# Sistema de Recuperación de Información

Liset Silva Oropesa C511
Yenli Gil Machado C512

Facultad de Matemática y Computación, MATCOM, Universidad de la Habana, Cuba

**Abstract.** Análisis de la implementación de un *Sistema de Recuperación de Información (SRI)*. Explicación del Modelo de Recuperación de Información seleccionado y otras funcionalidades implementadas para mejorar su funcionamiento y cuáles podrían haber sido implementadas y con qué objetivo. Las principales limitantes que se observaron y cómo podrian solucionarse. Las métricas que se ejecutaron para la evaluación del modelo utilizado y un análisis detallado de cuáles fueron sus resultados. Definiciones globales del Modelo Vectorial y como se implementaron.

**Keywords:** Modelo Vectorial, Sistema de Recuperación de Información, Medidas de Evaluación

## 1 Introducción

En el mundo de hoy, con el avance vertiginoso de la computación, se han desarrollado herramientas que posibilitan el acceso a la información mediante Internet de cualquier persona del mundo. Pero para posibilitar que esta información llegue al usuario que la solicita es necesario desarrollar mecanismos o algoritmos capaces de encontrar dicha información necesitada. Con este objetivo surgen y se desarrollan los *Sistemas de Recuperación de Información*, para poder satisfacer, de manera rápida y precisa, las necesidades que se les presenta al usuario.

## 2 Conjunto de Datos

Se utilizaron dos set de datos para la ejecución del proyecto, estos son **CISI** y **CRAN**. *CISI* cuenta con 1460 documentos y 112 consultas, mientras que *CRAN* se compone de 1400 decumentos y 225 consultas. Ambos sets cuentan con la información de cuales son los documentos relevantes para cada consulta. Estos sets estan compuestos por los siguientes archivos: *(véase **Tabla 1.**)*

## 3 Modelo de Recuperación de Información

El Modelo de Recuperación de Información que se escogio implementar fue el **Modelo Vectorial** porque dicho modelo se basa en la *similitud* que pueda existir entre un documento y la consulta que se esta realizando, proporcionando un ranking de documentos respecto a la funcion de similitud que se implemente. Un factor importante es que tiene en cuenta la frecuencia de los terminos tanto en los documentos como en la consulta.

**Table 1.** Contenido de los sets de pruebas

| Archivos | Contenido |
|---|---|
| ALL.json | Conjunto de documentos del set, representados por un id, Titulo, Autor y Texto. |
| QRY.json | Conjunto de consultas realizadas por especialistas en el tema que aborda el set |
| REL.json | Relación de cuales documentos son relevantes para cada una de las consultas del set. |

### 3.1 Definiciones Teoricas

**Definition 1.** *El* peso $w_{i,j}$ *asociado al par* $(t_i, d_j)$ *es positivo y no binario. Sea el* peso $w_{i,q}$ *asociado al par* $(t_i, q)$ *y nla cantidad total de terminos indexados en el sistema. Entonces los vectores consulta* $q$ *y documentos* $d$ *se representan:*

$$q = (w_{1,q}, w_{2,q}, ..., w_{n,q}) \tag{1}$$

$$d_j = (w_{1,j}, w_{2,j}, ..., w_{n,j}) \tag{2}$$

Para calcular el valor de dichos *pesos* de los terminos en los documentos y consultas tenemos las siguientes proposiciones:

**Proposition 1.** *Sea* $freq_{i,j}$ *la frecuencia de ocurrencia del término* $t_i$ *en el documento* $d_j$*. Entonces la* frecuencia normalizada $f_{i,j}$ *se calcula de la siguiente manera:*

$$f_{i,j} = \frac{freq_{i,j}}{max_l freq_{l,j}} \tag{3}$$

*Donde el máximo término* $l$ *es el término que mayor frecuencia de ocurrencia posea en el documento* $d_j$*.Si el término no aparece en el documento entonces* $f_{i,j} = 0$*.*

