Recuperación de Información

Modelos de Recuperación de Información (y evaluación)

Normand Agustín

1. Utilizando la colección provista por el equipo docente , cuya estructura es la siguiente:

vocabulary.txt → [id termino, idf, término]
documentVectors.txt → [id doc, lista(id terminos)]
queries.txt → [id query, lista(id terminos)]
relevants.txt → [id query, listarelevantes (id doc)]
informationNeeds.txt → [id in, texto libre]

a) Calcule los conjuntos de respuestas usando el modelo booleano y el modelo vectorial (asuma en todos los casos T F = 1).
b) Compare los resultados contra los relevantes y trate de explicar las diferencias.
c) Usando las necesidades de información reescriba los 5 queries y repita la operación.
d) Indique si pudo mejorar la eficiencia a partir de las nuevas consultas.

a)
Dado que no está especificado en la consigna, uní los términos de cada una de las querys con "or". Por otro lado, los resultados no fueron ordenados, para no realizar un ranking subjetivo, dado que el modelo booleano no posibilita realizar un ranking de documentos recuperados.

Un ejemplo de ranking subjetivo sería, poner primero los que cumplen todas las condiciones, es decir, la intersección de todos los conjuntos, luego el conjunto de los documentos que correspondan al primer término de la query, luego al segundo, etc.

Conjunto de respuestas modelo booleano:
- Query 1: 5, 15, 17, 25, 29, 32, 36
- Query 2: 1, 4, 6, 8, 9, 10, 14, 18, 19, 21, 29, 32, 36, 37, 38
- Query 3: 2, 8, 10, 11, 14, 18, 19, 20, 22, 32, 33, 34
- Query 4: 7, 8, 10, 13, 18, 27, 30, 32, 35
- Query 5: 8, 10, 18, 32

Por otro lado, el modelo vectorial si nos brinda la posibilidad de ordenar por relevancia los documentos, es decir, crear un ranking.

Conjunto de respuestas usando modelo vectorial:
- Query 1: 36, 17, 25, 29, 32, 15, 5
- Query 2: 8, 32, 10, 18, 4, 21, 9, 38, 37, 6, 14, 19, 29, 36, 1
- Query 3: 32, 22, 8, 10, 34, 18, 14, 11, 20, 33, 19, 2
- Query 4: 8, 32, 10, 18, 35, 27, 30, 7, 13
- Query 5: 8, 32, 10, 18

b)
Conjunto de respuestas modelo booleano:
- Query 1:
  Recuperados: **5**, **15**, 17, **25**, 29, 32, **36**
  Relevantes: **5**, **15**, **25**, **36**

En este caso, se recuperan todos los relevantes, recall de 1, pero hay documentos recuperados que no son relevantes, posiblemente, estos contenían los términos de la query, pero para la necesidad de información del usuario, a pesar de esto, no eran importantes.
Eliminar la palabra "LIVER-CIRRHOSIS" de la query, mejoraría la precisión y no empeoraría el recall.

- Query 2:
  Recuperados: **1**, **4**, **6**, 8, **9**, 10, **14**, 18, **19**, **21**, **29**, **32**, **36**, **37**, **38**
  Relevantes: **1, 4, 6**, **9**, **14**, 17, **19**, **21**, 23, **29**, **36**, **37**, **38**, 147, 195, 196

  En este caso, no tenemos recall de 1, no se recuperaron todos los documentos que eran relevantes para el usuario. Esto sucede porque los documentos (17, 23, 147, 195 y 196) no contienen ninguno de los términos de la query, por este motivo, no son recuperados.
  Por otro lado, los que son recuperados y no son relevantes (8, 10, 18) contienen alguno o todos los términos de la query, pero a pesar de esto, no son documentos valiosos.
  Eliminar la palabra "PATIENTS" no empeoraría el recall, y mejoraría la precisión.
  El término 142, "PANCREATIC-JUICE", no está en ninguno de los documentos.

- Query 3:
  Recuperados: **2**, 8, 10, **11**, 14, 18, 19, **20**, **22**, 32, **33**, **34**
  Relevantes: **2**, 3, **11**, 12, 13, 16, **20**, **22**, 24, 26, 28, **33**, **34**

  Caso similar al de la query 1, recall de 1.
  Por otro lado, la palabra "PATIENTS" en la query, provoca que se recuperen los documentos 8, 10, 18 y 32, que ninguno es relevante. Eliminar esta palabra de la query mejoraría la precisión.
  Otra palabra que no aporta documentos relevantes es "DIET", agrega los documentos 14, 19 y 32 al conjunto de recuperados, ninguno relevante. No es el término de la query cuyo idf es el menor.
  El término 180 "SWEATING", no está en ninguno de los documentos.

