UNIVERSIDAD
REY JUAN CARLOS

MÁSTER EN INGENIERÍA EN SISTEMAS DE DECISIÓN

Curso Académico 2018/2019

Trabajo Fin de Máster

Marco de trabajo para evaluar la relevancia de artículos de dominio científico

Autor : Adrián Alonso Barriuso

Tutor : Dr. Alberto Fernández Isabel

Dedicado a

mi sobrino, Alberto, algún día te explicaré todo esto....

Agradecimientos

Gracias a mi familia y amigos por estar ahí cuando había que estar. Gracias a todos los compañeros del DSLAB por vuestra ayuda e inspiración, mención especial para Isaac. Gracias a Medlab Media Group y a mi tutor, Alberto, por vuestra confianza y apoyo.

Resumen

La comunidad investigadora se enfrenta cada vez a un mayor número de publicaciones y tendencias que deben atender a la hora hacer sus propias publicaciones. Estos tópicos y tendencias son estado del arte en el momento de su publicación o presentación en conferencias, no obstante, pueden perder relevancia con el paso del tiempo. Esta medida de relevancia de publicaciones representa un desafío para la comunidad investigadora, la cuál invierte mucho tiempo leyendo literatura a menudo desfasada o poco relevante. Para abordar este problema, se introduce un marco de trabajo para evaluar la relevancia de artículos científicos a través de, principalmente, un lexicón y una red neuronal. Se han llevado a cabo diversos experimentos aplicados al dominio de la medicina que demuestran el buen funcionamiento del marco de trabajo.

Palabras clave: Medida de relevancia, Generación de diccionario, Reputación de artículos, Relevancia científica, Sistema basado en conocimiento.

Abstract

The research community is faced with an increasing number of publications and trends that need to be taken into account when making their own publications. These topics and trends are state of the art at the time of publication or presentation at conferences, however, they may lose relevance over time. This measure of publication relevance represents a challenge for the research community, which spends a lot of time reading often outdated or irrelevant literature. To address this problem, a framework is introduced to assess the relevance of scientific articles primarily through a lexicon and neural network. Several experiments applied to the field of medicine have been carried out that demonstrate the good performance of the framework.

Keywords: Relevance metric, Dictionary generation, Article reputation, Scientific relevance, Knowledge based system

Índice general

Índice de figuras

Capítulo 1

Introducción

En este capítulo introductorio se presenta, en primer lugar, el contexto científico sobre el que se enmarca el presente proyecto. Después, se introduce el objetivo principal y finalmente los objetivos específicos.

1.1. Contexto

La comunidad investigadora invierte la mayor parte de su tiempo documentándose, investigando literatura previa y el estado del arte. A menudo, la relevancia de los diferentes tópicos sobre los que se investiga fluctúa en gran medida con el tiempo, lo que provoca que, en ocasiones, los investigadores pierdan mucho tiempo con literatura desfasada o poco relevante. Por tanto, este proyecto introduce un marco de trabajo completo que tiene como finalidad organizar y medir la relevancia de artículos de investigación.

Dado que se trabaja con información textual, se utilizan técnicas de Procesamiento de Lenguaje Natural (NLP, por sus siglas en inglés) [1] que sirven para poder tratar con este tipo de información desde un punto de vista computacional. A través de estas técnicas, las cuáles se combinan con métricas de reputación [2], se construye un lexicón de relevancias para evaluar la relevancia de los documentos de un modo similar a un análisis de sentimientos [3]. Para complementar este lexicón, se emplean técnicas de Deep Learning [4] para evaluar la relevancia de los términos cuya relevancia sea desconocida por el lexicón y la reputación de cada artículo en caso de estar disponible.

1.1.1. Dominio de aplicación

Aunque este marco de trabajo puede ser entrenado para cualquier dominio de investigación científica, los experimentos se han llevado a cabo en el dominio médico. Se han descargado y parseado mas de dos millones de artículos de investigación médica provenientes de Pubmed Central . Cabe destacar que, aunque los artículos de investigacion podrían ser considerados un dominio de aplicación por si mismo, la ciencia abarca demasiados temas para que el sistema funcione de forma deseable, por tanto, el sistema sólo debe ser utilizado para evaluar los documentos del mismo dominio de aplicación concreto con el que haya sido entrenado.

1.2. Objetivos

El principal objetivo del presente trabajo es la creación del sistema completo de evaluación de relevancias, lo que comprende un marco de trabajo que incluye la interfaz para la introducción de documentos y a la salida devuelva la relevancia de los mismos.

1.2.1. Objetivos específicos

- Obtención del corpus de documentos científicos: Se necesita un gran volumen de documentos para la creación del lexicón, el entrenamiento de la red neuronal y para la validación del sistema.

- Limpieza y almacenaje de los documentos: Una vez descargados, deben ser limpiados y debidamente almacenados en base de datos. Se almacenan con los metadatos necesarios para la creación del lexicón y la red neuronal, como la fecha, el DOI, el resumen y la reputación de cada artículo.

- Construcción del lexicón de relevancias: Se aplican diversas métricas para la creación del lexicón. A mayor volumen de documentos utilizado, mayor riqueza y precisión del lexicón, lo que implica una menor dependencia de la red neuronal.

- Construcción de la red neuronal: Debido a las limitaciones del lexicón, ya sea por limitaciones computacionales o de volumen de corpus considerado, se crea una red neuronal de apoyo para predecir las relevancias fuera del lexicón.

- Creación del flujo de evaluación de relevancia: Se analizan documentos o textos nuevos combinando las métricas de relevancia proporcionadas por el lexicón, la red neuronal y la reputación para estimar la relevancia de los mismos.

1.3. Estructura de la memoria

La memoria tiene los siguientes capítulos fundamentales:

1. Estado del arte: En este capítulo se revisa la literatura en la que se apoya el siguiente trabajo, explicando ideas utilizadas y otras descartadas.

2. Propuesta: Se trata del capítulo central y más importante, en el cuál se describe la propuesta del marco de trabajo completo.

3. Experimentos y resultados: Se describe el caso uso del marco de trabajo, presentando los diferentes experimentos realizados.

4. Conclusiones: En este capítulo final se evalúan los resultados obtenidos y objetivos completados. También se revisan posibles líneas futuras de investigación y mejora.

Capítulo 2

Estado del arte

En este capítulo se hace un análisis de la literatura relacionada con las diferentes características del presente trabajo. Se describen ideas utilizadas, modificadas y/ó descartadas con sus correspondientes referencias.

