

# Sistemas de Recuperación de Información

Claudia Olavarrieta Martínez y Marcos Adrián Valdivié Rodríguez

Universidad de la Habana

**Abstract.** El volumen de información existente aumenta de forma drástica diariamente, y con ello, la necesidad de los usuarios de sistemas que les permitan recuperar datos de manera eficiente. En este trabajo se exploran tres modelos de Recuperación de Información, denominados clásicos en la literatura. La implementación de los modelos booleano, vectorial y probabilístico, así como modificaciones para su mejora se realizan con el propósito de realizar una comparación y análisis de sus comportamientos en distintos escenarios.

## 1 Introducción

La cantidad de información que se produce y almacena crece diariamente de manera exponencial, y con ello aumenta la necesidad de los usuarios de poder acceder a estas de forma eficiente. De aquí que surjan los sistemas de recuperación de información con el objetivo de facilitar el acceso de los usuarios a todos los datos, la mayoría de las veces sin orden ni estructura.

Un sistema de recuperación de información (SRI) debe ser capaz de almacenar, recuperar y mantener información. Entiéndase por información: textos, imágenes, productos, audio, video, entre otros. El objetivo principal de este tipo de sistemas es minimizar la carga de un usuario para localizar la información que necesita. Esto puede definirse como el tiempo que debe esperar un usuario para elaborar la consulta, ejecutarla y filtrar entre las respuestas lo que resulte relevante [2].

Dentro de los SRI existen varios modelos denominados clásicos para recuperar los datos, de los cuales se derivan muchos de los modelos actuales, los cuales resultan más eficientes. Dentro de dichos modelos se encuentran el modelo booleano, el modelo probabilístico y el modelo vectorial. Además, sobre estos modelos se pueden implementar sistemas de retroalimentación, expansión de consultas y sugerencias de información similar, lo que tributa a un mejoramiento de los resultados de cada sistema.

Este trabajo tiene un enfoque puramente académico y se traza como objetivo principal la implementación de los mencionados modelos clásicos. Además se implementaron sistemas de retroalimentación directa e indirecta con el objetivo de mejorar los resultados. Se realizará también, una comparación de los modelos empleados con dos conjuntos de datos y distintas métricas.

## 2 Modelos básicos de Recuperación de Información

Fueron implementados los tres modelos clásicos de recuperación de información: el modelo booleano, vectorial y probabilístico. Se realizaron algunas modificaciones y se incluyó retroalimentación para mejorar los resultados de las consultas de los usuarios.

### 2.1 Modelo Booleano

El modelo booleano se distingue por el uso del álgebra booleana y la teoría de conjuntos para obtener, dado una consulta, el conjunto de documentos relevantes a esta. En este caso, el conjunto de documentos resulta relevante si satisface la consulta conteniendo los términos de esta, no utiliza función de similitud. Dada la sencillez de este modelo se considera el punto de inicio en el campo de la recuperación de información y fue un sistema que recibió gran atención y fue adoptado por muchos de los primeros sistemas bibliográficos [1].

El modelo fue implementado siguiendo las principales características del modelo clásico, excepto en la forma de realizar la consulta. Mientras las expresiones booleanas tienen una semántica precisa, resulta complicado expresar una necesidad de información de esta manera, así como la traducción y el procesamiento para poder ser utilizada por el modelo [1]. Por lo tanto, la forma de realizar la consulta fue modificada y consiste en solamente una conjunción de los elementos. Es decir, se busca y retorna de los documentos de la colección aquellos que contengan cada una de las palabras de la consulta.

Se implementó además una variación de este modelo, el cual llamaremos "booleano relajado" que consiste en ordenar los documentos según un criterio de similitud con la consulta. Siguiendo el enfoque de [3] se utiliza una variación de la medida de similitud de Jaccard, a la que nombran "Jaccard con penalización". De esta forma se penaliza la puntuación del ranking teniendo en cuenta el número de términos con los que realmente coincide una consulta cuando se compara con un documento de la colección. La fórmula planteada, donde  $Q$  representa la consulta y  $D$  el documento, es la siguiente:

$$PJaccard(Q, D) = \frac{|D| \cap |Q|}{|D| \cup |Q|} - (1 - \frac{|D| \cap |Q|}{|Q|})$$

Siguiendo esta idea se utilizaron otras medidas: Socal & Sneath, Dice y Kulczyinski. Los mejores resultados se obtuvieron con la medida *PJaccard*, sin embargo, estos no fueron significativos al compararlos con los otros modelos.

