	perfecto	
variable	variable	
huevos	huevos	
conmigo	conmigo	
parque	parque	
payaso	payaso	
excepto	excepto	
inauguro	inauguro	
futuro	futuro	
adjetivo	adjetivo	
sigilosamente		sigilosamente
puesta	puesta	
hace	hace	
autobús	autobús	
mechas	mechas	
literales	literales	
adentró	adentró	
justo	justo	
pacífico	pacífico	
discusión	discusión	
civilización		civilización
mundo	mundo	
a~veces	a~veces	
felicidad	felicidad	
veinte	veinte	
iban	iban	
dibujo	dibujo	
inolvidable		inolvidable
variable	variable	
gusta	gusta	
luces	luces	
era	era	
contempló	contempló	
arquitectura		arquitectura
huevos	huevos	
conexión	conexión	

huele	huele	
mochila	mochila	
estaba	estaba	
distintas	distintas	
estaba	estaba	
ayuntamiento		ayuntamiento
lugar	lugar	
satisfecho	satisfecho	
discusión	discusión	
escribimos	escribimos	
peligro	peligro	
cueva	cueva	
excepto	excepto	
cerca	cerca	
año	año	
protozoos	protozoos	
ordenada	ordenada	
persona	persona	
felicidad	felicidad	
también	también	
rosita	rosita	
interior	interior	
necesitar	necesitar	
monosémicas		monosémicas
luces	luces	
palabra	palabra	
también	también	
verdadera	verdadera	
de~litros	de~litros	
sencillo	sencillo	
israelíes	israelíes	
volvió	volvió	
venenosas	venenosas	
desechando		desechando
salida	salida	
gimnasio	gimnasio	
cuales	cuales	
sabemos	sabemos	
interior	interior	
interrogarle		interrogarle
galaxia	galaxia	
conservantes		conservantes
coger	coger	
hombre	hombre	
huevos	huevos	
madera	madera	
beneficiario		beneficiario
hemos	hemos	
cuesta	cuesta	
buscaba	buscaba	
consigue	consigue	
padre	padre	
excepto	excepto	
futbol	futbol	
desinencias		desinencias
desarrollo	desarrollo	
cuerpo	cuerpo	
convencerlos		convencerlos
de~cartón	de~cartón	
se~enfado	se~enfado	
a~cabo	a~cabo	
debilidad	debilidad	
sábado	sábado	
agua	agua	
berenjena	berenjena	
teniendo	teniendo	
porcentajes		porcentajes
alrededor	alrededor	
hojas	hojas	
hago	hago	
hace	hace	
beber	beber	
con~mi	con~mi	
imágenes	imágenes	
adverbios	adverbios	

para~sobrevivir	para~sobrevivir	
asteroide	asteroide	
ecualizar	ecualizar	
participantes		participantes
persuasión	persuasión	
jersey	jersey	
mucho	mucho	
palabras	palabras	
extensión	extensión	
frecuencias		frecuencias
escribimos	escribimos	
aya	aya	
líquido	líquido	
hay	hay	
origen	origen	
berenjena	berenjena	
se~ha	se~ha	
muchos	muchos	
inmovilizó	inmovilizó	
porcentajes		porcentajes
cantaríamos		cantaríamos
jamón	jamón	
llegada	llegada	
tecnologías		tecnologías
esos	esos	
necesita	necesita	
participación		participación
energía	energía	
masculino	masculino	
bajo	bajo	
difícil	difícil	
tiene	tiene	
cuerpo	cuerpo	
miércoles	miércoles	
estaba	estaba	
varias	varias	
giraba	giraba	
gatito	gatito	
leí	leí	
empecé	empecé	
arbolazo	arbolazo	
paquete	paquete	
vamos	vamos	
ejemplo	ejemplo	
convencerlos		convencerlos
disfraces	disfraces	
el~interior	el~interior	
sujeto	sujeto	
manita	manita	
zoo	zoo	
garbanzos	garbanzos	
significativa		significativa
significativo		significativo
años	años	
justo	justo	
conurbano	conurbano	
escribe	escribe	
bella	bella	
por~lo~tanto		por~lo~tanto
disparó	disparó	
arquitectura		arquitectura
ayuntamiento		ayuntamiento
que~te	que~te	
envuelve	envuelve	
vudú	vudú	
herramientas		herramientas
comunicación		comunicación
transportes	transportes	
guerra	guerra	
trimestre	trimestre	
cogía	cogía	
hiciste	hiciste	
verdadera	verdadera	
blanco	blanco	
resultados	resultados	

