Proyecto de Sistemas de Recuperación de Información

Enrique Martínez González, Carmen Irene Cabrera Rodríguez, and Osmany Pérez Rodríguez

C512

Abstract. La localización de materiales de naturaleza no estructurada para satisfacer una necesidad de información en una larga colección almacenada es una actividad cada día más importante. Con el fin de lograr este objetivo se ha desarrollado un sistema de recuperación de información basándose en la utilización de los modelos vectorial y booleano capaces de encontrar documentos interrelacionados con una consulta planteada mediante una interfaz visual web. Se ha desarrollado de igual forma un mecanismo para poder expandir la consulta permitiendo apreciar posibles modificaciones de esta con el objetivo de tener una mayor efectividad en la búsqueda de información. Finalmente se aprecian las comparativas de los modelos implementados probados en dos sets de datos distintos.

Keywords: Sistema de Recuperación de Información, modelo vectorial, modelo booleano, consulta expandida

Table of Contents

1 Introducción

En la actualidad, dado el auge que ha tenido el Internet, existe una enorme disponibilidad de información en línea y documentos, que resulta abrumadora para los usuarios. De ahí que sea interés de muchos investigadores desarrollar una herramienta automática que permita la obtención de información relevante a una consulta determinada. Esta consulta debe ser resuelta rápidamente obteniendo documentos útiles acerca del tema solicitado.

Esta resulta una tarea no trivial y difícil de llevar a cabo; pues, mientras las personas pueden leer un documento para comprenderlo y ver acerca de qué contenido trata y luego decidir si es relevante a la consulta; la computadora carece de conocimiento humano y de la capacidad del lenguaje.

Las primeras incursiones en esta área, datan de mitades del siglo XX, comenzando con estudios del set de datos la colección Cranfield, utilizando un gran número de técnicas distintas cuyo rendimiento fue bueno. Desde entonces, muchos trabajos han sido publicados para abordar este problema, pudiendo observar un aumento significativo de las investigaciones sobre el tema a partir de 1992. Son numerosos los enfoques que se han llevado a cabo y que son aplicados ampliamente en varios dominios. El presente trabajo tiene como propósito tratar los enfoques planteados en el modelo booleano y el modelo vectorial para la obtención de documentos relevantes, además de plantear un algoritmo de expansión de consulta.

2 Modelo de Recuperación de Información

Se puede definir un modelo de recuperación de información como una cuádrupla en donde se tiene un conjunto de representaciones lógicas de los documento de la colección al que llamaremos D, un conjunto de las representaciones lógicas de la consulta realizada por el usuario al que llamaremos Q, para modelar las representaciones de los documentos y las consultas, y sus relaciones se utiliza un framework que llamaremos F y finalmente una función que es capaz de ordenar de alguna manera los documentos pertencientes a D relevantes ante la consulta Q, esta función posee el nombre de ranking a la que llamaremos R.

2.1 Representación de documentos

Para la realización de este proyecto se han utilizado 3 conjuntos de documentos, dos de ellos para la evaluación de la calidad de los modelos, y el tercero para la construcción de una herramienta visual que permite al usuario realizar consultas para probar de forma interactiva la herramienta desarrollada. Los dos conjuntos de documentos utilizados para la evaluación poseen el nombre de **Cranfield** y **Vaswani**, estos poseen 1400 y 11000 documentos respectivamente, ambos con temática de resúmenes científicos.

La representación de estos documentos es obtenida mediante un proceso de construcción de índices y de índices inversos. En el índice se almacena la información de los términos que aparecen en cada documento, y la frecuencia con

la que aparece, además de utilizar este mismo cómputo para calcular múltiples valores que serán posteriormente utilizados en el modelo vectorial, este cómputo adelantado permitirá la obtención de documentos relevantes a consultas de manera más rápidas, los detalles se explicarán posteriormente en el artículo. En el índice inverso se almacena por cada término perteneciente al conjunto, en qué documentos este aparece. Resaltar que los términos utilizados en estos dos cómputos no son todas las palabras que aparecen en los documentos, estas palabras se filtran utilizando la biblioteca **Spacy** para tokenizar el documento y después eliminar aquellos tokens que no son de importancia para el procesamiento: signos de puntuación, de monedas, dígitos, espacios, stopwords, y luego transformadas a su forma base utilizando la funcionalidad **lemmatise** que posee esta biblioteca.