2.1. Relevancia de documentos

Se define relevancia como importancia o cualidad de destacable. A lo largo de los años, se han propuesto diferentes aproximaciones matemáticas que determinan la relevancia de un texto, concepto o término. Debido a su simplicidad, eficacia y bajo coste computacional, el algoritmo *TF-IDF*(Term Frequency - Inverse Document Frequency) [5] ha sido el más utilizado por la comunidad científica. Parte de la asunción de que cuanto más sea repetido un término o concepto dentro de un documento, más relevante es (Term Frequency) dentro del mismo. No obstante, si ese término es muy común en un conjunto de textos, su relevancia baja (Inverse Documento Frequency). A pesar de sus virtudes, sólo tener en cuenta la parte frecuencial de los términos le resta potencial, de ahí que la comunidad científica haya propuesto numerosas modificaciones [6] con resultados desiguales para resolver sus limitaciones, como el STW (supervised term weighting) [7], TF-RF(Term Frequency Relevance Frequency) [8] o el más reciente TF-IGM [9]. Todas estas aproximaciones son interesantes, pero dada su mayor complejidad y falta de continuidad de uso por la comunidad investigadora, se ha optado por la utilización del TF-IDF, quedando su comparación con el resto de aproximaciones para trabajos futuros.

2.2. Generación automática de lexicones

La generación automática de lexicones de propósito específico es un desafío al que la comunidad científica lleva años enfrentándose. La mayoría de lexicones creados hasta la fecha están elaborados, en el mejor de los casos, con métodos semi supervisados [10]. La mayoría de las aproximaciones realizadas tienen como objetivo el análisis de sentimientos [11], siendo el mayor exponente SentiWordNet [12], el cuál está creado con técnicas de aprendizaje supervisado. Hay alguna propuesta novedosa de ténicas automáticas, como la introducida por [13], que tiene en cuenta caractéristicas sintácticas y semánticas de las palabras para construir un diccionario de sentiemientos. Parte del presente trabajo requiere de un lexicón generado de manera automática, pero de relevancias, no de sentimientos, por tanto este tipo de aproximaciones u otras mas recientes como la de [14] no son aplicables. Por otra parte, otras aproximaciones de gran relevancia actual, como la última versión de SenticNet [15] , emplean redes neuronales recurrentes (RNN) para la construcción de su particular representación del conocimiento para la creación de un diccionario de sentimientos.

2.3. Algoritmos de reputación

Los algoritmos de reputación tienen como objetivo cuantificar la confianza de una persona u objeto específico en base a información previa [2]. Revisando la literatura más reciente al respecto, [16] propone un marco de trabajo visual para dar soporte a la comunidad científica. Una de sus principales funcionalidades es la de proporner un algoritmo de reputación de artículos de investigación, en base a métricas como la veteranía de los autores y el número de citas.

Otras aproximaciones, como la propuesta en en artículo [17], la cuál obtiene la reputación de críticas de películas, en base a principios similares a los empleados en la anterior publicación, como el número de citas de las críticas y el contexto en el que se encuentran, no obstante, los metadatos empleados para el cálculo sólo tienen sentindo dentro de este dominio de aplicación y no están disponibles para el presente caso.

2.4. Redes neuronales aplicadas a texto

Las redes neuronales sólo admiten números como entrada, por tanto, para que puedan ser usadas con información textual, se debe encontrar una forma de codificar el texto en números y vectores. La forma más sencilla de hacerlo podría ser la de asignar un número único de forma arbitraria a cada una de las diferentes palabras del corpus. Esto puede parecer suficiente para que la red neuronal aprenda patrones y pueda hacer predicciones. No obstante, esta codificación no contiene ninguna información sobre cómo las palabras se relacionan entre sí de manera semántica, por tanto se utilizan aproximaciones como el *word embedding* [18]. Lo cuál consiste en mapear las relaciones semánticas de palabras en vectores, de forma que por ejemplo, los sinónimos tendrán vectores parecidos. Esta representación geométrica de el lenguaje elimina ruido y facilita el trabajo de las redes neuronales. Se ha utilizado un word embeding pre-entrenado con un vocabulario de 400000 palabras de 100 dimensiones de dominio general provenientes de Glove [19], queda por tanto, el entrenamiento propio de word embeddings para trabajos futuros. Más allá de representación óptima de palabras como vectores, la información textual tiene un orden temporal o secuencial, por lo que la utilización de redes neuronales convolucionales (CNN) [20] es una buena solución para tareas de clasificación de texto por encima de las redes neuronales recurrentes, ya que ofrecen un rendimiento similar con un menor coste computacional [21].

Capítulo 3

Propuesta

En este capítulo, se describe la propuesta del sistema completo, definiendo entradas y salidas explicadas a nivel de diseño. Se empieza con una subsección donde se ve la arquitectura y el propósito general y después se entra en detalle para cada uno de los módulos en las subsiguientes secciones.

3.1. Arquitectura general

En la Figura 3.3, se pueden observar el módulo principal: *Estimador de relevancia*, que es el encargado de estimar las relevancias de los artículos de entrada, y sus correspondientes submódulos: *Procesador de texto, Calculador de reputación y Calculador de relevancia*. Se cuenta además, con dos fuentes de información precalculadas utilizadas por el submódulo *Calculador de relevancia*, a saber, *Lexicón de relevancia* que consiste en cuatro lexicones[22] de relevancias de términos médicos y *Red neuronal*, que consiste en cuatro modelos entrenados de redes neuronales convolucionales (CNN)[20] por sus siglas en inglés. Por otra parte, se cuenta con un módulo de visualización(*Visualización*), que se utiliza para ver la salida del sistema y para proporcionar la entrada (*Texto*). Por último, se utiliza una fuente de información externa en tiempo real.

En primer lugar se entra en detalle en cómo se construye Lexicón de relevancia, después Red neuronal y, por último, el módulo de estimación de relevancia completo, el cuál emplea todo lo anterior para calcular las relevancias de nuevos documentos.