callejón	callejón	
servicios	servicios	
arquitectura		arquitectura
encuentra	encuentra	
hacemos	hacemos	
manantial	manantial	
era	era	
hermano	hermano	
callejón	callejón	
ilustrador	ilustrador	
observó	observó	
longeva	longeva	
volar	volar	
pequeña	pequeña	
explica	explica	
salida	salida	
beneficiario		beneficiario
contrarreloj		contrarreloj
be	be	
fuera~de	fuera~de	
garbanzos	garbanzos	
necesitaba	necesitaba	
porque	porque	
banco	banco	
he~puesto	he~puesto	
interior	interior	
harán	harán	
hibridación		hibridación
circulante	circulante	
enseño	enseño	
hermano	hermano	
me~regalo	me~regalo	
antropónimos		antropónimos
hayas	hayas	
ha~producido		ha~producido
cuesta	cuesta	
ayer	ayer	
garbanzos	garbanzos	
parecía	parecía	
también	también	
ayuntamiento		ayuntamiento
pequeño	pequeño	
dicen~poco		dicen~poco
páginas	páginas	
ejercicio	ejercicio	
hacemos	hacemos	
mexicana	mexicana	
quedara	quedara	
haber	haber	
de~casa	de~casa	
adjetivo	adjetivo	
fricción	fricción	
carácter	carácter	
hube	hube	
clerecía	clerecía	
octubre	octubre	
hablar	hablar	
sustituir	sustituir	
aliviado	aliviado	
jirafa	jirafa	
señor	señor	
futuro	futuro	
protozoos	protozoos	
cantaríamos		cantaríamos
necesitaba	necesitaba	
proceso	proceso	
hábil	hábil	
quedara	quedara	
jueves	jueves	
frecuencias		frecuencias
me~sé	me~sé	
y~vio	y~vio	
interior	interior	
sigilosamente		sigilosamente

hueco	hueco	
perfecto	perfecto	
padre	padre	
era	era	
accesibilidad		accesibilidad
arboles	arboles	
prerrequisito		prerrequisito
partir	partir	
vociferar	vociferar	
animales	animales	
sonrió	sonrió	
bailar	bailar	
oficina	oficina	
ambiente	ambiente	
cobo	cobo	
triángulos	triángulos	
meses~del	meses~del	
alienígena	alienígena	
queso	queso	
conexión	conexión	
hace	hace	
aúno	aúno	
artilugio	artilugio	
nueva	nueva	
convencerlos		convencerlos
decir	decir	
envuelve	envuelve	
realizarnos	realizarnos	
feria	feria	
dicho	dicho	
bosque	bosque	
dentro	dentro	
de~hierbabuena		de~hierbabuena
hacer	hacer	
por~la~supervivencia		por~la~supervivencia
bastante	bastante	
conversación		conversación
dermatología		dermatología
gusta	gusta	
repente	repente	
libro	libro	
para~poder		para~poder
mayonesa	mayonesa	
piratas	piratas	
municipal	municipal	
garbanzos	garbanzos	
octubre	octubre	
cual	cual	
literales	literales	
agradecerá	agradecerá	
berenjena	berenjena	
pequeño	pequeño	
mentita	mentita	
catastrófica		catastrófica
corazón~a~cien		corazón~a~cien
obtener	obtener	
me	me	
significativos		significativos
habla	habla	
mayores	mayores	
hará	hará	
zapato	zapato	
personajes	personajes	
trabajadora		trabajadora
protectoras		protectoras
cuerpo	cuerpo	
lavavajillas		lavavajillas
participantes		participantes
adverbios	adverbios	
guerra	guerra	
ello~se~debe		ello~se~debe
caballero	caballero	
savias	savias	
deben	deb	

