Informe de Sistema de Recuperación de Información

Eziel Christians Ramos Piñón
https://github.com/ezielramos/information-retrieval-models.git

Índice

1. Introducción

Aunque desde mediados del siglo XX se viene trabajando en el área de la Recuperación de información, en los últimos años su relevancia ha aumentado notablemente. Entre otros posibles factores desencadenantes de este efecto, quisiera destacar dos: en primer lugar, el crecimiento espectacular y constante de la web, con el consiguiente aumento en el número de documentos digitales a disposición de los usuarios de la red. Es de gran importancia en la actualidad, que la información recuperada por el usuario sea capaz de satisfacer sus necesidades. Sin embargo, ésta es una de las grandes problemáticas que se presenta, debido fundamentalmente al gran volumen de información existente actualmente. Además, se requiere que los sistemas de búsqueda y recuperación de información ofrezcan resultados que posean alta relevancia para que el usuario obtenga lo que en realidad pueda interesarle. Una de las etapas fundamentales en el proceso de recuperación de información es la elección del modelo adecuado para representar los documentos y las consultas. Estos modelos tratan de calcular el grado en que determinado elemento de información responde a determinada consulta. Los tres modelos clásicos y más utilizados son el modelo booleano, vectorial y probabilístico. En el presente trabajo se describen las etapas del proceso de recuperación de información, desde el procesamiento de la consulta hecha por un usuario, a la representación de los documentos, el funcionamiento del motor de búsqueda y la obtención de los resultados.

2. Diseño del sistema

El sistema desarrollado cuenta con varias funcionalidades. Primero, dado una consulta, es necesario realizar el preprocesamiento y la representación de la consulta y los documentos. Posteriormente, el Modelo de Recuperación de Información es utilizado para obtener una ordenación de los documentos relevantes. Además, con el objetivo de mejorar los resultados obtenidos para una consulta, se implementó retroalimentacion de la relevancia. A continuación, cada una de estas funcionalidades son explicadas más extensivamente.

2.1. Preprocesamiento de la consulta y los documentos

Para el procesamiento de la consulta del usuario y de los documentos en el corpus es necesario representarlos de una manera lógica para ser procesados por la máquina. Para ello se realizan operaciones de procesamiento textual que permiten estructurarlos. Esto facilita la obtención de información necesaria para identificar los documentos relevantes. El siguiente preprocesamiento textual es usado para procesar la consulta y los documentos:

1. *Eliminación de los signos de puntuación*: Los signos de puntuación son eliminados, así como otros caracteres especiales que no se piensa que aporten información en la tarea de recuperación de información.

2. *Minúsculas*: Las mayúsculas/minúsculas no aportan información relevante, así que todo el texto es representado en minúsculas.

3. *Eliminación de stopwords*: los stopwords son las palabras que no proveen información útil para la clasificación de un texto.

4. *Stemming*: Es un método para obtener la forma canónica de la palabra. Hay algunos algoritmos de stemming que ayudan en sistemas de recuperación de información, en el sistema se usa la librería de python *nltk*, específicamente la clase *SnowballStemmer*.

2.2. Modelo de Recuperación de Información

Un modelo de recuperación de información (MRI) es un cuádruplo $[D, Q, F, R(q_j, d_j)]$:

- D es un conjunto de representaciones lógicas de los documentos de la colección.
- Q es un conjunto compuesto por representaciones lógicas de las necesidades del usuario. Estas representaciones son denominadas consultas.
- F es un framework para modelar las representaciones de los documentos, consultas y sus relaciones.
- R es una función de ranking que asocia un número real con una consulta $q_j \in Q$ y una representación del documento $d_j \in D$. La evaluación de esta función establece un cierto orden entre los documentos de acuerdo a la consulta.

2.2.1. Modelo Booleano

Como uno de los modelos utilizados se encuentra el MRI Booleano por su facilidad a la hora de implementación. Está basado en la teoría de conjuntos y el álgebra booleana, solo considera si los términos indexados están presentes o no en un documento, los pesos de los términos indexados son binarios, es decir, $w_{i,j} \in \{0, 1\}$ y las consultas están formadas por términos relacionados por tres conectores: not, and y or.