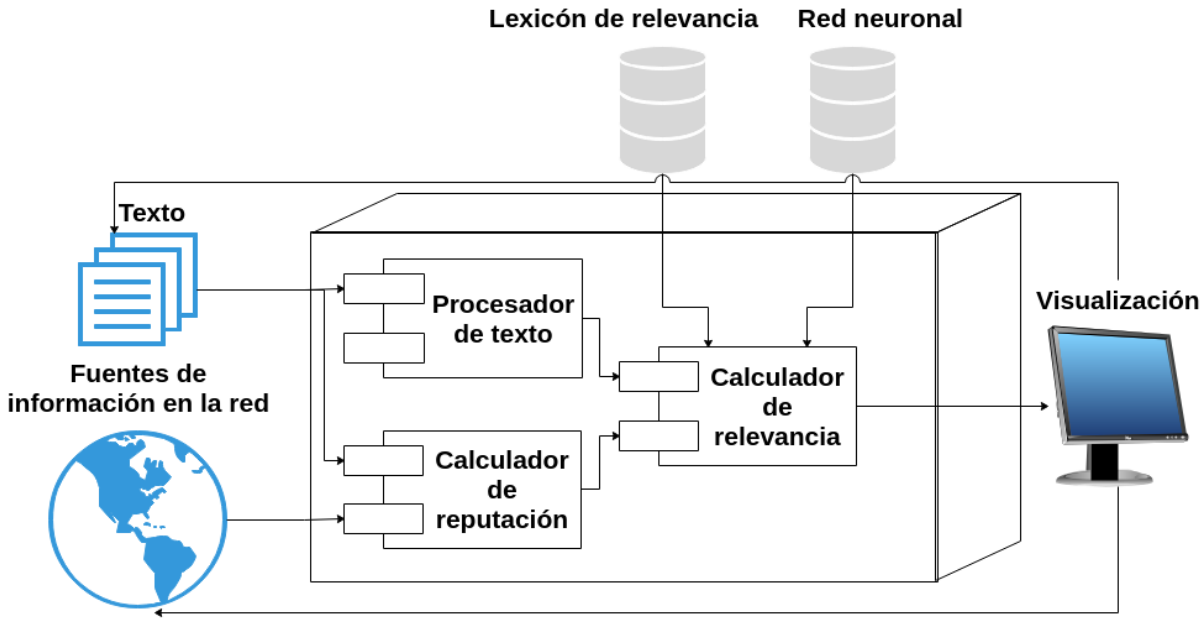


Figura 3.1: Arquitectura general

3.2. Creación del lexicón de relevancias

El lexicón de relevancias tiene un papel fundamental a la hora de estimar la relevancia de los artículos, en concreto, contiene la relevancia de las palabras del dominio de aplicación utilizado para crear el mismo. Para la creación del lexicón se ha diseñado un marco de trabajo completo a parte, cuya arquitectura puede verse en la Figura 3.2. Esta cuenta con un módulo de extracción, transformación y carga (ETL por sus siglas en inglés)[23], un módulo de gestión de conomicimiento y uno de visualización. Se cuenta, además, con dos bases de datos, una orientada a documentos[24](Consolidación de información de documentos) y una ElasticSearch[25] para visualización con Kibana [26].

3.2.1. Módulo ETL

El módulo ETL se encarga de la obtención y el preprocesado del corpus de artículos médicos. Se divide a su vez en dos submódulos: Procesador del corpus y calculador de reputación. Corpus processor obtiene artículos de Pubmed Central[1] utilizando técnicas de web scrapping

[1] `https://www.ncbi.nlm.nih.gov/pmc/`

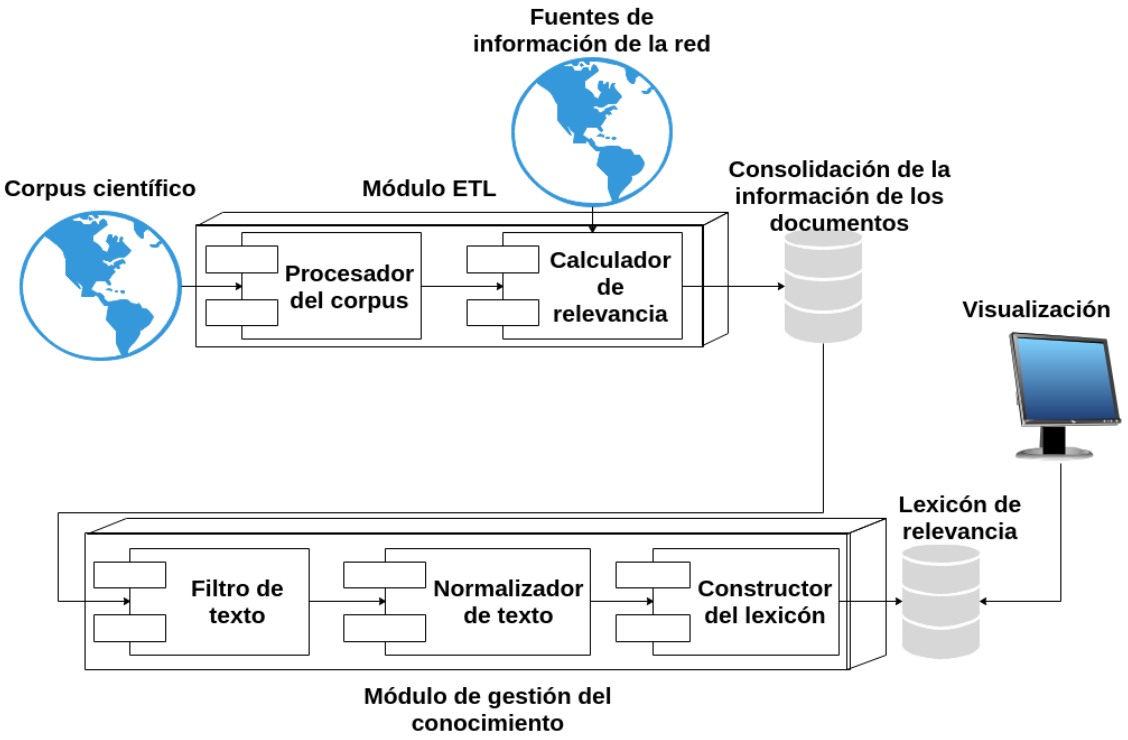


Figura 3.2: Arquitectura de generación de lexicones

[27]. Estos artículos en formato XML [28], son parseados y almacenados en formato JSON [29] en Consolidación de información de documentos con una estructura de párrafos y frases, eliminando tablas, gráficas u otros artefactos no aprovechables por el sistema. Una vez un documento se almacena en la base de datos, se procede a calcular su resumen automático aplicando el clásico algoritmo Text Rank [30] utilizando una popular implementación en Python [31]. Este tipo de resúmenes ofrecen una ordenación de las frases de un documento por relevancia, sirviendo como factor de filtrado de información poco relevante o redundante y a la vez como factor de compresión, haciendo la información más manejable en memoria. Se obtiene un 20 porciento del tamaño original de los artículos y se almacena como un nuevo artibuto del documento en la base de datos.

Una vez preprocesados y almacenados los artículos, *Calculador de reputación* calcula la reputación de los mismos para que esta sea añadida como nuevo atributo y ser utilizada posteriormente por el Módulo de gestión del conocimiento. El algoritmo de reputación empleado es una adaptación del introducido en el artículo [16]. La reputación a priori de un artículo viene

dada por:

$$rep_p = \alpha \cdot rep\_authors_p + (1 - \alpha) \cdot citations_p\,, \tag{3.1}$$

donde $rep\_authors_p$ es la reputación media del los autores del artículo y $citations_p$ su número de citas total en el momento de la consulta. El parámetro $\alpha \in (0, 1)$ actúa como modulador de importancia relativa entre las citas y los autores. La reputación de cada autor viene dada por:

$$rep_i = \omega_1 \cdot inf\_citation\_count + \omega_2 \cdot citation\_velocity + \omega_3 \cdot seniority + \omega_4 \cdot papers\,, \tag{3.2}$$

donde $\sum_{i=1}^{4} \omega_i = 1$. El parámetro $inf\_citation\_count$ representa el número de citas altamente influyentes del autor [32]. $citation\_velocity$ indica lo popular que es el autor durante los últimos 3 años. $seniority$ es la cantidad de años transcurridos entre la primera y última publicación del autor. Por último, $papers$ es el número total de artículos publicados por el autor. Estos parámetros son extraídos de la API REST de Semantic Scholar[33] a través del Identificador Digital de Objeto (DOI, por sus siglas en inglés)[34] del documento. Una vez que el módulo Reputation Calculator ha calculado la reputación del artículo, esta se añade a *Consolidación de información de documentos*, como un atributo nuevo.