conmigo	conmigo	
berenjena	berenjena	
se~enfado	se~enfado	
conjunto	conjunto	
sigilosamente		sigilosamente
vocal	vocal	
vayamos	vayamos	
berenjenas	berenjenas	
vos~los	vos~los	
regalo	regalo	
me~se	me~se	
perros	perros	
bajita	bajita	
por~la	por~la	
uno	uno	
de~vidrio	de~vidrio	
que~vio	que~vio	
ayer	ayer	
viejo	viejo	
fuegos	fuegos	
envuelve	envuelve	
guerra	guerra	
municipal	municipal	
atún	atún	
sabe	sabe	
señor	señor	
escena	escena	
tuvieran	tuvieran	
composicional		composicional
dibujo	dibujo	
razón	razón	
cuando	cuando	
invariables	invariables	
huele	huele	
muestra	muestra	
extranjero	extranjero	
pasado	pasado	
tierra	tierra	
arquitectura		arquitectura
invariables	invariables	
ya	ya	
sigilosamente		sigilosamente
verdadera	verdadera	
veinte	veinte	
silla	silla	
se~enfado	se~enfado	
burro	burro	
estaban	estaban	
se~ha	se~ha	
ejercicios	ejercicios	
pertenece	pertenece	
agradecerá	agradecerá	
conectarse	conectarse	
desinencias		desinencias
a	a	
pensaba	pensaba	
rejilla	rejilla	
ilustrador	ilustrador	
vidrio	vidrio	
alienígena	alienígena	
gusta	gusta	
distintas	distintas	
sigilosamente		sigilosamente
también	también	
creciente	creciente	
huevos	huevos	
tenis	tenis	
palabra	palabra	
una	una	
confianza	confianza	
comisaría	comisaría	
galaxia	galaxia	
introducción		introducción
agua	agua	
proceso	proceso	

ingredientes		ingredientes
balón	balón	
dedique	dedique	
momento	momento	
gimnasio	gimnasio	
lápiz	lápiz	
gobierno	gobierno	
por~si~solos		por~si~solos
prorrumpir	prorrumpir	
estabas	estabas	
canción	canción	
hay	hay	
viernes	viernes	
antropónimos		antropónimos
sacerdotisa		sacerdotisa
pequeña	pequeña	
accesibilidad		accesibilidad
hicieron	hicieron	
montable	montable	
pasado	pasado	
vio	vio	
extraña	extraña	
octubre	octubre	
prever	prever	
vía	vía	
huevos	huevos	
si~se~tratase		si~se~tratase
proceso	proceso	
debía	debía	
hojalata	hojalata	
civilización		civilización
y~ángel	y~ángel	
conversación		conversación
absorber	absorber	
iban	iban	
distorsionado		distorsionado
beneficioso		beneficioso
cerca	cerca	
alrededor	alrededor	
hacía	hacía	
pudieron	pudieron	
prefijos	prefijos	
compañeros		compañeros
proceso	proceso	
perrera	perrera	
adverbios	adverbios	
acercó	acercó	
verbal	verbal	
vieja	vieja	
se~ha	se~ha	
estricto	estricto	
jueves	jueves	
kimono	kimono	
compró	compró	
gusano	gusano	
deber	deber	
accesibilidad		accesibilidad
ello~se~debe		ello~se~debe
hermana	hermana	
conversación		conversación
admiten	admiten	
jugar	jugar	
fui~al	fui~al	
cuaderno	cuaderno	
noche	noche	
veces	veces	
desobediente		desobediente
siglo	siglo	
hace	hace	
israelíes	israelíes	
necesita	necesita	
industrial	industrial	
hora	hora	
haz	haz	
personas	personas	

C Extracto de la base de datos echo alemán.

