



MÁSTER EN INGENIERÍA EN SISTEMAS DE DECISIÓN

Curso Académico 2018/2019

Trabajo Fin de Máster

Marco de trabajo para evaluar la relevancia de los artículos en el dominio científico

Autor : Adrián Alonso Barriuso

Tutor : Dr. Alberto Fernández Isabel

*Dedicado a
mi familia, pareja, amigos y a todos los que me aguantan, en el buen sentido.*

Agradecimientos

Resumen

Summary

Índice general

1. Introducción	13
1.1. Contexto	13
1.1.1. Dominio de aplicación	13
1.2. Objetivos	13
1.2.1. Objetivo General	13
1.2.2. Objetivos específicos	13
1.3. Estructura de la memoria	13
2. Estado del arte	15
2.1. Algoritmos de reputación	15
2.2. Obtención de relevancias	15
3. Propuesta	17
3.1. Arquitectura general	17
3.2. Creación del lexicón de relevancias	18
3.2.1. Módulo ETL	18
3.2.2. Módulo de Gestión del conocimiento	20
3.2.3. Text filter	20
3.2.4. Text normalizer	20
3.2.5. Lexicon builder	21
3.3. Creación de la red neuronal	21
3.4. Estimación de relevancias de artículos	21
4. Experimentos y resultados	23
4.1. Planificación temporal	23

4.2. Experimentos	23
5. Conclusiones	25
Bibliografía	27

Índice de figuras

3.1. Arquitectura general	18
3.2. Arquitectura de generación de lexicones	19
3.3. Flujo de trabajo	22
4.1. Diagrama de Gantt del desarrollo	23

Capítulo 1

Introducción

1.1. Contexto

1.1.1. Dominio de aplicación

1.2. Objetivos

1.2.1. Objetivo General

1.2.2. Objetivos específicos

-
-

1.3. Estructura de la memoria

1. Estado del arte:
2. Propuesta:
3. Experimentos y resultados:
4. Conclusiones:

Capítulo 2

Estado del arte

2.1. Algoritmos de reputación

2.2. Obtención de relevancias

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: *Relevance estimator module*, que es el encargado de estimar las relevancias de los artículos de entrada, y sus correspondientes submódulos: *Text processor*, *Reputation calculator* y *Relevance calculator*. Se cuenta además, con dos fuentes de información precalculadas utilizadas por el submódulo *Relevance calculator*, a saber, *Relevance lexicon* que consiste en cuatro lexicones[1] de relevancias de términos médicos y *Neural network*, que consiste en cuatro modelos entrenados de redes neuronales convolucionales (CNN)[2] por sus siglas en inglés. Por otra parte, se cuenta con un módulo de visualización(*Visualization*), que se utiliza para ver la salida del sistema y para proporcionar la entrada (*Text*). Por último, se utiliza una fuente de información externa en tiempo real (*Web information resources*).

En primer lugar se entra en detalle en cómo se construye *Relevance lexicon*, después *Neural network* y, por último, *Relevance estimator module*, el cuál emplea todo lo anterior para calcular las relevancias de nuevos documentos.