### 2.2 Modelo Vectorial

El modelo vectorial consiste en el uso de las operaciones entre vectores y el álgebra lineal para obtener los documentos relevantes. La implementación de este modelo sigue todas las pautas de modelo clásico. Dada una consulta y el conjunto de documentos, se ordenan estos últimos según la similitud, calculada

como el coseno del ángulo que posean los vectores que representan al documento y a la consulta. De esta forma se obtiene un conjunto de documentos como respuesta, ordenados, que pueden poseer coincidencias parciales y que por tanto, satisfacen la necesidad de información del usuario de una mejor manera [1].

Se implementó además, el algoritmo de Rocchio para la retroalimentación en el modelo vectorial.

### 2.3 Modelo Probabilístico

Este modelo está basado en la Teoría de las Probabilidades y el Teorema de Bayes para su funcionamiento. Dado una consulta y un documento el modelo intenta estimar la probabilidad de que el usuario encontrará dicho documento interesante (relevante).

Se implementó además la Pseudo-Retroalimentación de Relevancia para mejorar los resultados obtenidos.

## 3 Colección de Documentos

El sistema cuenta con cuatro posibles colecciones de documentos a utilizar, todos en idioma inglés, aunque solo Cranfield y Med fueron usados para el análisis estadístico de los modelos al ser los únicos que poseían un conjunto de posibles consultas anotadas con sus posibles respuestas. Estas colecciones se cargan en memoria dependiendo de la escogida por el usuario en la interfaz visual.

### 3.1 Cranfield

Los experimentos de Cranfield realizados en la década de 1960 fueron una serie de estudios experimentales sobre la recuperación de información realizados por Cyril W. Cleverdon en el College of Aeronautics, hoy conocido como Cranfield University. El conjunto de datos, conocido como Cranfield 1400, cuenta con 1400 resúmenes de documentos, 365 consultas y las opiniones de relevancia de todos los pares  $\langle consulta, documento \rangle$ . Para cada documento se cuenta con: identificador, título, autor y resumen [4].

### 3.2 IMDB

IMDb son las siglas de *International Movie Database*, el cual consiste en una base de datos online de información relacionada con películas, series, videojuegos, entre otros. Contiene además, biografías, videos de casting, producción, tráilers, comentarios de fans y críticas especializadas. Con los ratings y comentarios de los usuarios conforman un ranking del contenido que ofrecen [5]. La colección de documentos utilizada en el proyecto consiste en un conjunto de 10662 comentarios de usuarios sobre algún contenido del sitio.

### 3.3 Newsgroups

El tercer conjunto utilizado consiste en grupos de correos que se corresponden a diversos temas. La estructura de estos correos no está predefinida, por lo que el procesamiento de los mismos consistió en leer los ficheros completamente. Debido a la densidad de este dataset, se decidió incorporar solamente tres temas de los más de quince suministrados.

### 3.4 MED

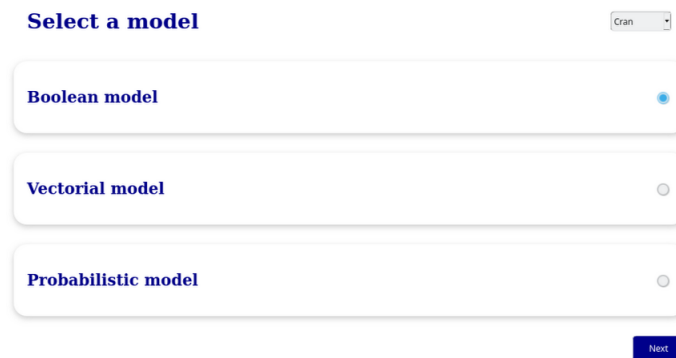
Conjunto de datos relacionados con biología, que de manera similar al conjunto Cranfield contiene los pares de relevancia  $\langle consulta, documento \rangle$ . Los documentos solo cuentan con identificador y resumen, y se tiene un total de 1033 documentos, 30 consultas preelaboradas y los documentos que resultan relevantes a estas.