Bank	Bank	
Bus	Bus	
Bus	Bus	
Bus	Bus	
Nest	Nest	
Lšwe	Lšwe	
Tisch	Tisch	
Bein	Bein	
Bein	Bein	
Bein	Bein	
Material	Material	
Schmirgelpapier		Schmirgelpapier
Bleistift	Bleistift	
Bleistift	Bleistift	
Bleistift	Bleistift	
Bohrer	Bohrer	
Holzbrett	Holzbrett	
schmirgeln		schmirgeln
schmirgeln		schmirgeln
schmirgeln		schmirgeln
wenn	wenn	
gemacht	gemacht	
Vorlage	Vorlage	
und male	und male	
glŸcklich	glŸcklich	
schwierig	schwierig	
Sachen	Sachen	
Hubschrauber		Hubschrauber
Hubschrauber		Hubschrauber
Hubschrauber		Hubschrauber
eines	eines	
eines	eines	
eines	eines	
Idee	Idee	
Idee	Idee	
Idee	Idee	
Ideen	Ideen	
Holz	Holz	
Holz	Holz	
Holz	Holz	
siehst	siehst	
dir	dir	
fŸhrt	fŸhrt	
entgegen	entgegen	
fŠllt	fŠllt	
ihm	ihm	
verkŸrzt	verkŸrzt	
finanziell	finanziell	
haushalten	haushalten	
wŠre	wŠre	
schnell	schnell	
springen	springen	
liebt	liebt	
liebt	liebt	
liebt	liebt	
rennt	rennt	
rennt	rennt	
rennt	rennt	
fŠhrt	fŠhrt	
fŠhrt	fŠhrt	
fŠhrt	fŠhrt	
viel	viel	
schwingen	schwingen	
backt	backt	
backt	backt	
backt	backt	
Vogelfutter		Vogelfutter
Vogelfutter		Vogelfutter
Vogelfutter		Vogelfutter
Vogelfutter		Vogelfutter

trinkt	trinkt	
vertragen	vertragen	
Klassenvertrag		Klassenvertrag
Klassenvertrag		Klassenvertrag
Klassenvertrag		Klassenvertrag
VerkŠuferin		VerkŠuferin
VerkŠuferin		VerkŠuferin
VerkŠuferin		VerkŠuferin
VerkŠuferin		VerkŠuferin
Spinne	Spinne	
rutscht	rutscht	
rutscht	rutscht	
rutscht	rutscht	
schlank	schlank	
schlank	schlank	
schlank	schlank	
kratzt	kratzt	
versprochen		versprochen
versprochen		versprochen
versprochen		versprochen
versteht	versteht	
versteht	versteht	
versteht	versteht	
klatscht	klatscht	
klatscht	klatscht	
klatscht	klatscht	
Speck	Speck	
FrŠulein	FrŠulein	
FrŠulein	FrŠulein	
FrŠulein	FrŠulein	
Quadrat	Quadrat	
Quadrat	Quadrat	
Quadrat	Quadrat	
krallt	krallt	
blank	blank	
blank	blank	
stŠrkt	stŠrkt	
stŠrkt	stŠrkt	
stŠrkt	stŠrkt	
behandelt	behandelt	
bŸcken	bŸcken	
bŸcken	bŸcken	
bŸcken	bŸcken	
feucht	feucht	
WŠlder	WŠlder	
zwischen	zwischen	
springt	springt	
Kartoffeln	Kartoffeln	
Kartoffeln	Kartoffeln	
Kartoffeln	Kartoffeln	
spucken	spucken	
spucken	spucken	
spucken	spucken	
Strumpf	Strumpf	
Stricknadel		Stricknadel
erkannt	erkannt	
erkannt	erkannt	
erkannt	erkannt	
kŠlter	kŠlter	
kŠlter	kŠlter	
kŠlter	kŠlter	
schluckt	schluckt	
schluckt	schluckt	
beginnt	beginnt	
beginnt	beginnt	
beginnt	beginnt	
klirrt	klirrt	
klirrt	klirrt	
klirrt	klirrt	
gleich	gleich	
Suppe	Suppe	
bleibt	bleibt	