La similitud entre un documento $\vec{d_j}$ y la consulta q se define como:

$$sim(d_j, q) = \begin{cases} 1 & si \quad \exists \vec{q_{cc}} : (\vec{q_{cc}} \in \vec{q_{fnd}}) \wedge (\forall k_i, g_i(\vec{d_j}) = g_i(\vec{q_{cc}})) \\ 0 & en\ otro\ caso \end{cases}$$

2.2.2. Modelo Vectorial

Uno de los modelos de recuperación de información usado en este sistema es el MRI Vectorial ya que este es simple, rápido, y en algunos casos, brinda mejores resultados en la recuperación de información que el resto de los MRI clásicos. Sin embargo, sabemos que esto no significa que sea el mejor modelo; no existe un modelo general que dé solución a todos los problemas. De manera general, todos los modelos de recuperación de información tienen alguna noción de peso. Esta definición siempre está relacionada con la importancia de un término en un documento. Cada documento va a tener asociado un vector de términos indexados con la información del peso de cada uno obtenida a partir de una función, $\vec{d_j} = (w_{1j}, w_{2j}, \ldots, w_{nj})$, y la consulta $\vec{q_j} = (w_{1q}, w_{2q}, \ldots, w_{nq})$. Es común a todos los modelos que si un término no aparece en un documento su peso sea cero.

En el modelo vectorial, el peso de un término t_i en el documento d_j está dado por:

$$w_{i,j} = tf_{i,j} \times idf_i$$

Sea $freq_{i,j}$ la frecuencia del término t_i en el documento d_j. Entonces la frecuencia normalizada $f_{i,j}$ del término t_i en el documento $d_j (tf_{i,j})$ se calcula como:

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

donde el máximo se calcula sobre los términos del documento d_j.

Sea N la cantidad total de documentos en el sistema y n_i la cantidad de documentos en los que aparece el término t_i. La frecuencia de ocurrencia de un término t_i dentro de todos los documentos de la colección idf_i está dada por:

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

.

La medida $tf_{i,j}$ es una medida de similitud intra-documento. Se considera la frecuencia de cada término en un documento dividido entre la frecuencia máxima de ese mismo documento para normalizar esta medida. Esto se hace con el objetivo de evitar valores de frecuencia sesgados por la longitud del documento. Por otro lado, idf_i es una medida inter-documentos. Además de la medida de frecuencia de términos es importante analizar la frecuencia de ese término en todos los documentos de la colección. Un término que aparezca en pocos documentos de la colección tiene mayor valor frente a una consulta pues permite discriminar una mayor cantidad de documentos.

Sea además $freq_{i,q}$, la frecuencia del término t_i en el texto de consulta q, el peso de una consulta está dado por la siguiente fórmula:

$$w_{i,q} = (\alpha + (1 - \alpha)\frac{freq_{i,q}}{max_l freq_{l,q}}) \times \log(\frac{N}{n_i})$$

Donde α es una constante de suavisado, para la cual en conferencia se sugiere como valores mas usados 0,4 y 0,5, de manera general se obtuvieron mejores valores para $\alpha = 0{,}4$ por lo cual esta es la constante usada en la implementación de este modelo.

Una vez definidos los pesos de los documentos y las consultas solo falta la función de ranking para completar la definición del modelo vectorial. Esta función permite tener una ordenación por relevancia de los documentos y se basa en la similitud entre la consulta y los documentos empleando el coseno del ángulo comprendido entre los vectores documentos d_j y la consulta q:

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

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

Luego de obtener este ranking, se ordenan los documentos de acuerdo a su medida de similitud de mayor a menor, ya que documentos más similares tienen

una mayor similitud. Para la recuperación de los documentos, puede establecerse un umbral de similitud y recuperar los documentos cuyo grado de similitud sea mayor que este umbral.