Con estas representaciones se está comprimiendo un conjunto de documentos en dos diccionarios de **Python** que nos permite la rápida inserción y extracción de información acerca de documentos y términos presentes en el sistema.

2.2 Representación de consultas

Los dos conjuntos de documentos planteados anteriormente que fueron utilizados para la evaluación de la calidad de los modelos implementados poseen un conjunto de consultas y para cada una de estas poseen los documentos que son relevantes y su valor de relevancia está determinado por un valor que va de -1 hasta 4, siendo -1 el valor para un documento que no posee mucha relevancia con la consulta y siendo 4 el valor que poseen los documentos con más valor para la necesidad del usuario. **Cranfield** posee 225 consultas con un total de 1800 documentos relevantes en total mientras que **Vaswani** tiene 93 consultas para un total de 2100 documentos relevantes. Resaltar que estas consultas no están orientadas al modelo booleano ya que no están expresadas mediante el uso de una expresión booleana sobre los términos utilizando las operaciones clásicas como, **OR**, **AND** y **NOT**. Para la comparación de modelos se modificó estos sets de consultas en la evaluación del modelo booleano, susituyendo los espacios entre términos por operadores **OR**.

Modelo Booleano

Para la representación de las consultas en el modelo booleano se ha creado un **Lexer** y un **Parser** utilizando la biblioteca **Ply** permitiendo así implementar una gramática para poder leer consultas que son expresiones booleanas con las operaciones definidas anteriormente, para el agrupamiento dentro de la expresión, cumpliendo con la propiedad asociativa y distributiva que poseen las expresiones booleanas se han empleado los signos de corchetes []. La gramática empleada posee las siguientes reglas:

```
Expr : Expr AND Expr
Expr : Expr OR Expr
Expr : [ Expr ]
```

```
Expr : NOT Expr
Expr : Term
Term : word

word = r"[a-zA-Z0-9_.,?!'/\(\)-]+"
```

El lexer, divide en tokens la consulta y desecha aquellos caracteres que no se consideran necesarios. Al mismo tiempo apoyándose en la biblioteca **Spacy**, se le aplica la función **lemmatise** a cada término, de la misma forma que se le realizó a los documentos, para llevar las palabras a su forma original. El resultado del **Parser** es un **AST** modificado, que posee toda la información valiosa de la consulta.

Modelo Vectorial

En el modelo vectorial lo primero que se realiza es la tokenización de los términos de la misma forma que se realizó con los documentos en un inicio, esto se hace con el objetivo de poder utilizar la función **lemmatise** de la biblioteca **Spacy**. Luego se crea un índice para la consulta, en donde se guarda por cada término la frecuencia con la que este aparece en la interrogante del usuario. Además se aprovecha este momento para realizar cómputo sobre los términos que será utilizado próximamente en la recuperación de documentos del modelo.

2.3 Framework

Para la representación de los documentos se utilizaron los diccionarios de **Python**, estos permiten la representación de los términos como un conjunto, y se puede almacenar información como su frecuencia. Además permite la rápida recuperación de estos.

Para la representación de las consultas se emplearon técnicas distintas dependiendo del modelo, en caso del modelo booleano se utilizó un árbol de sintaxis abstracta representado mediante diccionarios de **Python**, en los que se almacenaban los hijos de cada nodo y la operación que representaba. En caso del modelo vectorial la consulta se representa mediante un diccionario al igual que los documentos en los que se guarda la frecuencia de cada término dentro de la consulta y otras informaciones que serán explicadas posteriormente.

