

Pre-entrega Proyecto Final de Sistemas de Recuperación de la Información

ALEJANDRO CAMPOS, DARIAN DOMINGUEZ

Facultad de Matemática y Computación
Universidad de la Habana
2021

Abstract

En el mundo de hoy se tiene acceso a mucha información. Sería muy complicado para una persona buscar, entre millones y millones de documentos, cuáles serían útiles y relevantes para su objetivo. Los sistemas de recuperación de la información facilitan este trabajo. Estos reciben una consulta por parte del usuario y procesan toda la colección de documentos, que luego de analizarlos y representarlos de una forma que el sistema entiende, devuelven en un corto período de tiempo aquellos archivos más relevantes a la petición del usuario.

I. INTRODUCCIÓN

Realizar búsquedas de información es una actividad diaria en todos los ambientes que reviste especial importancia. En áreas de investigación es una actividad que puede insumir un tiempo importante entre operaciones que van desde: definir los criterios, iniciar la búsqueda, filtrar los resultados, encontrar la información y luego tenerla a disposición de una manera clasificada y ordenada. La fase de búsqueda se inicia sobre las expectativas planteadas por el investigador y finaliza cuando el mismo encuentra finalmente el o los documentos que poseen la información de interés. [1]

Varios documentos pueden responder a la consulta del usuario, por ello, los sistemas implementan un ranking de documentos

ordenados según la relevancia a la petición del usuario, dependiendo del modelo sobre los que son implementados. Existen discímiles modelos para recuperar información, estos se diferencian en la manera de representar los documentos y la consulta, en dependencia de los términos que la conforman. A partir de esta forma de representación, se le asocia a cada documento un valor calculado mediante una función de ranking, que varía en dependencia del modelo utilizado.

Este informe tiene como objetivo describir el diseño, implementación, evaluación y análisis de un Sistema de Recuperación de Información.

II. DISEÑO DEL SISTEMA

Luego del estudio y análisis de los modelos estudiados en [2], se decidió usar el modelo vectorial para el desarrollo del sistema de recuperación de la información. En la presente sección se explicarán los motivos de esta selección, no sin antes explicar en qué consiste este modelo.

Brevemente, según este modelo, cada documento es representado mediante un vector de n elementos, siendo n igual al número de términos indizables que existen en la colección documental. Hay, pues, un vector para cada documento, y, en cada vector, un elemento para cada término o palabra susceptible de aparecer en el documento. Cada uno de esos elementos es cubierto u ocupado con un valor numérico. Si la palabra no está presente en el documento, ese valor es igual a 0. En caso contrario, ese valor es calculado teniendo en cuenta diversos factores, dado que una palabra dada puede ser más o menos significativa (tanto en general como, sobre todo, en ese documento en concreto); este valor se conoce con el nombre de peso del término en el documento. Las consultas son representadas también mediante un vector de las mismas características que las de los documentos. Esto permite calcular fácilmente una función de similaridad dada entre el vector de una consulta y los de cada uno de los documentos.[3]

El modelo vectorial es ampliamente usado en operaciones de recuperación de información, así como también en operaciones de categorización automática, filtrado de información, etc. Aunque tiene la desventaja de asumir independencia de términos, el cálculo del valor asociado a cada documento para realizar el ranking se determina por la cercanía de este con respecto a la consulta del usuario, definida por el coseno del ángulo entre el vector que representa al documento y el de la consulta. Podemos decir, además, que el esquema de ponderación de este modelo (tf-idf como se verá posteriormente) permite un mejor rendimiento en la recuperación de información.

III. IMPLEMENTACIÓN DEL SISTEMA

El sistema se ha implementado usando el lenguaje de programación python, cuyas bibliotecas ofrecen facilidades para el trabajo con el procesamiento textual. La implementación se encuentra en la carpeta **code**. En las siguientes subsecciones se explica el uso de cada uno de los módulos más importantes del sistema.

I. Procesamiento textual

En el módulo `text_processing.py` se realizan todas las fases del preprocesamiento textual, tokenización, normalización, lematización o reducción de las palabras a su raíz gramatical y eliminación de stopwords. Para esto se usó la librería *nltk*.