braucht	braucht	
sein	sein	
gestochen	gestochen	
gestochen	gestochen	
gestochen	gestochen	
MŸcke	MŸcke	
schwimmen		schwimmen
wŠrmer	wŠrmer	
Kalb	Kalb	
Kalb	Kalb	
Kalb	Kalb	
streiche	streiche	
fliegt	fliegt	
fliegt	fliegt	
fliegt	fliegt	
steigen	steigen	
trŠgt	trŠgt	
trŠgt	trŠgt	
trŠgt	trŠgt	
dicker	dicker	
stopfen	stopfen	
Kind	Kind	
Mutter	Mutter	
Mutter	Mutter	
Mutter	Mutter	
Vater	Vater	
Katze	Katze	
hšren	hšren	
sprechen	sprechen	
sprechen	sprechen	
sprechen	sprechen	
trocken	trocken	
trocken	trocken	
trocken	trocken	
Menschen	Menschen	
Spiel	Spiel	
Spiel	Spiel	
Spiel	Spiel	
Spiel	Spiel	
Spiele	Spiele	
laufen	laufen	
Platz	Platz	
Platz	Platz	
Platz	Platz	
pfeift	pfeift	
gut	gut	
klatschen	klatschen	
klatschen	klatschen	
klatschen	klatschen	
Ballstaffeln		Ballstaffeln
Ballstaffeln		Ballstaffeln
Ballstaffeln		Ballstaffeln
Ball	Ball	
BŠlle	BŠlle	
fliegt	fliegt	
fliegt	fliegt	
fliegt	fliegt	
fliegt	fliegt	
Tor	Tor	
HŠnde	HŠnde	
dunklen	dunklen	
darunter	darunter	
LaugengebŠck		LaugengebŠck
LaugengebŠck		LaugengebŠck
hungrig	hungrig	
hungrig	hungrig	
hungrig	hungrig	
Šngstlich	Šngstlich	
Šngstlich	Šngstlich	
freundlich	freundlich	

Ketten	Ketten
Ketten	Ketten
Ketten	Ketten
versteckte	versteckte
versteckte	versteckte
versteckte	versteckte
schnell	schnell
legten	legten
legten	legten
legten	legten
schlafen	schlafen
glatt	glatt
zufrieden	zufrieden
dick	dick
rund	rund
hŸbsches	hŸbsches

Kohlenkeller	Kohlenkeller
Kohlenkeller	Kohlenkeller
Kohlenkeller	Kohlenkeller
netten	netten

Gespenster	Gespenster

blitzen	blitzen
erzŠhlten	erzŠhlten

Gruselgeschichten	Gruselgeschichten
Gruselgeschichten	Gruselgeschichten
Gruselgeschichten	Gruselgeschichten
lebte	lebte

Mitternacht	Mitternacht
Mitternacht	Mitternacht
Mitternacht	Mitternacht
jetzt	jetzt

versteckten	versteckten
versteckten	versteckten
versteckten	versteckten
versteckten	versteckten
ihrem	ihrem
ihrem	ihrem
ihrem	ihrem
liest	liest
liebsten	liebsten
liebsten	liebsten
liebsten	liebsten
GŠnge	GŠnge
Buckel	Buckel
Buckel	Buckel
Buckel	Buckel
guckt	guckt
guckt	guckt
guckt	guckt
guckt	guckt
Narbe	Narbe
Kinn	Kinn
Kinn	Kinn
Kinn	Kinn

furchterregend	furchterregend

sieht	sieht
sieht	sieht
sieht	sieht
gefŠllt	gefŠllt
gefŠllt	gefŠllt
gefŠllt	gefŠllt
helle	helle
dieser	dieser
wird	wird
wieder	wieder
Moos	Moos
Tulpen	Tulpen
sehen	sehen
Beet	Beet
Erdbeeren	Erdbeeren
Erdbeeren	Erdbeeren
Erdbeeren	Erdbeeren
Lšcher	Lšcher
Lšcher	Lšcher
Lšcher	Lšcher