2.2.3. Modelo Probabilístico

Otro de los modelos de recuperación de información usado en este sistema es el MRI Probabilístico, el cual intenta resolver el problema de la recuperación de información desde las probabilidades. Este tiene su fundamento en el cálculo de la probabilidad de que un documento sea relevante para la consulta realizada. La idea de este modelo es que dada una consulta q, existe un conjunto R(conjunto ideal) de documentos que contiene exactamente los documentos relevantes respecto a q. El proceso de consulta es un proceso de especificar las propiedades que cumple el conjunto ideal. El **Principio probabilístico** plantea que dada una consulta q y un documento d_j, el modelo probabilístico trata de estimar la probabilidad de que el usuario asuma el documento como relevante. Se asume que esta probabilidad depende únicamente de la representación de los documentos y las consultas. Para computar la relevancia de un documento, la similitud en el modelo probabilístico se define como el cociente o la razón de dos probabilidades, la probabilidad de relevancia (R) del documento d_j dividida por la probabilidad de no relevancia ($\bar{R}$) del mismo, ambas respecto a una consulta q.

$$sim(d_j, q) = \frac{P(R|\vec{d_j})}{P(\bar{R}|\vec{d_j})}$$

Para obtener estas probabilidades nos basamos en el Principio probabilístico, el Teorema de Bayes y la información de ocurrencia de los términos en el documento.

Para el cálculo de las probabilidades se empleó el Modelo de Independencia Binaria(BIM de sus siglas en inglés) que establece varios supuestos. El modelo BIM permite calcular las probabilidades en la función de similitud teniendo en cuenta las siguientes consideraciones:

- Los documentos y las con sultas son representados como vectores binarios de términos. De esta manera $\vec{d_j} = (w_{1j}, w_{2j}, \ldots, w_{nj})$ donde $w_{ij} \in \{0, 1\}$ y la consulta $\vec{q_j} = (w_{1q}, w_{2q}, \ldots, w_{nq})$ donde $w_{iq} \in \{0, 1\}$. Para ambos vectores se cumple que el peso de un término es 1 si el término ocurre en en la consulta o el documento correspondiente y 0 si no ocurre.
- Se asume que los términos ocurren de manera independiente en los documentos.
- Documentos diferentes pueden ser modelados como el mismo vector.
- Solamente interesa realizar una ordenación por relevancia (ranking) de los documentos.

La similitud en de un documento con la consulta es expresada mediante la fórmula

$$sim(\vec{d_j}, q) = \sum_{i=1}^{m} w_{i,q} * w_{i,j} * \log \frac{p_i(1 - r_i)}{r_i(1 - p_i)}$$

$\vec{d_j}$: vector binario que representa al j-ésimo documento de la colección
q: vector binario que representa a la consulta
$w_{i,q}$: peso del i-ésimo término en la consulta q
$w_{i,j}$: peso del i-ésimo término en el j-ésimo documento de la colección
m: cantidad de términos del vocabulario
p_i: probabilidad de ocurrencia del término t_i en documentos relevantes dada la consulta q
r_i: probabilidad de ocurrencia del término t_i en documentos no relevantes dada la consulta q

Donde p_i se estima proporcional a la probabilidad de ocurrencia en la colección(df_i es la frecuencia en los documentos, N cantidad de documentos en la colección) $p_i = \frac{1}{3} + \frac{2}{3}\frac{df_j}{N}$

Para estimar r_i se asume que la cantidad de documentos relevantes es mucho menor que la cantidad de documentos no relevantes. De esta manera su valor puede ser aproximado por la cantidad de documentos de la colección en que ocurre el término usando la expresión $r_i = \frac{n}{N}$, n:total de documentos de la colección donde aparece el término t_i.

2.3. Retroalimentación

Para los usuarios es difícil plantear las consultas de modo que expresen sus necesidades informativas. Muchas veces los SRI no logran dar respuesta a la necesidad de información del usuario. Por ello una idea para mejorar los resultados en la recuperación de información es involucrar al usuario en el proceso de recuperación de información para obtener mejores resultados.