**Proposition 2.** *Sea* $N$ *el total de documentos del sistema y* $n_i$ *la cantidad de documentos donde ocurre el término* $t_i$*. Se define como frecuencia de ocurrencia de un término* $t_i$ *dentro de la colección de documentos* $idf_i$ *y esta dada por:*

$$idf_i = \log \frac{N}{n_i} \tag{4}$$

**Proposition 3.** *El peso* $w_{i,j}$ *del término* $t_i$ *en el documento* $d_j$ *se calcula:*

$$w_{i,j} = f_{i,j} * idf_i \tag{5}$$

**Proposition 4.** *El peso* $w_{i,q}$ *del término* $t_i$ *en la consulta* $q$ *se calcula:*

$$w_{i,q} = (\alpha + (1 - \alpha) * \frac{freq_{i,j}}{max_l freq_{l,j}}) * \log \frac{N}{n_i} \tag{6}$$

Finalmente, para calcular la similitud utilizaremos la **Función de Ranking** que calcula la correlación utilizando el coseno del ángulo comprendido entre los vectores documentos $d_j$ y las consultas $q$:

$$sim(d_j, q) = \frac{d_j * q}{|d_j| * |q|} \tag{7}$$

Esta *función de ranking* es la que nos va a permitir determinar cuales son los documentos que se encuentran *más cerca* de la consulta realizada por el usuario.

### 3.2 Consideraciones de la implementación

Primero se realiza un procesamiento sobre los documentos y las consultas, para determinar cuales son los términos que incurren en el sistema, añadirlos a una lista de términos(*ListTerm*), donde cada término tiene asociado un vector cuyo *len* es la cantidad de documentos que contiene el sistema y en el se guarda la frecuencia de ocurrencia de los términos en cada documento. De la misma manera se procede con las consultas, para determinar la frecuencia de ocurrencia de cada uno de los términos en la misma.

Cada una de las definiciones expuestas en la sección anterior, fueron definidas y calculadas para el desarrollo del **Modelo Vectorial**, obteniendo un *Ranking* de documentos, en donde los primeros son los que se encuentran *más cerca* de la consulta, y devolviendo como recuperados los primeros 20 documentos del *Ranking*, puesto que representan menos del 2% de los documentos totales de los sets.

El proyecto fue implementado en *Python*, uttilizando principalmente la libreria numpy para el tratamiento de las matrices de los términos y los vectores de documentos y consultas. La estructuración del mismo viene dado por la siguiente tabla:

**Table 2.** Estructuracion del proyecto

| Archivos | Contenido |
|---|---|
| modelo.py | Implementación del Modelo Vectorial. |
| evalmodelo.py | Implementación de las medidas de evaluacion del modelo |
| prodoc.py | Procesamiento de los documentos. |
| proccon.py | Procesamiento de las consultas. |
| aplicacion.py | Estructuración del proyecto e intervaz gráfica. |

## 4 Funcionalidades Adicionales

Como funcionalidad adicional se implementó la *Retroalimentación* con la finalidad de proporcionar un mejor resultado cuando se halle el *Ranking* de docu-

mentos. Ambos conjuntos de sets de pruebas tienen por cada consulta cuales de los documentos son relevantes, entonces mediante el **Algoritmo de Rocchio** se obtiene una nueva consulta a la que se le aplica el *Modelo Vectorial*. Esto se puede extender para una implementación donde el usuario determine cuales son, a su criterio, los documentos relevantes y volver a procesar la consulta hasta que el usuario quede satisfecho con los documentos que se le brindan.

Una funcionalidad que no se desarrolló en el proyecto, pero que resultaria muy útil para predecir que documentos pueden interesarle al usuario, son los *Algoritmos de Agrupamiento*, en especial el *K-Means*, pues con el se podrían agrupar en $k$ grupos los documentos y al procesar una consulta se sugierenn los documentos que pertenezcan a los mismos grupos que los 10 primeros documentos del *ranking*.