2.4 Ranking

El cómputo que se hizo en la etapa del análisis de los documentos y la consulta es de vital importancia para el rápido desarrollo del cálculo de la interrelación entre estos. En el caso de los documentos se calcula para cada térmio el valor de la frecuencia de un término en el documento, y además se calcula la frecuencia normalizada del término i en el documento j que esta es:

$$tf_{i,j} = \frac{freq_{i,j}}{max_l freq_{l,j}} \tag{1}$$

Como se puede ver, ninguno de los factores en la definición está fuera del conjunto de documentos analizados, por lo que este cálculo se puede desarrollar en esta fase de precómputo. Igualmente sucede con la frecuencia de ocurrencia de un término dentro de todos los documentos de la colección (idf). Con estos dos valores mencionados, tf e idf, podemos calcular el peso del término i en el documento j ya que este es:

$$w_{i,j} = tf_{i,j} * idf_i \tag{2}$$

Este peso será utilizado en el modelo vectorial. Igualmente se puede realizar cómputo sobre la consulta en cuanto es recibida, calculando el valor del peso del término i en la consulta q con esta ecuación:

$$w_{i,q} = (a + (1-a) * \frac{freq_{i,q}}{max_l freq_{l,q}}) * log\frac{N}{n_i} \tag{3}$$

N es el total de documentos en la colección y a es el término de suavizado, en la evaluación obtenida se utilizó a con un valor de 0.5.

Modelo Booleano

Recordemos que el resultado de la representación de la consulta en este modelo es un **AST**, por lo que sobre este árbol se realiza un recorrido de todos los nodos siguiendo un patrón **Visitor** creando el conjunto solución a la consulta utilizando para esto el índice invertido de términos creado. Este conjunto de solución a la consulta es el resultado de ir realizando interesecciones, uniones y diferencias de diferentes conjuntos de documentos durante el recorrido del árbol, cada operación de conjunto aplicada depende de la operación contenida en el nodo visitado, detallar que las hojas del árbol representan los términos de la consulta y son nodos que poseen los conjuntos de documentos en los que aparece este término. Finalmente el nodo raíz del **AST** contiene todos los documentos que cumplen con las condiciones presentadas en forma de consulta por el usuario. En caso de este modelo no existe superioridad de un documento relevante sobre otro relevante, sencillamente la relación de relevancia es binaria.

Modelo Vectorial

Es en este modelo donde vamos a hacer uso de todo el precálculo que se hizo sobre los documentos y la consulta, pues aquí la similitud de un documento y una interrogante está dada por el coseno del ángulo entre el vector documento y el vector consulta. Estos vectores están definidos sobre el espacio vectorial de n dimensiones en donde n es la cantidad de términos que se encuentran presentes en todo el sistema. Las componentes de cada vector son binarias, siendo 1 si el término del sistema representado por esa componente se encuentra en el

documento o consulta analizado, o siendo 0 en caso contrario. El coseno entre el vector representante del documento j y la consulta q se calcula de la forma:

$$sim(d_j, q) = \frac{\sum_{i=1}^{n} w_{i,j} * w_{i,q}}{\sqrt{\sum_{i=1}^{n} w_{i,j}^2} + \sqrt{\sum_{i=1}^{n} w_{i,q}^2}} \quad (4)$$

Se pudiese definir el peso total de un documento o consulta como la raíz de la sumatoria del cuadrado de los pesos de todos los términos en un documento o consulta, como vemos estos se pueden calcular en la etapa de preprocesamiento de los documentos y la consulta respectivamente, estos cálculos fueron también realizados en estas etapas para agilizar la velocidad de la recuperación de los documentos relevantes.

Luego queda hallar los términos en común entre la consulta y el documento para encontrar el numerador del cálculo del coseno y ya obtendría el valor de la similitud entre un documento y una consulta. Mientras mayor valor de similitud, mayor relevancia del documento ante la consulta.