Algoritmo general:

1. El usuario plantea la consulta
2. El sistema devuelve un conjunto de documentos
3. El usuario selecciona de estos documentos los que considera relevantes o no.
4. El sistema obtiene una mejor representación de las necesidades del usuario utilizando esta información
5. Se regresa al paso 2

Un algoritmo de retroalimentación aplicado al modelo vectorial es el Algoritmo de Rocchio el cual fue implementado.

En un SRI real, se tiene una consulta y solo se conoce parcialmente el conjunto de los documentos relevantes y no relevantes:

- Dr: Conjunto conocido de documentos relevantes
- D_nr: Conjunto de documentos no relevantes
- α, β, λ: pesos establecidos para cada termino de la consulta.

$$q_m = \alpha q_0 + \frac{\beta}{|Dr|} \sum_{d_j \in D_r} dj - \frac{\lambda}{|D_n r|} \sum_{dj \in D_{nr}} dj$$

donde se usó los siguientes valores de los parámetros:

$$\alpha = 1, \beta = 0{,}75, \lambda = 0{,}15$$

2.3.1. Resultados obtenidos

A continuación se muestran como mejoran los resultados para varias iteraciones del algoritmo de Rocchio en el modelo vectorial para la consulta:
q = *Computerized information retrieval systems. Computerized indexing systems.*
El dataset empleado para los datos que se muestran en la tabla es **CISI**.

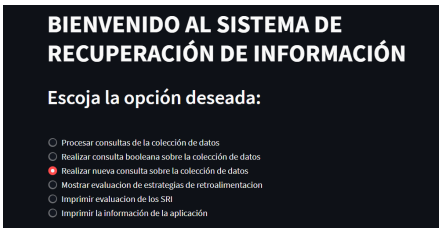
Iteraciones	F1
$I0$	0,14937
$I1$	0,15154
$I2$	0,15413
$I3$	0,16169
$I4$	0,1617
$I5$	0,1679
$I6$	0,16935
$I7$	0,16772
$I8$	0,17107

En la tabla se puede observar como después de 8 iteraciones hay una mejora de la métrica F1.

3. Implementación

El sistema fue implementado en Python 3.10.4. Las principales bibliotecas utilizadas y sus usos fueron:

- csv: Para almacenar los datos obtenidos de similitud de los documentos respecto a las consultas.
- nltk: Para el procesamiento de texto. Se usa para realizar stemming, para la eliminación de los stopwords mediante la lista de stopwords presente en nltk.
- streamlit: Fue usado para la realización de la interfaz visual del proyecto. Se creó una interfaz visual como se observa en la figura.



3.1. Ejecución del sistema

Para correr el sistema se ejecuta el siguiente comando desde una terminal abierta en la dirección del archivo ...\information-retrieval-models

```
streamlit run main.py
```

El proyecto funciona con las dependencias requeridas en los sistemas operativos Windows y Linux.

4. Evaluación del sistema

Con el objetivo de analizar el rendimiento del sistema este fue probado utilizando distintos corpus, que contenían varias consultas de prueba con los documentos que son relevantes. Las colecciones de prueba usadas fueron:

- **CRANFIELD**: Contiene 1400 documentos.
- **CISI**: Contiene 1460 documentos.
- **LISA**: Contiene 6004 documentos.

Dado que la cantidad de documentos de la colección **LISA** es muy grande, solo se usó esta colección para los modelos Vectorial y Probabilístico, en el caso del modelo Booleano se empleó **CRANFIELD** y **CISI**.

4.1. Medidas de evaluación

Para describir las medidas de evaluación implementadas comencemos con las siguientes definiciones:

- RR: conjunto de documentos recuperados relevantes.
- RI: conjunto de documentos recuperados irrelevantes.
- NR: conjunto de documentos no recuperados relevantes.
- NI: conjunto de documentos no recuperados irrelevantes

La precisión es una de las medidas fundamentales. La precisión tiende a decrecer cuando la cantidad de documentos recuperados aumenta. Calcula la fracción de los documentos recuperados que son relevantes:

$$P = \frac{|RR|}{|RR \cup RI|}$$

Por otro lado, tenemos el Recobrado, que es fundamental en los procesos de recuperación de información. En contraposición a la precisión el recobrado aumenta a medida que incorporamos más documentos a la respuesta, pues es cada vez más probable que los elementos del conjunto de documentos relevantes estén contenidos en nuestra respuesta. Esto hace que siempre sea posible tener el mayor valor de recobrado, 1. Esto se lograría devolviendo la colección completa

aunque, lógicamente, no es una solución factible. Representa la fracción de los documentos relevantes que fueron recuperados.