## 4 Aplicación visual

Se desarrolló una interfaz visual utilizando el framework de python *Django* para facilitar la interacción con los distintos aspectos de la aplicación.

El flujo de la aplicación es el siguiente:

1. Primeramente el usuario es dirigido a la página principal (Figura 1) donde selecciona uno de los tres modelos clásicos y el conjunto de datos con el que desea trabajar.



**Fig. 1.** Página para la selección del modelo a estudiar

2. Se redirige al usuario a la página de inserción de consultas (Figura 2) donde aparece un cuadro de texto para insertar la consulta y opciones de personalización según el modelo escogido (Figura 3):

**Fig. 2.** Cuadro de texto para insertar consulta

- La opción *Relaxed Consult* se utiliza para activar el modelo booleano relajado mencionado anteriormente (Figura 3 a)).
- La opción *Enable retroalimentation* (Figura 3 b)) se utiliza para activar la retroalimentación asistida en el modelo vectorial, al enviar una consulta con dicha opción seleccionada, el usuario verá una lista desplegable en la esquina superior derecha de los documentos devueltos mediante la cual será capaz de marcar a dicho documento como relevante o no para la consulta insertada. Además, se mostrará un botón con el título *Send retroalimentation* que deberá ser usado para obtener la nueva colección de documentos relevantes con respecto a la retroalimentación proporcionada (Figura 4).
- La opción *Retroalimentation iterations* (Figura 3 c)) se utiliza para indicar la cantidad de iteraciones que se desee hacer para la Pseudo-Retroalimentación de relevancia del modelo probabilístico. Su valor por defecto es 1, lo que indica que no se hará ninguna iteración y se devolverán los documentos según el algoritmo original.
- La opción *Size* se utiliza para indicar la mayor cantidad de documentos que se desea obtener como respuesta para la consulta especificada. Su valor por defecto es 20.

**Fig. 3.** Opciones de personalización: a) Modelo booleano, b) Modelo Vectorial, c) Modelo Probabilístico

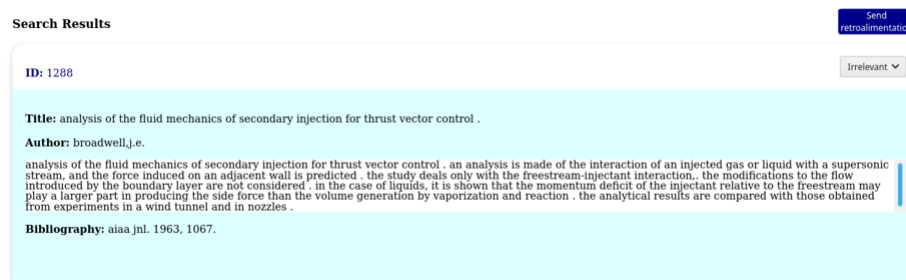


Fig. 4. Retroalimentación en el modelo vectorial

3. Al presionar el botón buscar los resultados serán mostrados en el fondo de la página.

## 5 Implementación

El trabajo fue desarrollado con el lenguaje python y el framework Django para la interfaz visual. El código fuente puede verse en el siguiente [repositorio de Github](#)

### 5.1 Booleano y Probabilístico

Es importante destacar en los modelos Booleano y Probabilístico el uso de *numpy* para la representación de las consultas y los documentos como colecciones. De esta forma se ahorra una cantidad considerable de memoria a la hora del procesamiento, ya que las palabras que aparecen en cada documento o consulta son en general una pequeña parte del total de palabras lo que hace innecesario tener una constancia explícita.

Además, *numpy* posee implementaciones de las principales operaciones entre conjuntos (unión, intersección, diferencia...), lo que permite facilitar y acelerar el manejo de los documentos y consultas del modelo como arreglos de *numpy*.