## 5 Medidas de Evaluación

Para la evaluación del modelo implementado, se tuvieron en cuenta varias métricas que se explicaran a continuación:

1. **Precisión**: Conjunto de documentos recuperados que son relevantes.
2. **Recobrado**: Conjunto de documentos relevantes que fueron recuperados.
3. **MedidaF**: Permite enfatizar la *Precisión* sobre el *Recobrado* o viceversa. En particular en este proyecto se enfatizó en el *Recobrado*, pues nos interesa más determinar la mayor cantidad de relevantes que fueron recuperados. Para esto se le asignó $\beta = 0.5$
4. **MedidaF1**: Se utiliza para armonizar la *Precision* y el *Recobrado*. Es un caso particular de la *MedidaF*. Un valor máximo de esta medida representa el esfuerzo para encontrar el mejor compromiso entre la *Precisión* y en *Recobrado*.
5. **R-Precisión**: Mide la *Precisión* en la $R$ posición del *Ranking* de documentos relevantes a una consulta. En particular, en nuestro proyecto lo realizamos hasta la posición 15 del *ranking* de documentos.
6. **Fallout**: Esta medida se utiliza para determinar la cantidad de documentos irrelevantes en el *ranking*. Igualmente, tenemos en cuenta hasta la posición 20 del *ranking*.

### 5.1 Resultados Obtenidos

A partir de las medidas de evaluación anteriores, aplicadas a los 2 set de datos utilizados: *CRAN* y *CISI*, obtuvimos los siguientes resultados:

1. Utilizando el set de prueba *CRAN*, hallando *15-Precisión* y el *Fallout* hasta la posición 20 del ranking en las 10 primeras consultas. *(véase **Tabla 3.**)* Observamos que realiza un *Recobrado* del 40%, que no es perfecto pero es una medida medianamente buena porque nos implica que se recuperaron un promedio del 40% de los documentos relevantes por consulta. Mientras que el 1% del total de documentoss irrelevantes fueron recuperados.

**Table 3.** CRAN ejecutando las 10 primeras consultas

| Medida | Resultado |
|---|---|
| Precisión | 0.19000000000000003 |
| Recobrado | 0.4279233716475095 |
| MedidaF | 0.20116930283920903 |
| MedidaF1 | 0.23234542056548912 |
| 15-Precisión | 0.19999999999999998 |
| Fallout | 0.011651416744406852 |

**Table 4.** CRAN ejecutando las 10 primeras consultas

| Medida | Resultado |
|---|---|
| Precisión | 0.19000000000000003 |
| Recobrado | 0.4279233716475095 |
| MedidaF | 0.20116930283920903 |
| MedidaF1 | 0.23234542056548912 |
| 15-Precisión | 0.19999999999999998 |
| Fallout | 0.008631561353584788 |

2. Utilizando el set de prueba *CRAN*, hallando *15-Precisión* y el *Fallout* hasta la posición 15 del ranking en las 10 primeras consultas.*(véase **Tabla 4.**)* Como podemos observar, el *Fallout* creció lo que significa que encontró un mayor número de documentos recuperados que no son relevantes.
3. Utilizando el set de prueba *CISI*, hallando *15-Precisión* y el *Fallout* hasta la posición 15 del ranking en las 10 primeras consultas.*(véase **Tabla 5.**)*

**Table 5.** CISI ejecutando las 10 primeras consultas

| Medida | Resultado |
|---|---|
| Precisión | 0.08720643877293495 |
| Recobrado | 0.0751341367898166 |
| MedidaF | 0.07343301641785624 |
| MedidaF1 | 0.23234542056548912 |
| 15-Precisión | 0.08666666666666666 |
| Fallout | 0.009536838562679579 |

En este set de datos las medidas de evaluación demuestran una clara disminución, por lo que demuestran que el sistema no esta recuperando una cantidad media aceptable de documentos relevantes. Es válido destacar que

este set de datos tiene mayor número de documentos y por tanto, mayor cantidad de palabras.