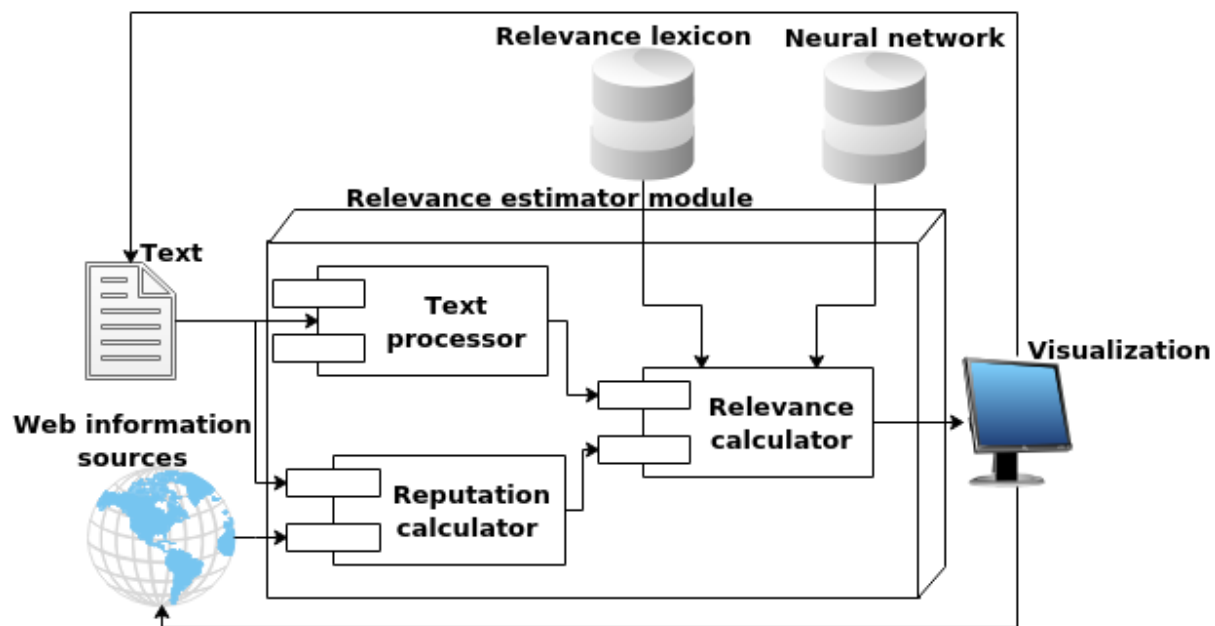


Figura 3.1: Arquitectura general

3.2. Creación del lexicon de relevancias

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

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: Corpus processor y Reputation calculator. Corpus processor obtiene artículos de Pubmed Central¹ utilizando técnicas de web scrapping [?]. Estos artículos en formato XML [?], son parseados y almacenados en formato JSON [?] en Document

¹<https://www.ncbi.nlm.nih.gov/pmc/>

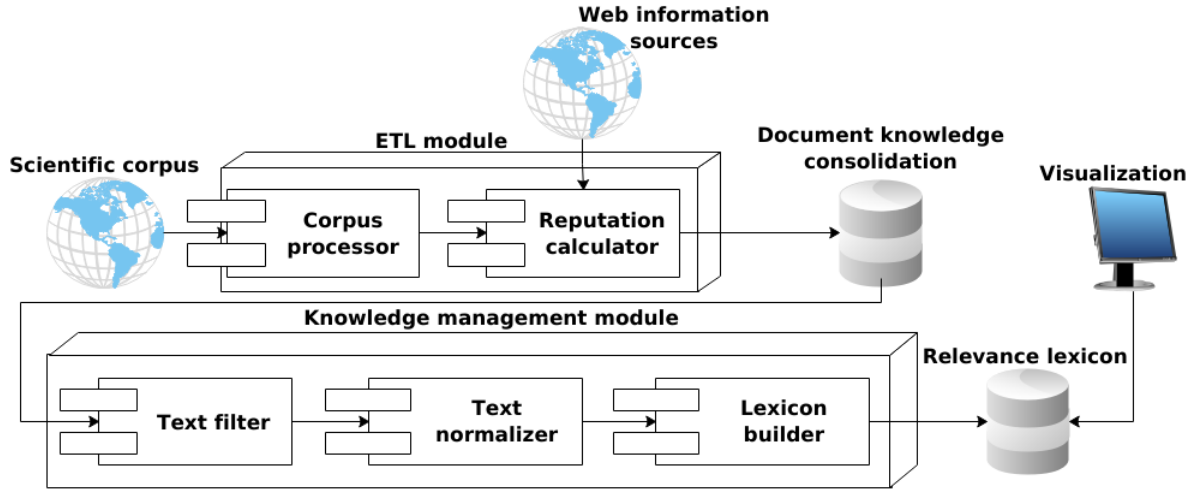


Figura 3.2: Arquitectura de generación de lexicones

knowledge consolidation 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 [?] utilizando una popular implementación en Python [?]. 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 por ciento del tamaño original de los artículos y se almacena como un nuevo atributo del documento en la base de datos.

Una vez preprocesados y almacenados los artículos, Reputation calculator calcula la reputación de los mismos para que esta sea añadida como nuevo atributo y ser utilizada posteriormente por el módulo Knowledge management. El algoritmo de reputación empleado es una adaptación del introducido en el artículo [?]. La reputación a priori de un artículo viene dada por:

$$rep_p = \alpha \cdot rep_authors_p + (1 - \alpha) \cdot citations_p, \quad (3.1)$$

donde $rep_authors_p$ es la reputación media de 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:

$$rep_i = \omega_1 \cdot inf_citation_count + \omega_2 \cdot citation_velocity + \omega_3 \cdot seniority + \omega_4 \cdot papers, \quad (3.2)$$

donde $\sum_{i=1}^4 \omega_i = 1$. El parámetro *inf_citation_count* representa el número de citas altamente influyentes del autor. [?]. *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[?]. Una vez que el módulo Reputation Calculator ha calculado la reputación del artículo, esta se añade a *Document knowledge consolidation*, como un atributo nuevo.