Farbe	Farbe
Farbe	Farbe
Farbe	Farbe
Anleitung	Anleitung
wollen es	wollen es
wollen es	wollen es
wollen es	wollen es
wollen	wollen
runter	runter
ihre	ihre
Haare	Haare
mussten	mussten
nŠchsten	nŠchsten
nŠchsten	nŠchsten
nŠchsten	nŠchsten
half	half
gehen	gehen
gehen	gehen
gehen	gehen
gehen	gehen
gehen	gehen
gehen	gehen
alle	alle
alle	alle
alle	alle
alle	alle
Anpfiff	Anpfiff
Anpfiff	Anpfiff
Anpfiff	Anpfiff
Anpfiff	Anpfiff
stark	stark
stark	stark
stark	stark
stŠrker	stŠrker
wŠhrend	wŠhrend
wŠhrend	wŠhrend
wŠhrend	wŠhrend

kaufmŠnnisch	kaufmŠnnisch
kaufmŠnnisch	kaufmŠnnisch
kaufmŠnnisch	kaufmŠnnisch
hauptsŠchlich	hauptsŠchlich
hauptsŠchlich	hauptsŠchlich
hauptsŠchlich	hauptsŠchlich
kontrollieren	kontrollieren
kontrollieren	kontrollieren
kontrollieren	kontrollieren
kontrollieren	kontrollieren
kontrollieren	kontrollieren
unterschiedlich	unterschiedlich
unterschiedlich	unterschiedlich
unterschiedlich	unterschiedlich
unterschiedlich	unterschiedlich

Deutschland	Deutschland

deutsche	deutsche
englische	englische
gehaltvoll	gehaltvoll
gehaltvoll	gehaltvoll
gehaltvoll	gehaltvoll
gehabt	gehabt
gehabt	gehabt
gehabt	gehabt
passiv	passiv
passiv	passiv
passiv	passiv
Ÿber	Ÿber
erstatten	erstatten
vielleicht	vielleicht
interessant	interessant
interessant	interessant
interessant	interessant
interessant	interessant
TrŠnen	TrŠnen
wohnlich	wohnlich

vielschichtig	vielschichtig
vielschichtig	vielschichtig

vielschichtig	vielschichtig
vielschichtig	vielschichtig
vielschichtig	vielschichtig
wŸtend	wŸtend
ErlŠuterung	ErlŠuterung
ErlŠuterung	ErlŠuterung
ErlŠuterung	ErlŠuterung
mŸhsam	mŸhsam
gehŠuft	gehŠuft
sieht	sieht
Stab	Stab
Hof	Hof
glŠnzte	glŠnzte
Vieh	Vieh
Vieh	Vieh
Vieh	Vieh
Vieh	Vieh
Proviant	Proviant
Proviant	Proviant
Proviant	Proviant
Proviant	Proviant
Proviant	Proviant

Fu§ballmannschaft	Fu§ballmannschaft
Fu§ballmannschaft	Fu§ballmannschaft
Fu§ballmannschaft	Fu§ballmannschaft
Fu§ballmannschaft	Fu§ballmannschaft
Fu§ballmannschaft	Fu§ballmannschaft
Mannschaftsspiele	Mannschaftsspiele
Mannschaftsspiele	Mannschaftsspiele
Mannschaftsspiele	Mannschaftsspiele
Fu§ballturnier	Fu§ballturnier
Fu§ballturnier	Fu§ballturnier
Fu§ballturnier	Fu§ballturnier
TischtennisschlŠger	TischtennisschlŠger
TischtennisschlŠger	TischtennisschlŠger
TischtennisschlŠger	TischtennisschlŠger
TischtennisschlŠger	TischtennisschlŠger
TischtennisschlŠger	TischtennisschlŠger
Rei§verschluss	Rei§verschluss
Rei§verschluss	Rei§verschluss
Rei§verschluss	Rei§verschluss
Rei§verschluss	Rei§verschluss
Rei§verschluss	Rei§verschluss
Geburtstagsgeschenk	Geburtstagsgeschenk
Geburtstagsgeschenk	Geburtstagsgeschenk
Geburtstagsgeschenk	Geburtstagsgeschenk
Geburtstagsgeschenk	Geburtstagsgeschenk
Geburtstagsgeschenk	Geburtstagsgeschenk
Fernsehprogramm	Fernsehprogramm
Fernsehprogramm	Fernsehprogramm
Fernsehprogramm	Fernsehprogramm
Fernsehprogramm	Fernsehprogramm