4. Utilizando el set de prueba *CISI*, hallando *15-Precisión* y el *Fallout* hasta la posición 15 del ranking en las 20 primeras consultas. *(véase **Tabla 6.**)*

**Table 6.** CISI ejecutando las 20 primeras consultas

| Medida | Resultado |
|---|---|
| Precisión | 0.08 |
| Recobrado | 0.07166581951451637 |
| MedidaF | 0.06924300170138739 |
| MedidaF1 | 0.06475198141405869 |
| 15-Precisión | 0.07999999999999999 |
| Fallout | 0.009741944021937489 |

En este caso se procesaron las 20 primeras consultas del set *CISI* y no se observa una gran diferencia con respecto al ejemplo anterior, por lo que demuestra que el sistema se comporta persistentemente con todas las consultas. Tanto las medidas de *Precisión* como de *Recobrado* muestran bajos índices, por lo que se recuperaron pocos documentos relevantes. Así como creció la cantidad de documentos irrelevantes recuperados en los primeros 20 del *ranking*.

5. Utilizando el set de prueba *CRAN*, hallando *15-Precisión* y el *Fallout* hasta la posición 15 del ranking en las 20 primeras consultas. *(véase **Tabla 7.**)*

**Table 7.** CRAN ejecutando las 20 primeras consultas

| Medida | Resultado |
|---|---|
| Precisión | 0.15250000000000002 |
| Recobrado | 0.4235450191570881 |
| MedidaF | 0.16675632180371286 |
| MedidaF1 | 0.2024594597651462 |
| 15-Precisión | 0.16333333333333336 |
| Fallout | 0.009012447140440402 |

Como se puede observar en la tabla anterior con respecto a los anteriores ejemplos del set *CRAN*, la *Precisión* se mantiene alrededor de los mismos parámetros, de la misma manera que el *Recobrado*, sin embargo tanto la *MedidaF* como la *MedidaF1* disminuyeron sus valores. Pero en cuanto a el *Fallout* muestra un claro incremento, lo que representa el aumento del número de documentos recuperados irrelevantes.

## 6 Limitaciones

El principal problema existente en el *Sistema de Recuperación de Información* implementado se encuentra en la demora al ejecutar el *Modelo Vectorial* en las consultas. Producto a que el valor de la frecuencia de ocurrencia de términos se guarda en una matriz bidimensional $A_{i,j}$ donde el peso $w_{i,j}$ representa la frecuencia de ocurrencia del término $t_i$ en el documento $d_j$, esta matriz en ambos sets de prueba posee grandes dimensiones y por tanto se demora en realizar las verificaciones y calculos en las consultas. Esto trae consigo que el sistema se tarde un tiempo en procesar una cierta cantidad de consultas, y mientras mayor sea el número a procesar, más se demora.

Otra problemática, que incide indirectamente en recuperado de documentos, es que el *Modelo Vectorial* no tiene en cuenta las relaciones entre las palabras del documento; es decir, asume que no existe relación entre los términos, que son independientes.

## 7 Guía Visual

Se deben tener instalados los paquetes de *numpy*, *streamlit* de *Python*. Para ejecutar el proyecto se abre una consola con la dirección de la carpeta que contiene el código del programa y se ejecuta la siguiente sentencia:

```
streamlit run aplicacion.py
```

Cuando inicia la páguina web, se debe escribir en el recuadro de texto *CRAN* o *CISI*, ambos nombres en mayúsculas para que el proyecto determine cual de los sets de datos debe utilizar. Después el usuario puede elegir entre procesar las primeras 10 consultas del set o ejecutar el mismo una consulta. Si elige que se ejecuten las preguntas del set, ira devolviendo al usuario el *top10* de los documentos que recuperó por cada consulta y al final el resultado de la evaluación de las métricas explicadas anteriormente. Mientras que en caso contrario, si el usuario realiza una consulta, procesara esta consulta y devolvera también el *top10* de los documentos recuperados por el modelo.

## References


1. Conferencias del Curso de Sistema de Información, Facultad de Matemática y Computación (MATCOM) (2021)
2. Jurafsky,D. , Martin, J.H.: Speech and Lenguage Processing (2020)
3. Mitra,B. , Craswell, N.: An Introduction to Neural Information Retrieval (2018)
4. Romá,T.: LOS SISTEMAS DE RECUPERACIÓN DE LA INFORMACIÓN (SRI) DE LAS BASES DE DATOS DOCUMENTALES Y LA CALIDAD DE LOS RESULTADOS OBTENIDOS (2014)