### 5.2 Aspectos importantes

Se utilizó el módulo *sklearn* para el preprocesamiento de las consultas y los documentos. Específicamente las clases *CountVectorizer* y *TfidfTransformer*, la primera permite obtener las representaciones de los documentos contando la cantidad de ocurrencias de cada palabra y la segunda, las representaciones *TF-IDF* utilizadas en el modelo vectorial. La clase *CountVectorizer* permite además, obtener la representación binaria de los documentos de manera sencilla.

Se omitió el procesamiento de las *stopwords*, ignorando aquellas palabras que aparecieran en una gran cantidad de documentos. Específicamente, la clase *CountVectorizer* permitía establecer un umbral, en este caso de 85%, donde las palabras que aparecen en más de esta cantidad de documentos fueron eliminadas.

La clase *EnglishStemmer* del módulo *nltk* fue usada, como su nombre lo indica, para realizar el proceso de stemming del conjunto de palabras de los documentos.

El módulo *basics* consta de las definiciones de clases usadas comúnmente por todos los modelos utilizados en la aplicación.

- *BasicModel*: Constituye la clase básica para todos los modelos, esta clase recibe en su constructor objetos de tipo *BasicStorage*, *Vectorizer*, *BasicConsultor*, *BasicProcessor* y *BasicQueryProcessor* que utilizan para almacenar y analizar los documentos y las consultas. Además, posee los métodos *AddDocument* que se encarga de analizar un documento y guardarlo en el *Storage*, y el método abstracto *Consult*, que dada una consulta debe retornar el conjunto de documentos relevantes a ella.
- *BasicConsultor*: Constituye la clase básica para definir una consulta para un modelo en específico, contiene un método abstracto llamado *Consult* que dado un conjunto de documentos y una consulta debe retornar los documentos relevantes a dicha consulta.
- *BasicProcessor*: Constituye la clase básica para definir la forma en que un documento será analizado para obtener su representación vectorial. Recibe como argumento un objeto de la clase *Vectorizer* y contiene un método abstracto llamado *ProcessDocument* que dado un documento debe retornar la representación vectorial del mismo.
- *BasicQueryProcessor*: Constituye la clase básica para definir la forma en que una consulta será analizada para obtener su representación vectorial. Recibe como argumento un objeto de la clase *Vectorizer* y contiene un método abstracto llamado *ProcessQuery* que dado una consulta debe retornar la representación vectorial de la misma.
- *BasicStorage* : Constituye la clase básica para representar la forma en que los documentos serán manipulados y guardados. Contiene los métodos abstractos *SaveDocument*, *GetAllDocuments*, *GetDocuments* y *GetDocumentRepresentations*.
- *Consult* y *Document*: Constituyen las clases básicas para representar respectivamente una consulta y un documento. Contiene campos relevantes a estas entidades como el conjunto de documentos relevantes en el caso de las consultas y el título o autor en el caso de los documentos.
- *Utils*: Contiene el método *angle*, que retorna el coseno del ángulo entre dos vectores.
- *Vectorizer*: Contiene los métodos y atributos necesarios para procesar un vector para cada uno de los modelos implementados. Utiliza el objeto *EnglishStemmer* de *nltk* para realizar un *stemming* de los documentos y las consultas y los objetos *CountVectorizer* y *TfidfTransformer* de *sklearn* para obtener los vectores representativos.

Las implementaciones específicas de estas clases para cada uno de los modelos implementados se encuentran en los módulos Boolean, Vectorial y Probabilistic respectivamente.

### 5.3 Requerimientos

Los requerimientos del proyecto pueden verse en el fichero [requirements.txt](#)

## 6 Evaluación de los modelos

Se evaluó el comportamiento de los tres modelos implementados con dos conjuntos de datos: el Cranfield y el Med. Los resultados completos de la experimentación, así como los valores de los parámetros utilizados para esta, pueden encontrarse en los archivos [evaluationCran.ipynb](#) y [evaluationMed.ipynb](#).