3.2.2. Módulo de Gestión del conocimiento

Este módulo consulta *Consolidación de información de documentos* y sirviéndose de 3 submódulos, construye finalmente el lexicón. Estos 3 submódulos son Filtro de texto, Normalizador de texto y Constructor del lexicón.

3.2.3. Filtro de texto

Este módulo analiza los resúmenes extraidos por el módulo ETL por cada documento, aplicando técnicas de procesamiento de lenguaje natural (NLP) [1] extrayendo los sustantivos y eliminando palabras vacías [35], tanto genéricas como de dominio médico[36] y académico[37]. Finalmente, se obtienen los lemas de los sustantivos, lo que permite cierta desambiguación, ya que el lema de una palabra depende de la función sintáctica de la misma. Además, se resuelven posibles conflictos entre mayúsculas, minúsculas, singulares y plurales.

3.2.4. Normalizador de texto

Este submódulo construye una matriz de términos por documentos [38] a partir de la cuál construye la matriz TF-IDF [5]. Los pesos resultantes se combinan con las reputaciones de los artículos dando la medida de relevancia de cada término rel_lex_t. La relevancia de cada término t en los N artículos pertenecientes al corpus C viene dada por:

$$rel\_lex_t = \log\left(\frac{1}{N}\cdot\sum_{p=1}^{N}\beta\cdot tfidf(t)_p + (1-\beta)\cdot rep_p\right), \forall p \in C\,, \tag{3.3}$$

donde rep_p es la reputación del artículo p, $tfidf(t)_p$ es el valor TF-IDF del término t en el artículo p y $\beta \in (0,1)$ es otro parámetro que modula la importancia relativa del valor TF-IDF sobre la reputación. Cabe destacar que $tfidf(t)_p \in (0,1)$ ya que han sido normalizados por simplicidad y se aplica el logaritmo para normalizar la distribución.

3.2.5. Constructor del lexicón

Este componente construye y organiza el lexicón a partir del texto normalizado. Contempla además la posibilidad de ponderar aún más el peso de aquellos términos de dominio específico proporcionados por un diccionario, médico en este caso [39].

Se organizan y construyen lexicones por cada conjunto de artículos correspondientes a cada año disponible, teniendo en cuenta la evolución de la relevancia de cada término a lo largo de los años. De esta manera se puede modular la curva de olvido [40] y tener en consideración las tendencias del dominio de aplicación.

Se construyen diferentes valores de relevancia para cada término específico $rel\_lex_t(y)$ de acuerdo a un año específico y, manteniendo las palabras del año anterior en el nuevo de la forma:

$$rel\_lex_t(y) = \rho\cdot rel\_lex_t + (1-\rho)\cdot rel\_lex_t(y-1)\,, \tag{3.4}$$

donde rel_lex_t es la relevancia proporcionada por el marco de trabajo para el término t y $rel\_lex_t(y-1)$ es la correspondiente al año anterior $y-1$. El parámetro ρ controla el peso del año anterior.

3.3. Creación del la red neuronal

Por muy grande que sea el corpus que se utilice para la creación de los lexicones, es virtualmente imposible contar con la relevancia de todos los términos existentes. Por tanto, se ha empleado una Red Neuronal Convolucional (CNN, por sus siglas en inglés)[20] para servir de apoyo al sistema. El propósito de la red es predecir la relevancia de aquellas frases que no contengan ninguna palabra presente en el lexicón. En la Tabla 3.1 se puede ver la configuración de la misma, la cuál es una versión optimizada de la aproximación introducida en [41]. Cabe destacar que la capa de *embedding* utiliza un modelo pre-entrenado de Glove [19] con un vocabulario de $400,000$ palabras de Wikipedia [42]. La capa de salida cuenta con activación *softmax*, la cuál proporciona valores entre 0 y 1. En las capas ocultas se cuenta con convoluciones, funciones de activación *relu* y funciones de *dropout*.

Para construir la red, se propone la siguiente metodología: En primer lugar, el usuario selecciona un conjunto de artículos del corpus que no hayan sido utilizados para construir el lexicón. Después, se procesa el texto separando por frases y términos, eliminando palabras vacías y lematizando de manera análoga a como se procede en el módulo de ETL, de esta manera se aumenta el número potencial de coincidencias entre el texto utilizado para la creación de la red y las palabras del lexicón. Se etiquetarán como relevantes (1) aquellas frases con mayor relevancia de acuerdo al lexicón y como no relevantes (0) las de menor relevancia o aquellas que no contengan palabras del lexicón. Se define, por tanto, un umbral de relevancia ϵ para elegir el mínimo necesario de acuerdo a lo estricta que se quiere que sea la red neuronal. Finalmente se debe elegir el número de frases etiquetadas para conformar el conjunto de entrenamiento y test. La red neuronal resultante debería ser capaz de prececir la relevancias de las frases sin palabras del lexicón.

Layers
1. Embedding input_dim 400000 output_dim 50
2. Dropout rate 0.4
3. Conv1D 250 filters of 3 with stride 1
4. Pool1D (max) with stride 1
5. Dense units 250
6. Dropout rate 0.4
7. Relu
8. Dense units 1
9. Softmax

Tabla 3.1: Capas de la red neuronal convolucional.

3.4. Estimación de relevancias de artículos

Se ha diseñado un flujo de trabajo para ilustrar como funciona todo el proceso del marco de trabajo. Se comienza eligiendo un texto para evaluar y se concluye devolviendo su relevancia normalizada entre 0 y 1 (ver Figura. 3.3).

En primer el sistema detecta si el documento cuya relevancia se desea conocer consiste en texto plano o un fichero (de tipo PDF, html o similar). En caso de ser un fichero, este es parseado, limpiado y partido en frases y párrafos. Estas tareas se llevan a cabo en los pasos *Parsear a texto crudo* y *Limpiar y tokenizar frases*, respectivamente.

Una vez se tienen mapeados los párrafos en listas de lemas de sustantivos, se consulta el lexicón por cada lista, acumulando el valor de relevancia de cada lema. En caso de que una frase no contenga ningún lema presente en el lexicón, la lista de lemas se tranforma en los vectores que entran a la red neuronal, que devuelve la predicción de relevancia. Estas tareas son llevadas a cabo por el módulo *Calculador de relevancia*. Una vez recorridas todas las listas de lemas, la relevancia viene dada por la Ecuación:

$$combined\_rel_p = \theta \cdot rel\_lex_p + (1-\theta) \cdot \frac{1}{K} \sum_{k=1}^{K} rel\_neural(s_k)\,, \tag{3.5}$$

donde rel_lex_p es la relevancia media del artículo proporcionada por el lexicón, $\{s_k\}_{k=1}^{K}$ es