$$R = \frac{|RR|}{|RR \cup NR|}$$

Con el objetivo de lograr una compensación entre la precisión y el recobrado se define la medida F. Permite enfatizar la precisión sobre el recobrado y viceversa.

$$F = \frac{1 + \beta^2 PR}{\beta^2 P + R}$$

La medida F1 es un caso particular de la medida F en la que la precisión y el recobrado tienen igual importancia.

$$F1 = \frac{2PR}{P + R}$$

4.2. Resultados obtenidos

Con el objetivo de medir la evaluación del sistema fueron consideradas las tres medidas mencionadas con anterioridad: la precisión, el recobrado y la medida F1. Como en las colecciones de prueba se tenían la cantidad de documentos relevantes para cada consulta y el resultado de los Modelos de Recuperación de Información Vectorial y Probabilístico es un ranking entre los documentos relevantes (existe una ordenación entre ellos) se utilizaron las medidas mencionadas que trabajan con el ranking de los documentos. El promedio de cada una de estas mediciones de cada consulta individual es el calculado para evaluar el rendimiento del sistema y se compara en las siguientes tablas.

Promedio de las métricas en el corpus **CISI**

Medidas	Promedio vectorial	Promedio probabilistico
P	0,036554832673950474	0,11828341179784385
R	0,7089932986183519	0,522500174218293
$F(\alpha = 1{,}5)$	0,1005403353944055	0,1877397092499148
$F(\alpha = 0{,}5)$	0,04475084907732169	0,12684791891023778
$F1$	0,06751915623763698	0,15340952395249338

Promedio de las métricas en el corpus **CRANFIELD**

Medidas	Promedio vectorial	Promedio probabilistico
P	0,0092605405510765	0,27825099849584606
R	0,2142209245942814	0,35851021638369773
$F(\alpha = 1{,}5)$	0,02661647564572625	0,2556840984880823
$F(\alpha = 0{,}5)$	0,011404043469662034	0,24311130994637395
$F1$	0,01748393444869466	0,2403203608779195

Promedio de las métricas en el corpus **LISA**

Medidas	Promedio vectorial	Promedio probabilistico
P	0,010620915032679739	0,2957307398483869
R	0,018201997780244172	0,16233399965002532
$F(\alpha = 1{,}5)$	0,010965835083501088	0,16405686618678242
$F(\alpha = 0{,}5)$	0,009101473406700141	0,2128072396802762
$F1$	0,009579887593388075	0,17516178807016147

De manera general se puede observar en estos valores que los resultados obtenidos por el Modelo de Recuperación de Información Probabilístico son mejores que los resultados obtenidos por el Modelo vectorial. Aunque en el caso de la medida R(Recobrado) para el dataset **CISI**, el valor promedio evaluado sobre todas las query de la colección para el Modelo Vectorial es $R = 0{,}7089\ldots$ el cual supera el valor obtenido para el Modelo Probabilístico que tiene un valor de $R = 0{,}5225\ldots$.

5. Conslusiones

En el presente artículo, se describió la implementación de un Sistema de Recuperación de Información basado en el Modelo de Recuperación de Información Booleano, Vectorial y Probabilístico. Se expone el diseño del sistema según cada etapa de la recuperación de información. Varias de sus funcionalidades son explicadas: cómo se realiza el preprocesamiento y la representación de la consulta y los documentos, las principales características y cómo es implementado el MRI Booleano, Vectorial y Probabilístico. Las herramientas empleadas para la programación son explicadas. Además, se describe como se utiliza nuestro sistema y los aspectos más importantes de la interfaz visual del mismo. Luego, se realiza una evaluación del sistema en distintas colecciones de prueba empleando distintas métricas estudiadas en clase.