Se utilizaron distintas medidas para evaluar los modelos, entre ellas:

- Precisión
- Recobrado
- Medida F con  $\beta = \sqrt{2}$
- Medida F1
- Tiempo de ejecución de la consulta
- Mean Average Precision (MAP): que consiste en tomar para cada documento relevante la razón de la cantidad de documentos relevantes que se encuentran hasta su posición en el ranking de documentos y la cantidad de documentos que se encuentran por delante de él, para luego devolver la media aritmética de dichos valores.

### 6.1 Conjunto de datos Cranfield

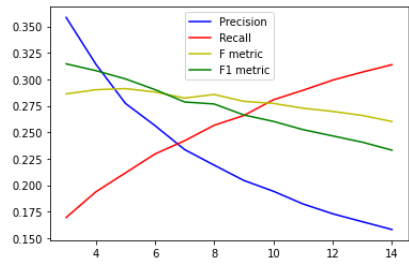
El modelo booleano, en su forma clásica, fue el que presentó mejores resultados logrando obtener valores de 0.69 de Precisión y Recobrado, y de 0.98 de las medidas F y F1. Sin embargo, esto no es completamente determinante, ya que el conjunto de consultas de la colección Cranfield está orientado a relacionar cada consulta con los documentos relevantes según su contenido, obviando en muchos casos la coincidencia parcial o total de los mismos.

La versión *relajada* de este modelo a la que se hace referencia anteriormente presentó los peores resultados, pero debemos incluirla debido a su utilidad, ya que permite la ordenación según la similitud con la consulta de los documentos de prueba. Las evaluaciones de las medidas de Precisión y Recobrado rondaron el valor de 0.24, mientras que las medida F fueron aproximadamente 0.28 (Figura 5).

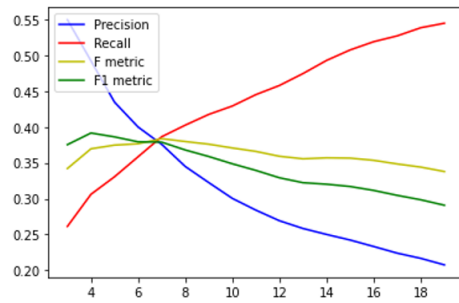
El modelo vectorial se desempeñó de manera sobresaliente, siendo no solo el más rápido con apenas 0.03 segundos promedio por consulta, sino también el que logró el mejor balance entre Precisión, Recobrado y las medidas F, llegando a valores de aproximadamente 0.37 en todas estas, y un MAP de 0.41 (Figura 6).

Los experimentos con la retroalimentación del modelo vectorial fueron aún mejores, logrando elevar las medidas de Precisión, Recobrado, F y F1 hasta valores de 0.49, pero a expensas de un mayor tiempo de cómputo (0.35 seg) (Figura 7).

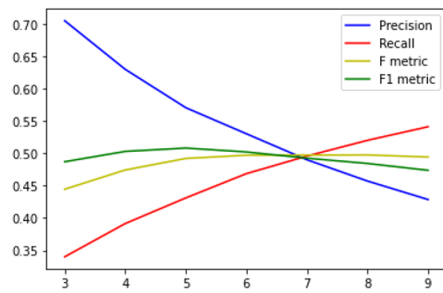




**Fig. 5.** Medidas para el modelo booleano relajado con el conjunto de datos Cranfield. Tamaño de la cantidad de documentos recuperados contra puntuación obtenida por las medidas de evaluación

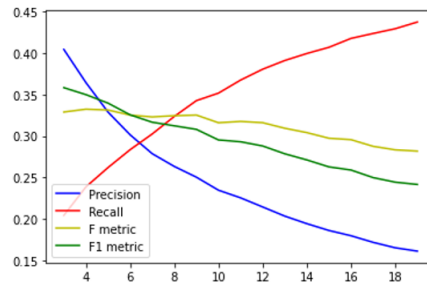


**Fig. 6.** Medidas para el modelo vectorial sin retroalimentación, según la cantidad de documentos devueltos.



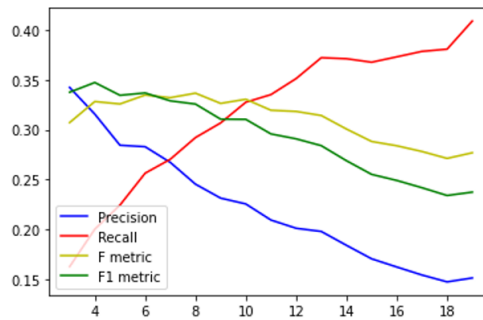
**Fig. 7.** Medidas para el modelo vectorial con retroalimentación, según la cantidad de documentos devueltos.