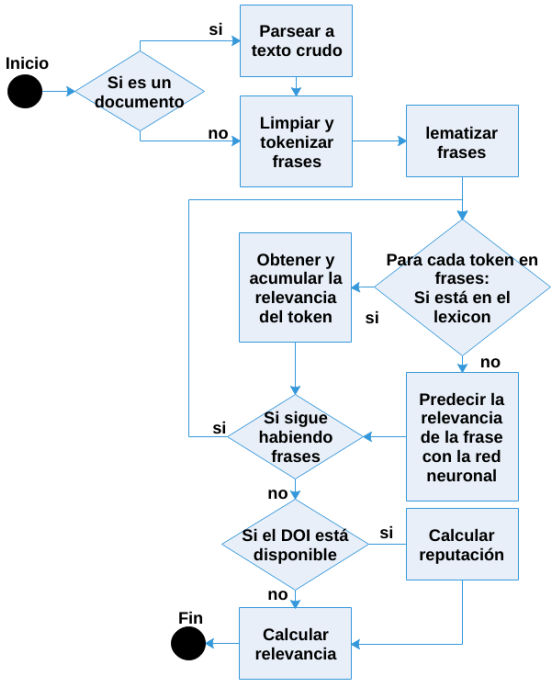


Figura 3.3: Flujo de trabajo de cálculo de relevancia de un artículo

el conjunto de frases cuya relevancia es desconocida y rel_neural es la relevancia predicha por la red neuronal para cada s_k. El parámetro $\theta \in (0, 1)$ modula la importancia relativa de la red neuronal.

En última instancia, en caso de que la entrada al sistema sea un documento y no texto plano, se comprueban los metadados del mismo para tratar de encontrar el DOI. En caso de estar disponible, el módulo *Calculador de relevancia*, calcula su reputación y esta es utilizada para calcular la relevancia final del documento, la cuál viene dada por:

$$combined\_rel\_doi_p = \gamma \cdot combined\_rel_p + (1 - \gamma) \cdot rep_p \,, \tag{3.6}$$

donde $combined\_rel_p$ es la relevancia combinada del lexicón y la red neuronal para el artículo y rep_p es su reputación. El parámetro $\gamma \in (0, 1)$ modula, su importancia relativa.

En caso de que el DOI no esté disponible, se devuelve el resultado de la Ecuación 3.5 como medida de relevancia final.

Capítulo 4

Experimentos y resultados

En este capítulo se presentan los experimentos realizados. En primer lugar se presenta un diagrama de Gantt [43] con la planificación temporal del desarrollo, después se detallan los diferentes lexicones creados, después las redes neuronales y, por último, los resultados evaluando la relevancia de diversos conjuntos de documentos.

4.1. Planificación temporal

En la Figura 4.1 se puede ver un diagrama de Gantt que ilustra el tiempo empleado en cada una de las etapas del desarrollo. La primera parte, se corresponde con el estudio e investigación del estado del arte, el cuál ha comprendido desde el inicio hasta el final del proyecto. Después, la obtención, pre-procesado y almacenaje de los datos, lo cuál incluye parseo, limpieza, cálculo de resúmenes y reputaciones. En la fase de creación de lexicones se construyen 4 diferentes por año. De forma análoga, se crean 4 redes neuronales por año. Finalmente, se realizan diversos experimentos para evaluar el rendimiento del sistema completo.

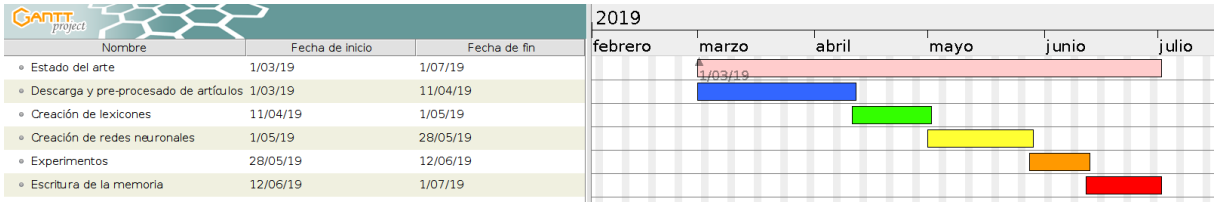


Figura 4.1: Diagrama de Gantt del desarrollo

Parámetro	Saturación
Número de artículos	196
citationVelocity	105
influentialCitationCount	208
Seniority	34
Citas del artículo	10

Tabla 4.1: Valores de saturación

4.2. Descarga y preprocesado de artículos

Siguiendo la metodología expuesta en la Sección 3.2.1, se procede a la descarga, limpieza y almacenaje de dos millones de artículos de Pubmed Central. De los mas de dos millones de artículos descargados, todos son parseados y limpiados, pero sólo se calcula el resumen y la reputación de cuatro conjuntos de 15000 artículos, que se corresponden con los años 2015, 2016, 2017 y 2018, para cada uno de los cuáles se construye un lexicón único. Se elige el número 15000 porque en pruebas anteriores se ha comprobado que es el número máximo de filas de la matriz de términos por documentos que cabe en la memoria del equipo de pruebas (16GB de RAM). Se fijan los parámetros de modulación para cada una de las ecuaciones del cálculo de reputaciones: En primer lugar, para la ecuación 3.1 se fija α a $0{,}5$ dando el mismo peso al número de citas que a la reputación de los coautores, ya que un alto volumen de citas es indicativo de un alto nivel de reputación, no obstante, si se le diera aún mas peso, se penalizarían demasiado los artículos mas recientes. En la ecuación 3.2 las omegas se fijan de la siguiente manera: $\omega_1 = 0.2$, $\omega_2 = 0.1$, $\omega_3 = 0.3$, $\omega_4 = 0.4$

Se decide dar más peso a los valores de veteranía y número de artículos publicados por el autor, ya que los otros dos parámetros, son valores que, aunque tienen relevancia, cuentan con un mayor nivel de subjetividad y son calculados con algoritmos propios de Semantic Scholar.

Una vez fijados los parámetros de modulación, se fijan los valores de saturación a partir del percentil 90 de cada parámetro tomando una muestra de 5000 artículos aleatorios, ver Tabla 4.1.

Al fijar los valores de saturación y normalizar entre 0 1 la reputación de todos los artículos está también entre 0 y 1.