Grab	Grab

Spinnennetz	Spinnennetz
Spinnennetz	Spinnennetz
Spinnennetz	Spinnennetz
Spinnennetz	Spinnennetz
Spinnennetz	Spinnennetz
reagiert	reagiert
verschieben	verschieben
gesellt	gesellt
trauerten	trauerten
Torwart	Torwart
Torwandschie§en	Torwandschie§en
Schiedsrichter	Schiedsrichter
Schiedsrichter	Schiedsrichter
Schiedsrichter	Schiedsrichter
Schiedsrichter	Schiedsrichter
Schiedsrichter	Schiedsrichter
Schiedsrichter	Schiedsrichter
Schiedsrichter	Schiedsrichter
dauernd	dauernd
doofe	doofe
doofe	doofe

doofe	doofe	
doofe	doofe	
Computer	Computer	
kaputt	kaputt	
pflegt	pflegt	
Tiger	Tiger	
Geschichten		Geschichten
Geschichten		Geschichten
Geschichten		Geschichten
erzŠhlen	erzŠhlen	
erzŠhlen	erzŠhlen	
erzŠhlen	erzŠhlen	
erzŠhlt	erzŠhlt	
mutig	mutig	
niemand	niemand	
niemand	niemand	
niemand	niemand	
frŸhstŸcken		frŸhstŸcken
frŸhstŸcken		frŸhstŸcken
frŸhstŸcken		frŸhstŸcken
frŸhstŸcken		frŸhstŸcken
frŸhstŸcken		frŸhstŸcken
frŸhstŸcken		frŸhstŸcken
zusammen	zusammen	
zusammen	zusammen	
sammelten	sammelten	
glŸcklich	glŸcklich	
krŸmeln	krŸmeln	
krŸmeln	krŸmeln	
krŸmeln	krŸmeln	
knuddeln	knuddeln	
Blume	Blume	
rieche	rieche	
kennengelernt		kennengelernt
kennengelernt		kennengelernt
kennengelernt		kennengelernt
Hoffnung	Hoffnung	
endlich	endlich	
waren	waren	
ziemlich	ziemlich	
Munition	Munition	
Munition	Munition	
Munition	Munition	
Munition	Munition	
bevor	bevor	
bevor	bevor	
bevor	bevor	
fehlte	fehlte	
Packung	Packung	
šsterreichischen		šsterreichischen
weil	weil	
konnte	konnte	
arbeiteten	arbeiteten	
Ritter	Ritter	
abliefern	abliefern	
lebten	lebten	
MŠdchen	MŠdchen	
Gestalt	Gestalt	
und	und	
wirkte	wirkte	
spŠter	spŠter	
spŠter	spŠter	
spŠter	spŠter	
spŠter	spŠter	
Essen	Essen	
holst	holst	
SchlickertŸte		SchlickertŸte
ersten	ersten	
Fussball	Fussball	
da	da	
wenn	wenn	
Verlierer	Verlierer	
Anstoss	Anstoss	
Anstoss	Anstoss	