En cuanto al modelo probabilístico, los resultados mostraron valores de 0.30 para la Precisión, 0.28 para el recobrado y 0.32 para las medidas F, con un tiempo de consulta promedio de 0.08, todo esto para el modelo sin retroalimentación (Figura 8).



**Fig. 8.** Medidas para el modelo probabilístico sin retroalimentación, según la cantidad de documentos devueltos

Al incorporar la retroalimentación automática, los valores de la precisión y recobrado disminuyeron hasta 0.27 y 0.25 respectivamente, pero aumentaron ligeramente hasta 0.33 en cuanto a las medidas F. El tiempo promedio, por otra parte, aumentó considerablemente, alcanzando valores de 1.8 segundos (Figura 9).



**Fig. 9.** Medidas para el modelo probabilístico con retroalimentación, según la cantidad de documentos devueltos

Los valores de Fallout no cambiaron mucho durante los experimentos, aunque es necesario destacar que fueron extremadamente bajos en el caso del modelo booleano clásico.

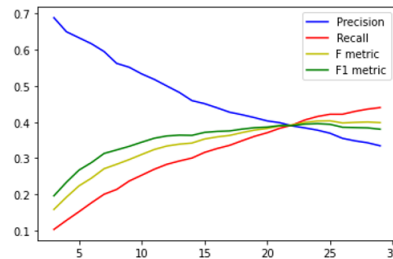
La figura 10 contiene los valores obtenidos durante la experimentación con los distintos modelos. Para esto se utilizó un total de documentos recuperados (parámetro *size* de las consultas) de 7.

	Model	Precision	Recall	F metric	F1 metric	Fallout	Time	Medium Average Precision
0	Boolean Model	0.687556	0.694815	0.987917	0.986958	0.000114	0.079754	NaN
1	Boolean Model (relaxed)	0.233651	0.242025	0.282397	0.278646	0.003853	0.253110	0.246722
2	Vectorial Model (classic)	0.375873	0.386385	0.383725	0.378882	0.003137	0.028871	0.412955
3	Vectorial Model (with retroalimention)	0.499048	0.506332	0.492632	0.487576	0.002517	0.333341	NaN
4	Probabilistic Model (classic)	0.301481	0.283543	0.325280	0.325584	0.003011	0.093682	0.306648
5	Probabilistic Model (with retroalimention)	0.262222	0.278114	0.332524	0.327048	0.003710	1.726148	NaN

**Fig. 10.** Medidas de evaluación para los distintos modelos

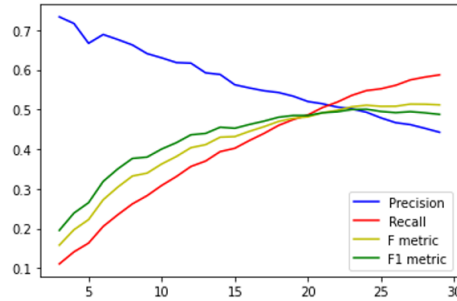
## 6.2 Conjunto de datos Med

El modelo booleano obtuvo las respuestas en un total de 0.05 segundos, las medidas obtenidas fueron de 0.09 en precisión, 0.02 en recobrado, 0.24 en la medida F, 0.28 en F1 y 6.70 en el Fallout. En este caso, la versión relajada obtuvo mejores resultados, la precisión alcanzó valores de 0.38, recobrado en 0.40, F en 0.39, F1 en 0.39 y Fallout en 0.01; todo esto se obtuvo en un tiempo de 0.18 segundos. Los gráficos correspondientes a estas medidas con distintos tamaños para documentos recuperados pueden apreciarse en la Figura 11.