4.3. Creación de lexicones por año.

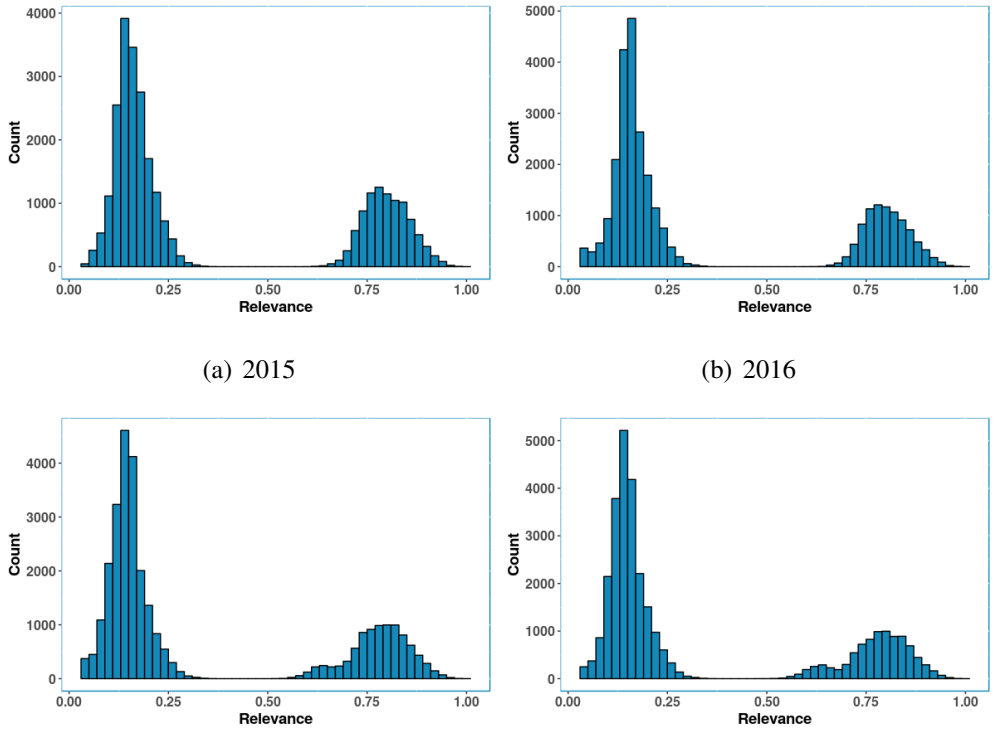
Siguiendo la metodología propuesta en la sección 3.2, se fijan los parámetros para las Ecuaciones 3.3 y 3.4:

Para la Ecuación 3.3, se toma $\beta = 0{,}5$ para dar la misma importancia a la reputación que a la componente léxica en la medida de relevancia. En la Ecuación 3.3 se toma $\rho = 0{,}8$, lo cuál da un mayor peso a la relevancia actual del término, teniendo en cuenta algo de la relevancia del año anterior. Cabe destacar, que para aquellos términos que dejen de estar presentes de un año para otro, su relevancia actual es de $(1 - \rho)0{,}2$. Que un término desaparezca por completo de un año para otro es indicativo de ser poco relevante y aunque se mantiene en el lexicón del año actual, su relevancia es penalizada. Por otra parte, a aquellos términos nuevos para cada lexicón (no aparecen en años anteriores) se les da toda la relevancia correspondiente, al ser considerados términos de relevancia emergente. Por motivos obvios, el lexicón correspondiente al año 2015 no cuenta con factor de memoria de los años anteriores, ya que este actúa como semilla.

Por otra parte, se ha tenido en consideración el uso de un diccionario médico abierto para darle más peso a los términos estrictamente médicos y así penalizar la relevancia de términos de carácter académico, típicos en el lenguaje de todos los artículos de investigación. En concreto, los mejores resultados se han obtenido dándole un 80 por ciento del peso a aquellos términos del diccionario médico y el resto a los que no están. Esto provoca una separación en la poblaciones claramente visible en los histogramas de las relevancias de los lexicones (ver Figura 4.2). Conforme se aumenta de año, puede verse un ensanchamiento de la población de los términos que se encuentran en el diccionario, fruto de la adición de los términos de años anteriores.

4.4. Creación de las redes neuronales por año.

Para crear el conjunto de entrenamiento, se cuenta con mas de 100000 artículos de cada año no utilizados para la creación de los lexicones. Siguiendo el procedimiento descrito en la Sección 3.3, se crea un conjunto con 320000 frases etiquetadas para entrenamiento y 80000 para test para los 4 lexicones. Se etiquetan como relevantes aquellas frases que contienen 4 o más sustantivos que tengan relevancia $0{,}7$ o mayor. Se utiliza este valor de corte porque es a partir del cuál se encuentra la mayoría de los términos mas relevantes de la población pertenecien-



(a) 2015 (b) 2016

(c) 2017 (d) 2018

Figura 4.2: Histograma de relevancias para cada lexicón.

te al diccionario médico. Siguiendo este criterio, se realiza un análisis exploratorio con 1000 artículos del año 2015 no utilizados para la creación del lexicón, obteniendo los siguientes resultados: 112808 frases totales analizadas, de las cuáles 96682 son etiquetadas como relevantes y 16126 como no relevantes. El alto porcentaje de frases relevantes indica que el lexicón tiene una riqueza aceptable, implicando una menor dependenencia de la red neuronal.

Se ha probado con dos conjuntos pre-entrenados de *Word2vec* [44] uno de Google y otro de Glove [19]. En el caso del Modelo de Google, el cuál cuenta con un diccionario de 3000000 palabras, se ha analizado el conjunto de completo de entrenamiento y test del año 2015, en el cuál hay un total de 5026862 sustantivos, de los cuáles 4809398 se encuentran en el diccionario de Word2Vec. Cuenta con 97104 sustantivos únicos de los cuáles 34203 se tienen codificados. Dado el elevado volumen de palabras codificadas en la capa de *embeddings* utilizando este modelo, el modelo final de predicción de relevancia ocupa 3,6GB en disco.

Por otra parte, el modelo de Glove, el cuál cuenta con 400000 palabras codificadas. Del total del mismo conjunto del 2015, hay 4826158 en el diccionario, siendo superior al modelo de Google a pesar de contar este último con más palabras codificadas. En cuanto a sustantivos únicos,

Año	Acierto en conjunto de test
2015	0,964
2016	0,962
2017	0,959
2018	0,959

Tabla 4.2: Resultados de test de las redes neuronales

se encuentran un total de 34203, de nuevo, superior al modelo de Google, que además da lugar a modelos entrenados mucho mas ligeros, de 80MB en disco, por tanto se elige trabajar con el modelo de Glove. En la Tabla 4.2 pueden verse los porcentajes de acierto con los conjuntos de test para cada año.

4.5. Experimentos

Se han analizado un total de 8000 artículos con el sistema completo, 2000 para cada año, prediciendo los artículos de cada año con el lexicón y la red del año anterior, a excepción de los artículos del año 2015, para los cuáles se ha utilizado la red y el lexicón del mismo año. Se ha calculado la relevancia con los siguientes parámetros: En la Ecuación 3.5, se toma θ = 0,8 dando la mayor parte del peso a la parte del lexicón, ya que la red funciona como apoyo al sistema. Por otra parte, en la Ecuación 3.6 se toma γ = 0,6 dando la mayor parte del peso al resultado de la Ecuación 3.5. En la Figura 4.3 se puede ver una representación de variables dos a dos de los resultados de predicción del año 2016 con la red y el lexicón del 2015.

Puede observarse la poca influencia que tienen los valores de reputación de la red neuronal sobre la relevancia total del sistema incluyendo la reputación, lo cuál es indicativo de que las frases cuya relevancia se desconoce tienen poca relevancia (buen funcionamiento de la red) que a su vez es consecuencia del buen funcionamiento del lexicón, ya que cerca del 90 porciento de las frases tienen relevancia conocida.

Aunque se han analizado documentos para los siguientes años, de las gráficas se extraen las mismas conclusiones, ya que tienen distribuciones muy parecidas.