3.2.2. Módulo de Gestión del conocimiento

Este módulo consulta *Document knowledge consolidation* y sirviéndose de 3 submódulos, construye finalmente el lexicón. Estos 3 submódulos son Text filter, Text normalizer y Lexicon builder.

3.2.3. Text filter

Este módulo analiza los resúmenes extraídos por el módulo ETL por cada documento, aplicando técnicas de procesamiento de lenguaje natural (NLP) [?] extrayendo los sustantivos y eliminando palabras vacías [?], tanto genéricas como de dominio médico[?] y académico[?]. 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. Text normalizer

Este submódulo construye una matriz de términos por documentos [?] a partir de la cuál construye la matriz TF-IDF [?]. 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:

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

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. Lexicon builder

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 [?].

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 [?] 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:

$$rel_lex_t(y) = \rho \cdot rel_lex_t + (1 - \rho) \cdot rel_lex_t(y - 1), \quad (3.4)$$

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 desarrollado una Red Neuronal Convolutiva (CNN, por sus siglas en inglés)[2] para servir de apoyo al sistema. El propósito de la red es la de predecir la relevancia de aquellas frases que no contengan ninguna palabra presente en el lexicón. En la Tabla ?? se puede ver la configuración de la misma, la cuál es una versión optimizada de la aproximación introducida en [?]

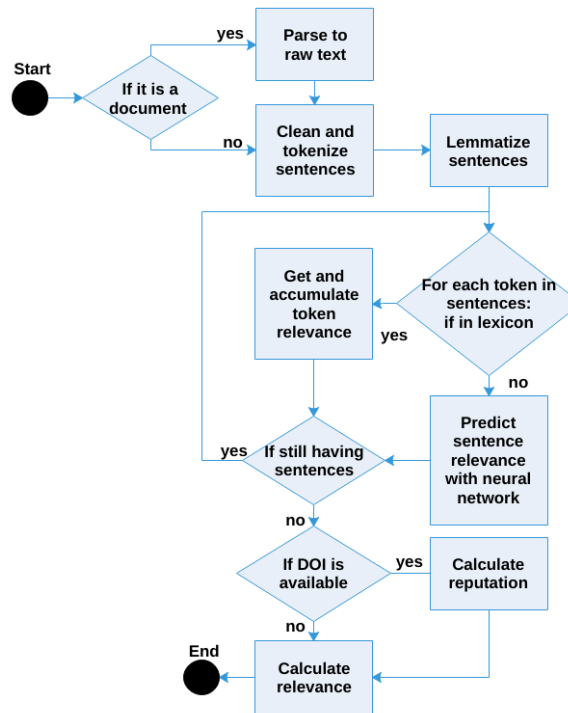


Figura 3.3: Flujo de trabajo

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: Layers of the Convolutional Neural Network.

3.4. Estimación de relevancias de artículos

Capítulo 4

Experimentos y resultados

4.1. Planificación temporal

En la Figura 4.1 se puede ver un diagrama de Gantt[3] que refleja el tiempo empleado en cada una de las fases del proyecto.

4.2. Experimentos

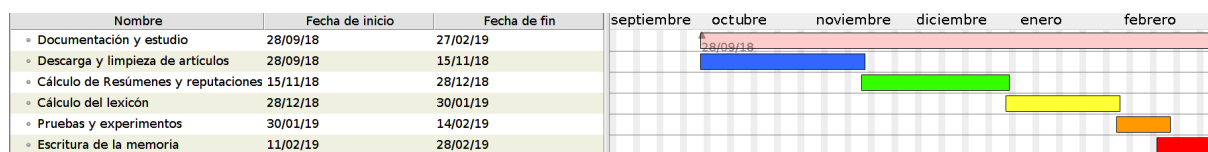


Figura 4.1: Diagrama de Gantt del desarrollo

Capítulo 5

Conclusiones

Bibliografía

- [1] James Pustejovsky. The generative lexicon. *Computational linguistics*, 17(4):409–441, 1991.
- [2] 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.
- [3] James M Wilson. Gantt charts: A centenary appreciation. *European Journal of Operational Research*, 149(2):430–437, 2003.