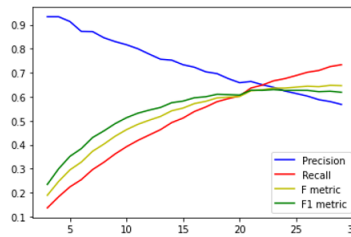
**Fig. 11.** Medidas para el modelo booleano relajado con el conjunto de datos Med. Tamaño de la cantidad de documentos recuperados contra puntuación obtenida por las medidas de evaluación

El desempeño del modelo vectorial fue nuevamente superior. En cuanto al tiempo dio su respuesta en 0.02 segundos, la precisión de 0.5, el recobrado 0.51, los valores de F y F1 obtuvieron 0.49, el Fallout 0.01 y el map 0.51.



**Fig. 12.** Medidas para el modelo vectorial sin retroalimentación, según la cantidad de documentos devueltos.

Para el modelo vectorial con retroalimentación se lograron mejorar los resultados obteniéndose una precisión y un recobrado de aproximadamente 0.65, las métricas de F y F1 obtuvieron 0.62, en un total de 0.25 segundos.



**Fig. 13.** Medidas para el modelo vectorial con retroalimentación, según la cantidad de documentos devueltos.

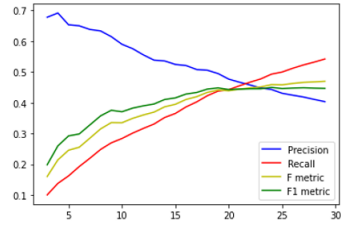
En el modelo probabilístico sin retroalimentación, los valores de precisión y recobrado alcanzaron 0.46, las medidas F un aproximado de 0.44 y 0.07 segundos.

Al añadir retroalimentación se obtuvieron resultados ligeramente mejores : la precisión, recobrado, F y F1 obtuvieron valores de 0.48, para un total de 0.66 segundos.

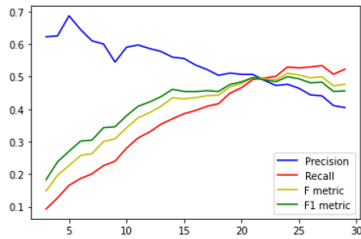
La figura 16 contiene un resumen de los valores obtenidos durante la experimentación con los distintos modelos. Para esto se utilizó un total de documentos recuperados (parámetro *size* de las consultas) de 22.

### 6.3 Comparación

Los resultados obtenidos fueron superiores en el caso del conjunto Med, esto se debe a que la anotación de los documentos relevantes a cada consulta era superior, es decir, el tamaño del conjunto de documentos relevantes es mayor.



**Fig. 14.** Medidas para el modelo probabilístico sin retroalimentación, según la cantidad de documentos devueltos



**Fig. 15.** Medidas para el modelo probabilístico con retroalimentación, según la cantidad de documentos devueltos

Model	Precision	Recall	F metric	F1 metric	Fallout	Time	Medium Average Precision
0 Boolean Model	0.095556	0.018561	0.243526	0.289725	0.000067	0.059382	NaN
1 Boolean Model (relaxed)	0.389394	0.391778	0.392189	0.390509	0.013289	0.186724	0.379562
2 Vectorial Model (classic)	0.506061	0.518464	0.497759	0.494212	0.010746	0.020193	0.519701
3 Vectorial Model (with retroalimentation)	0.650000	0.648388	0.628330	0.626595	0.007598	0.265164	NaN
4 Probabilistic Model (classic)	0.457576	0.466805	0.448182	0.445202	0.011802	0.075544	0.466980
5 Probabilistic Model (with retroalimentation)	0.448485	0.455712	0.453588	0.450774	0.011999	1.387827	NaN

**Fig. 16.** Medidas de evaluación para los distintos modelos con el conjunto Med

Esto permite que los modelos tengan mayor margen de error cuando seleccionan documentos relevantes lo que provoca que la evaluación de las métricas aumente considerablemente.