Por otra parte, en la Figura 4.4 puede verse una tabla con los 15 resultados de mayor relevancia del caso anterior. En en todos los casos, los valores de reputación son elevados, lo que

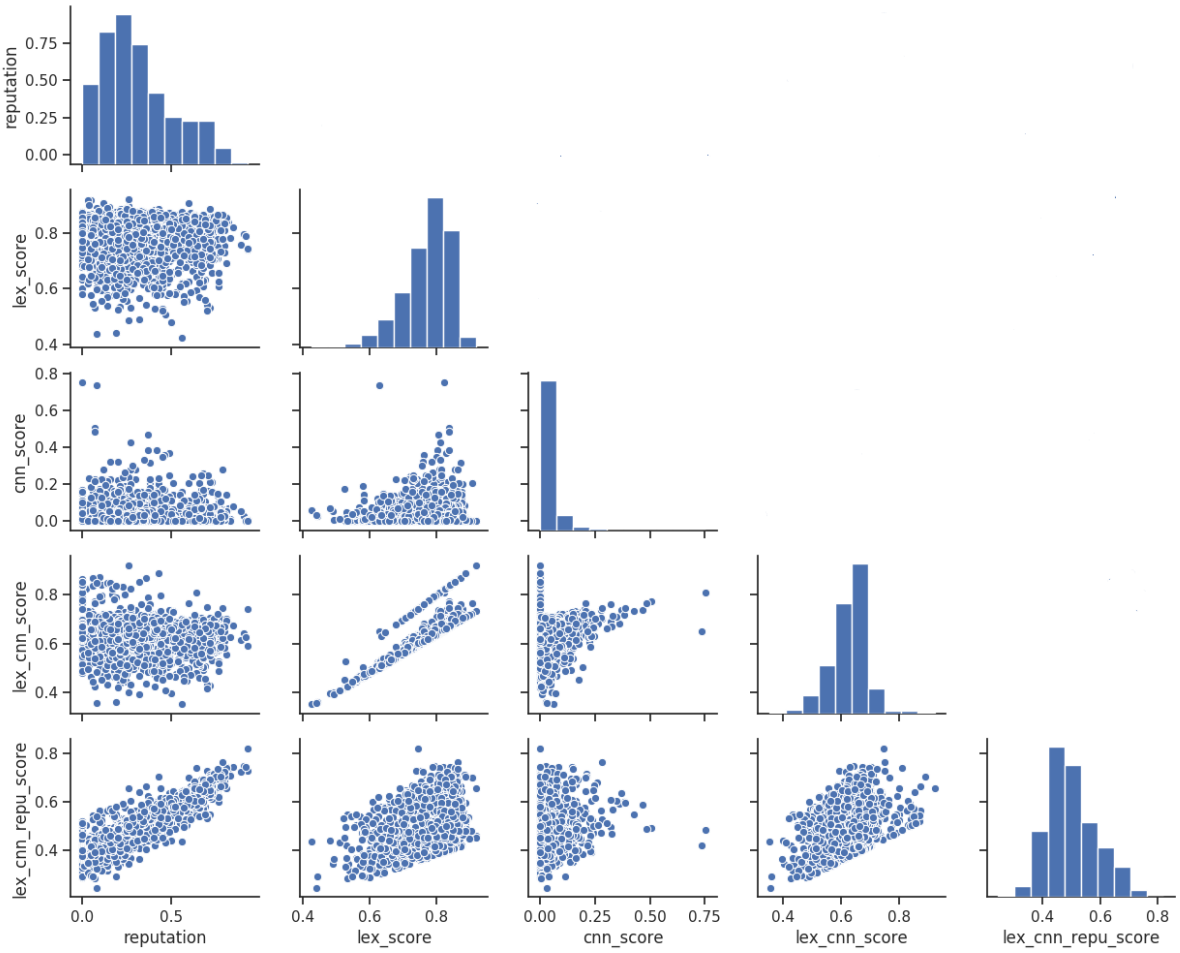


Figura 4.3: Resultados 2016

reputation	lex_cnn_repu_score	lex_score	cnn_score	total_sentences	lex_cnn_score
0.93	0.819	0.744	0	83	0.744
0.79	0.764	0.863	0.282	215	0.747
0.91	0.749	0.795	0.027	104	0.642
0.92	0.746	0.787	0	92	0.629
0.81	0.745	0.841	0.144	278	0.702
0.85	0.744	0.821	0.082	218	0.673
0.64	0.741	0.809	0	114	0.809
0.81	0.74	0.854	0.052	272	0.694
0.82	0.738	0.837	0.072	181	0.684
0.76	0.737	0.864	0.15	177	0.721
0.78	0.733	0.862	0.058	101	0.701
0.7	0.73	0.75	0	91	0.75
0.93	0.728	0.739	0.008	212	0.593
0.89	0.726	0.758	0.053	134	0.617
0.8	0.723	0.833	0.03	275	0.672

Figura 4.4: artículos analizados de mayor relevancia

está también acorde con lo elevados que son los valores de la puntuación del lexicón exclusivamente. En todos los casos el valor de relevancia proporcionado por la red neuronal es muy baja o oncluso 0, reforzando el carácter de apoyo de las red neuronal. En todos los casos se trata de artículos de gran impacto y número de citas, tendencia que se mantiene en la gran mayoría de artículos de alta reputación. En aquellos casos en los se tiene un elevado valor en la relevancia pero bajo número de citas o reputación en general, se debe a un muy elevado valor de la componente exclusivamente léxica, no obstante, es un comportamiento raramente dado.

Capítulo 5

Conclusiones

En cuanto a la consecución de los objetivos específicos, se puede decir que se han conseguido con éxito. Se ha obtenido un volumen de artículos más que suficiente para los experimentos, el desarrollo y trabajos futuros. Se han creado los lexicones y las redes neuronales, cuyo rendimiento ha sido evaluado por separado y en conjunto con resultados satisfactorios. El resultado final ofrece un marco de trabajo completo de fácil uso, bajo coste computacional (una vez entrenado) y de fácil reproducción si se desea aplicar a otro dominio.

Quedan, no obstante, para trabajos futuros la detección de correferencias de términos, lo que daría una componente frecuencial de éstos más precisa. También quedaría pendiente el entrenamiento y aplicación de embeddings creados con el propio corpus de entrenamiento, ya que como se ha visto en capítulo anterior, el conjunto de entrenamiento tiene un gran número de palabras para las que no se cuenta con represetacíon vectorial. Otra posible línea de mejora, consiste en sustituir la utilización de uni-gramas por la de n-gramas con conceptos del dominio de aplicación, en el caso del dominio médico, se podría utilizar una ontología como UMLS [45]. Esto plantearía, no obstante, otros desafíos, como la detección de conceptos y la desambiguación de los mismos dentro de los textos. Por último, aunque los lexicones han demostrado tener una alta cantidad de los términos más relevantes, su tamaño está limitado por el equipo de pruebas y podría aumentarse si distribuye el cálculo de los mismos o se aumenta la memoria del computador encargado de construirlos, con la consecuente disminución de dependencia de la red neuronal.