3 Resultados

Para la evaluación de la calidad de los modelos se emplearon las consultas pertenecientes a cada juego de datos, recordemos que estos juegos de datos son Cranfield y Vaswani. Se midieron sobre los modelos los valores de precisión que es la cantidad de documentos relevantes recuperados sobre el total de documentos recuperados, el recobrado que es la cantidad de documentos relevantes recuperados sobre la cantidad total de documentos relevantes en el conjunto de datos. Además se evaluaron las medidas F con un valor de beta igual a 0.5 y F_1. Se utilizaron 3 cantidades de documentos recuperados distintas por cada evaluación, siendo estas 10, 50 y 100. A continuación se muestran las distintas medidas obtenidas para los dos modelos en los sets de datos tomando 100 documentos recuperados en cada consulta:

Table 1. Evaluaciones de variables con límite de 100 documentos recuperados en Cranfield

Modelo	Precisión	Recobrado	F	F_1
Vectorial	0.005	0.056	0.006	0.010
Booleano	0.004	0.060	0.004	0.007

Resaltar que la forma de obtención de estas fue promediar las medidas en cada una de las consultas en los sets probados. Como se puede apreciar en el conjunto de datos Vaswani se obtienen mejores resultados para los dos modelos, en el caso del modelo vectorial se obtiene una precisión del 9% y en caso del booleano 1%, recordar que las consultas en este set de datos no estaban preparadas para

Table 2. Evaluaciones de variables con límite de 100 documentos recuperados en Vaswani

Modelo	Precisión	Recobrado	F	F_1
Vectorial	0.091	0.439	0.104	0.138
Booleano	0.010	0.043	0.011	0.014

ser usadas en un modelo con operaciones booleanas. Pero sin dudas se puede apreciar como la aplicación de estos modelos no da buenos resultados.

4 Aplicación visual

Además de la evaluación de los modelos se ha desarrollado una aplicación visual utilizando la biblioteca **Streamlit**, para permitir que un usuario pueda experimentar con estos modelos libremente. En esta aplicación se utiliza como set de datos un conjunto de documentos acerca de atletismo. Al ejecutar la aplicación por primera vez se comenzará a ejecutar todo el precómputo sobre los documentos mencionado previamente en este informe. Los resultados se muestran en una tabla, en la que se puede apreciar la similitud de los documentos con la consulta en el modelo vectorial, y en el booleano solo se listan los documentos que fueron seleccionados por los algoritmos planteados en el informe, este listado no corresponde con ninguna similitud mayor de un documento sobre otro.

5 Consulta expandida

En la aplicación visual, específicamente en el modelo vectorial se puede apreciar que si en la consulta algún término no aparece en los documentos se hará una recomendación de consulta con otros términos que sí se encuentran dentro del set de datos. Esta recomendación es realizada mediante la distancia Levenshtein entre el término de la consulta y los términos en los documentos. Luego se ordenan estas distancias y se sustituye en la consulta el término con menor valor de este cómputo. Para esto se utilizó la biblioteca **Fuzzywuzzy**. Específicamente la fórmula para calcular la distancia entre los términos a y b que esta biblioteca utiliza es:

$$dist(a,b) = \frac{len(a) + len(b) - LevenshteinDistance(a,b)}{len(a) + len(b)} \tag{5}$$

En donde *len* es una función que retorna la longitud de la palabra.

6 Detalles de implementación

Toda la realización del proyecto fue utilizando el lenguaje de programación **Python**, manteniendo un archivo con el nombre de **requirements.txt** en donde

se encuentran todas las bibliotecas utilizadas. Se desarrolló un **makefile** para la fácil instalación y ejecución de la apliación visual por parte del usuario.

7 Conclusiones

En la sociedad actual, con el predominio de los teléfonos inteligentes y las redes sociales, el impacto que tiene la información en numerosos ámbitos de la vida, se mide en segundos y mientras, la cantidad de información existente está más allá de lo que cualquier individuo puede digerir. Es por ello que resulta imperativo el desarrollo de modelos automáticos que permitan recolectar y recuperar información relacionada a consultas. Para la realización de esta tarea se trataron dos modelos empleados en los sistemas de recuperación: vectorial y booleano; se implementaron y evaluaron respecto a las principales variables de comparación, apreciando así que los resultados obtenidos de la aplicación de estos no es muy buena, por esta razón es que los modelos usados en la actualidad son modificaciones y mezclas de estos modelos bases.

Además se construyó una interfaz visual para el fácil uso por parte del usuario de estos sobre un set de documentos y se desarrolló un algoritmo de expansión de consulta para ayudar al usuario a la hora de realizar una.