El modelo booleano tuvo comportamientos diferentes en cada conjunto de datos. En el Cranfield obtuvo mejores resultados debido a que la relevancia de los documentos podía clasificarse de más fuerte a más débil, donde una relevancia fuerte indicaba una correspondencia casi total entre la consulta y el documento. Esto provocó que en este conjunto el booleano obtuviera buenos resultados. En el caso del conjunto Med solo anota la relevancia o no, sin ninguna escala por lo que el booleano, al no tener correspondencia total, obtuvo peores medidas.

El vectorial obtuvo los mejores resultados en ambos conjuntos, mejorando los valores de sus métricas cuando se aplicaba retroalimentación. En el segundo conjunto de datos aumentaron sus resultados hasta alcanzar valores de 0.69.

El modelo probabilístico en el conjunto Cranfield, obtuvo valores bajos en cuanto a sus métricas, los cuales empeoraron cuando se aplicaba la retroalimentación. En el segundo conjunto de prueba, se mejoraron las métricas y el comportamiento mejoró cuando se utilizaron las técnicas de retroalimentación. Esto se debe nuevamente a la diferencia y calidad de las anotaciones en los conjuntos.

## 7 Conclusiones

La pregunta de cuál modelo es el mejor no puede responderse sin tener en cuenta el contexto en el que se trabaja. En este caso se pudo apreciar que los resultados dependían en gran cantidad de las anotaciones del conjunto de datos y la correspondencia de relevancia de los documentos a las consultas.

En la práctica el modelo Booleano fue el más susceptible a los cambios en el conjunto de prueba para las evaluaciones y comparación realizadas. En cuanto a su comportamiento en la aplicación, pudimos apreciar que el modelo relajado, lograba un ranking entre los documentos según las métricas de similitud, es un comportamiento similar al modelo vectorial pero sin otorgarle pesos a la "importancia" de las palabras. Este tipo de modelo pudiera emplearse en sistemas de dominio más específico.

El modelo probabilístico no obtuvo resultados destacables en la pruebas realizadas, su comportamiento con pseudo-retroalimentación mejoraba con un conjunto de datos mejor anotados, como el caso del Med, o con una mayor cantidad de iteraciones. Al aumentar el número de pasos de la retroalimentación aumentaba considerablemente el tiempo de la consulta lo que no resulta conveniente para las necesidades de los usuarios.

Los experimentos realizados nos sugieren que dentro de los modelos clásico el modelo vectorial es el más adecuado, obteniendo resultados considerablemente superiores a los otros dos modelos analizados, además, el método de retroalimentación implementado mejora considerablemente la calidad de las respuestas obtenidas.

## Referencias

- [1] Ricardo Baeza-Yates, Berthier Ribeiro-Neto, et al. *Modern information retrieval*. Vol. 463. ACM press New York, 1999.
- [2] “Introduction to Information Processing Systems”. In: *Information Retrieval Systems: Theory and Implementation*. Boston, MA: Springer US, 1997, pp. 1–23. ISBN: 978-0-585-32090-8. DOI: [10.1007/978-0-585-32090-8\\_1](https://doi.org/10.1007/978-0-585-32090-8_1). URL: [https://doi.org/10.1007/978-0-585-32090-8\\_1](https://doi.org/10.1007/978-0-585-32090-8_1).
- [3] Jiménez E. Pinto D. Rosso P. “A Penalisation-Based Ranking Approach for the Mixed Monolingual task of WebCLEF 2006”. In: *WebCLEF* (2006).
- [4] Wikipedia contributors. *Cranfield experiments* — *Wikipedia, The Free Encyclopedia*. [https://en.wikipedia.org/w/index.php?title=Cranfield\\_experiments&oldid=1061646899](https://en.wikipedia.org/w/index.php?title=Cranfield_experiments&oldid=1061646899). [Online; accessed 14-June-2022]. 2021.
- [5] Wikipedia contributors. *IMDb* — *Wikipedia, The Free Encyclopedia*. <https://en.wikipedia.org/w/index.php?title=IMDb&oldid=1090928283>. [Online; accessed 14-June-2022]. 2022.