II. Representación de documentos y consultas

`representation.py` provee métodos para la representación de documentos y consultas. En este se calcula el *tf* de cada término en cada documento, así como *idf* de cada término y, a través de estos, se determina el peso de cada término i en el documento j (w_{ij}). Se precisan a continuación cada uno de los términos anteriores.

$$tf_{ij} = \frac{freq_{ij}}{\max_i freq_{ij}}$$

donde el numerador corresponde a la cantidad de veces que se repite el término i en el documento j , y el denominador es la cantidad de veces que se repite el término que más se repite en el documento j . También se tiene que

$$idf_i = \log \frac{N}{n_i}$$

donde N corresponde a la cantidad total de documentos de la colección y n_i a la cantidad de documentos en los que aparece el término

i. Finalmente, el peso de un término en el documento, viene dado por

$$w_{ij} = tf_{ij} * idf_i$$

. Como ya se dijo, los documentos son representados como vectores, cuyas componentes vienen dadas por los pesos de los términos que aparecen en estos.[4]

Las consultas también son representadas como vectores, el peso de un término en la consulta q viene dado por

$$w_{iq} = (a + (1 - a)tf_{ij}) * idf_i$$

donde a usualmente toma valor 0.4 o 0.5 [4]. Notemos que si $a = 0$ el peso de la consulta es el mismo que el del documento.

III. Documentos

Los documentos deben encontrarse en la carpeta "docs" en el mismo directorio del código del proyecto. El sistema trabajará con la colección de archivos con extensión .txt localizados en dicha ruta. En `document_processing.py` se procesan todos los documentos de la colección. A sus textos se les realiza el procesamiento textual y se calcula el peso de cada término resultante en cada documento.

IV. Función de similitud

El módulo `sim.py` contiene el cálculo de la función de similitud, mediante la cual se realiza el ranking de los documentos. Acorde al modelo empleado, esta función se calcula como el coseno del ángulo comprendido entre el vector que representa cada uno de los documentos y la consulta.

$$sim(d_j, q) = \frac{\vec{d}_j \cdot \vec{q}}{|\vec{d}_j| |\vec{q}|},$$

esto es

$$sim(d_j, q) = \frac{\sum_{i=1}^n w_{ij} w_{iq}}{\sqrt{\sum_{i=1}^n w_{ij}^2} \sqrt{\sum_{i=1}^n w_{iq}^2}}$$

[4]

V. Procesamiento de consulta y devolución de resultados

Al texto de la consulta se le realiza un preprocesamiento, como se expuso en la sección correspondiente. Luego se le calcula el peso de cada término para formar el vector consulta, a partir del cual la función de similitud le asocia un valor a cada documento y devuelve el ranking de estos, a partir de un umbral que se define previamente. Esto se realiza en `query_processing.py`.

VI. Expansión de consulta

El sistema implementa la expansión de consulta, lo cual permite expandir las consultas ingresadas por el usuario a fin de ampliar el espectro de las búsquedas que se realicen. El módulo `query_expansion.py` implementa esta funcionalidad, donde por cada término t de la consulta, esta es expandida automáticamente con sinónimos y palabras relacionadas con t .

VII. Main

El flujo del sistema es controlado por el módulo `main.py`, por el momento, en esta primera versión que se presenta, la consulta debe insertarse donde indica los comentarios del código. Igualmente puede modificarse el umbral a partir del cual se devuelve el ranking de documentos.

IV. EVALUACIÓN DEL SISTEMA

V. ANÁLISIS DE LOS RESULTADOS OBTENIDOS

VI. CONCLUSIONES

VII. TRABAJO FUTURO

- Incorporar al sistema una interfaz de usuario agradable
- Añadir funcionalidades al sistema como expansión de consulta y clustering de documentos

- Poner dinámica la ruta de la colección de documentos
- Incorporar al informe evaluación del sistema a partir de las métricas estudiadas y analizar los resultados obtenidos con estas
- Incorporar al informe conclusiones y recomendaciones para el trabajo futuro con el sistema implementado.

VIII. REFERENCIAS

- [1] http://sedici.unlp.edu.ar/bitstream/handle/10915/43840/Documento_completo.pdf?sequence=1&isAllowed=y
- [2] Conferencias 2 y 3 SRI para la carrera de Ciencia de la Computación, Universidad de La Habana. Curso 2022
- [3] <https://bid.ub.edu/04figure2.htm>
- [4] Conferencia 2 SRI para la carrera de Ciencia de la Computación, Universidad de La Habana. Curso 2022