Bibliografía

[1] Dan Jurafsky and James H Martin. *Speech and language processing*, volume 3. Pearson London, 2014.

[2] Audun Jøsang, Roslan Ismail, and Colin Boyd. A survey of trust and reputation systems for online service provision. *Decision support systems*, 43(2):618–644, 2007.

[3] Isaac Martín de Diego, Alberto Fernández-Isabel, Felipe Ortega, and Javier M Moguerza. A visual framework for dynamic emotional web analysis. *Knowledge-Based Systems*, 145:264–273, 2018.

[4] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.

[5] Juan Ramos et al. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, pages 133–142, 2003.

[6] Yili Wang and Heeyong Youn. Feature weighting based on inter-category and intra-category strength for twitter sentiment analysis. *Applied Sciences*, 9(1):92, 2019.

[7] Franca Debole and Fabrizio Sebastiani. Supervised term weighting for automated text categorization. In *Text mining and its applications*, pages 81–97. Springer, 2004.

[8] Man Lan, Chew Lim Tan, Jian Su, and Yue Lu. Supervised and traditional term weighting methods for automatic text categorization. *IEEE transactions on pattern analysis and machine intelligence*, 31(4):721–735, 2008.

[9] Kewen Chen, Zuping Zhang, Jun Long, and Hao Zhang. Turning from tf-idf to tf-igm for term weighting in text classification. *Expert Systems with Applications*, 66:245–260, 2016.

[10] David B Bracewell. Semi-automatic creation of an emotion dictionary using wordnet and its evaluation. In *2008 IEEE Conference on Cybernetics and Intelligent Systems*, pages 1385–1389. IEEE, 2008.

[11] Bo Pang, Lillian Lee, et al. Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval*, 2(1–2):1–135, 2008.

[12] Andrea Esuli and Fabrizio Sebastiani. Sentiwordnet: a high-coverage lexical resource for opinion mining. *Evaluation*, 17:1–26, 2007.

[13] Sixing Wu, Fangzhao Wu, Yue Chang, Chuhan Wu, and Yongfeng Huang. Automatic construction of target-specific sentiment lexicon. *Expert Systems with Applications*, 116:285–298, 2019.

[14] Dong Deng, Liping Jing, Jian Yu, Shaolong Sun, and Michael K Ng. Sentiment lexicon construction with hierarchical supervision topic model. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(4):704–718, 2019.

[15] Erik Cambria, Soujanya Poria, Devamanyu Hazarika, and Kenneth Kwok. Senticnet 5: Discovering conceptual primitives for sentiment analysis by means of context embeddings. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[16] Alberto Fernández-Isabel, Juan Carlos Prieto, Felipe Ortega, Isaac Martín de Diego, Javier M Moguerza, José Mena, Sara Galindo, and Liana Napalkova. A unified knowledge compiler to provide support the scientific community. *Knowledge-Based Systems*, 161:157–171, 2018.

[17] Filipa Peleja, João Santos, and João Magalhães. Reputation analysis with a ranked sentiment-lexicon. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 1207–1210. ACM, 2014.

[18] Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1555–1565, 2014.

[19] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[20] Soujanya Poria, Erik Cambria, and Alexander Gelbukh. Deep convolutional neural network textual features and multiple kernel learning for utterance-level multimodal sentiment analysis. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 2539–2544, 2015.

[21] Francois Chollet. *Deep Learning mit Python und Keras: Das Praxis-Handbuch vom Entwickler der Keras-Bibliothek*. MITP-Verlags GmbH & Co. KG, 2018.

[22] James Pustejovsky. The generative lexicon. *Computational linguistics*, 17(4):409–441, 1991.

[23] Panos Vassiliadis. A survey of extract–transform–load technology. *International Journal of Data Warehousing and Mining (IJDWM)*, 5(3):1–27, 2009.

[24] Jing Han, E Haihong, Guan Le, and Jian Du. Survey on nosql database. In *2011 6th international conference on pervasive computing and applications*, pages 363–366. IEEE, 2011.

[25] Clinton Gormley and Zachary Tong. *Elasticsearch: The definitive guide: A distributed real-time search and analytics engine*. .ºReilly Media, Inc.", 2015.

[26] Yuvraj Gupta. *Kibana Essentials*. Packt Publishing Ltd, 2015.

[27] Ryan Mitchell. *Web Scraping with Python: Collecting More Data from the Modern Web*. .ºReilly Media, Inc.", 2018.

[28] Tim Bray, Jean Paoli, C Michael Sperberg-McQueen, Eve Maler, and Franois Yergeau. Extensible markup language (xml) 1.0, 2000.

[29] Douglas Crockford. The application/json media type for javascript object notation (json). Technical report, 2006.

[30] Rada Mihalcea and Paul Tarau. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, 2004.

[31] Federico Barrios, Federico López, Luis Argerich, and Rosa Wachenchauzer. Variations of the similarity function of textrank for automated summarization. *CoRR*, abs/1602.03606, 2016.

[32] Marco Valenzuela, Vu Ha, and Oren Etzioni. Identifying meaningful citations. In *Workshops at the twenty-ninth AAAI conference on artificial intelligence*, 2015.

[33] Allen Institute for Artificial Intelligence and Semantic Scholar. Semantic Scholar API. `https://api.semanticscholar.org/`, 2018. [Online: accedido 20-Dic-2018].

[34] Norman Paskin. Digital object identifier (doi®) system. *Encyclopedia of library and information sciences*, 3:1586–1592, 2010.

[35] Aravind Chandramouli. Domain-specific stopword removal from unstructured computer text using a neural network, August 9 2018. US Patent App. 15/426,958.

[36] Sonal Gupta. *Distantly Supervised Information Extraction Using Bootstrapped Patterns*. PhD thesis, Stanford University, 2015.

[37] Clement Levallois. Lists of academic stopwords. `https://github.com/seinecle/Stopwords`, 2016. [Online: accessed 20-Jan-2019].

[38] Murugan Anandarajan, Chelsey Hill, and Thomas Nolan. Term-document representation. In *Practical Text Analytics*, pages 61–73. Springer, 2019.

[39] Merriam Webster. Merriam-webster medical dictionary. *Available from: Merriam-Webster and https://en. wikipedia. org/wiki/Iatrogenesis*, 2017.

[40] Lee Averell and Andrew Heathcote. The form of the forgetting curve and the fate of memories. *Journal of Mathematical Psychology*, 55(1):25–35, 2011.

[41] Krishna Bhavsar, Naresh Kumar, and Pratap Dangeti. *Natural Language Processing with Python Cookbook: Over 60 recipes to implement text analytics solutions using deep learning principles*. Packt Publishing Ltd, 2017.

[42] Dan O'Sullivan. *Wikipedia: a new community of practice?* Routledge, 2016.

[43] James M Wilson. Gantt charts: A centenary appreciation. *European Journal of Operational Research*, 149(2):430–437, 2003.

[44] Yoav Goldberg and Omer Levy. word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*, 2014.

[45] Karin Verspoor. Towards a semantic lexicon for biological language processing. *International Journal of Genomics*, 6(1-2):61–66, 2005.