Anstoss	Anstoss	
Anstoss	Anstoss	
Anstoss	Anstoss	
alle	alle	
dadurch	dadurch	
Sekunden	Sekunden	
Sekunden	Sekunden	
Sekunden	Sekunden	
stutzt	stutzt	
anschliessen		anschliessen
anschliessen		anschliessen
anschliessen		anschliessen
anschliessen		anschliessen
wollte	wollte	
geschubst	geschubst	
hatte	hatte	
letzte	letzte	
letzte	letzte	
letzte	letzte	
Pause	Pause	
besten	besten	
kam	kam	
Freundin	Freundin	
MŠdchen	MŠdchen	
versucht	versucht	
unser Problem		unser Problem
unser Problem		unser Problem
unser Problem		unser Problem
unser Problem		unser Problem
Problem	Problem	
Brille	Brille	
sieht	sieht	
sieht	sieht	
sieht	sieht	
nicht	nicht	
hšrt	hšrt	
prŸgeln	prŸgeln	
prŸgeln	prŸgeln	
prŸgeln	prŸgeln	
prŸgeln	prŸgeln	
Bescheid	Bescheid	
Bescheid	Bescheid	
Bescheid	Bescheid	
FahrgŠste	FahrgŠste	
FahrgŠste	FahrgŠste	
FahrgŠste	FahrgŠste	
brŸllte	brŸllte	
fragte	fragte	
stellte	stellte	
stellte	stellte	
stellte	stellte	
Zorn	Zorn	
Gšttervater		Gšttervater
durchgefŸhrt		durchgefŸhrt
Zeit	Zeit	
allgemeiner		allgemeiner
allgemeiner		allgemeiner
allgemeiner		allgemeiner
ganz	ganz	
teilnehmen		teilnehmen
freie	freie	
Teilnahme	Teilnahme	
Teilnehmer		Teilnehmer
Teilnehmer		Teilnehmer
Teilnehmer		Teilnehmer
galt	galt	
Ehre	Ehre	
vertrat	vertrat	
trainiert	trainiert	
Zuschauer	Zuschauer	
hervorging	hervorging	
hervorging	hervorging	
hervorging	hervorging	
ging	ging	
ging	ging	

ging	ging	
heimkehrten		heimkehrten
Beweis	Beweis	
atmen	atmen	
mŠchtige	mŠchtige	
Auftrag	Auftrag	
SŸssigkeiten		SŸssigkeiten
Sachen	Sachen	
kriegte	kriegte	
teilen	teilen	
nahm	nahm	
Bonbons	Bonbons	
das	das	
kannst	kannst	
drau§en	drau§en	
au§erdem	au§erdem	
au§erdem	au§erdem	
au§erdem	au§erdem	
au§erdem	au§erdem	
finden	finden	
und das	und das	
všllig	všllig	
všllig	všllig	
všllig	všllig	
um die	um die	
dann	dann	
presst	presst	
fŸllst	fŸllst	
soviel	soviel	
FlŸssigkeit		FlŸssigkeit
verdŸnnten		verdŸnnten
schŸtte	schŸtte	
nimm	nimm	
nimm	nimm	
nimm	nimm	
stecke	stecke	
haaren	haaren	
breiter	breiter	
Mund	Mund	
Kleid	Kleid	
Kleid	Kleid	
Kleid	Kleid	
schwarzen	schwarzen	
gelb	gelb	
Strumpf	Strumpf	
reitet	reitet	
Abenteuer	Abenteuer	
Entdeckungsreisen		
	Entdeckungsreisen	
Affen	Affen	
hat	hat	
geschŠtzt	geschŠtzt	
offene	offene	
Nase	Nase	
schmŸckt	schmŸckt	
T-Shirt	T-Shirt	
T-Shirt	T-Shirt	
T-Shirt	T-Shirt	
sie	sie	
Leggings	Leggings	
Leggings	Leggings	
Leggings	Leggings	
Leggings	Leggings	
zielte	zielte	
Zeit	Zeit	
dagestanden		dagestanden
dagestanden		dagestanden
dagestanden		dagestanden
Bewegung	Bewegung	
Revolver	Revolver	
Revolver	Revolver	
Revolver	Revolver	
anschlie§end		anschlie§end
hob	hob	
rief	rief	
altes	altes	