- Query 4:
  Recuperados: **7**, 8, 10, 13, 18, **27**, **30**, 32, **35**
  Relevantes: **7**, **27**, **30**, 31, **35**

  En este caso tenemos un recall de 0.8, y precisión de 0.44. Eliminar la palabra "PATIENTS" de la query, mejoraría la precisión y no empeoraría el recall.

- Query 5:
  Recuperados: **8, 32, 10, 18**
  Relevantes: **8, 10, 18, 32**

  Recall y precisión excelente. Es la única query que hubiera retornado un valor distinto de nulo, en caso de unir con and los términos de la query. Es decir, cada uno

de los documentos retornados, tienen todos los términos de la query, únicamente pasa en este caso, y no pasa en ninguno de los anteriores. En las otras consultas, ningún documento contiene todos los términos de la query.

Conjunto de respuestas modelo vectorial:

- Query 1:
  Recuperados: **36**, 17, **25**, 29, 32, **15**, **5**
  Relevantes: **5**, **15**, **25**, **36**

  El término 117 presente en el documento 17, ya determinamos en el modelo booleano que no es un término que aporte valor en este caso. Pero como este posee un IDF alto (5,2), tiene una buena posición en el ranking. Esto podría mejorarse si tuviéramos el TF de cada uno de los términos, para que el IDF no sea lo único que determine la posición en el ranking.
  Los documentos 29 y 32, que también contrarrestan a que la precisión at 4 no sea de 1, son recuperados debido al término "VITAMIN-A". Es un término con IDF bajo (2,9), que de los 4 documentos que son recuperados debido a este, 2 son relevantes y 2 no.

- Query 2:
  Recuperados: 8, 32, 10, 18, **4**, **21**, 9, **38**, **37**, **6**, **14**, **19**, **29**, **36**, **1**
  Relevantes: **1**, **4**, **6**, **9**, **14**, 17, **19**, **21**, 23, **29**, **36**, **37**, **38**, 147, 195, 196

- Query 3:
  Recuperados: 32, **22**, 8, 10, **34**, 18, 14, **11**, **20**, **33**, 19, **2**
  Relevantes: **2**, 3, **11**, 12, 13, 16, **20**, **22**, 24, 26, 28, **33**, **34**

- Query 4:
  Recuperados: 8, 32, 10, 18, **35**, **27**, **30**, **7**, 13
  Relevantes: **7**, **27**, **30**, 31, **35**

- Query 5:
  Recuperados: **8, 32, 10, 18**
  Relevantes: **8, 10, 18, 32**

Salvo en la query 5, el ranking no es muy bueno, los resultados relevantes suelen estar últimos o muy dispersos. Hay bastantes casos donde los documentos relevantes no contienen ninguno de los términos de la query, eso empeora los resultados, sin embargo, el recall es bueno en todos los resultados.

b) y c)
Para estas consignas realicé un script, para no volver a hacer los cálculos manualmente.

Query 1:
Para la necesidad de información: "What is the association between liver disease (cirrhosis) and vitamin A metabolism in CF?"
Los términos que resultan intuitivos agregar a la query son:

- ENERGY-METABOLIS = 72

- LIVER-CIRRHOSIS = 117
- VITAMIN-A = 191

Pero en realidad se obtiene mayor precisión sacando el término LIVER-CIRRHOSIS, sin perder recall.
Esto lo sé debido haber resuelto el ejercicio y notar que dicho término no suma ningún documento relevante al conjunto, por lo que empeora la precisión. Sin embargo, para un usuario final, sería imposible detectar que sacando uno de solo 3 términos que tiene la query, va a mejorar los resultados. Es decir no sería una query que pueda formular un usuario final fácilmente, sino alguien que conoce los vectores de documentos, los documentos relevantes, y puede armar la query más eficiente con mayores herramientas.

Por ejemplo, agregar el término CYSTIC-FIBROSIS sería una mejora válida, que podría hacer un usuario final, dado que está en la necesidad de información y no en la consulta. Pero no mejora las métricas debido a que tiene un IDF de 0.

Query 2:
Para la necesidad de información: "What is the role of Vitamin E in the therapy of patients with CF?", utilicé los términos:

- VITAMIN-E = 195
- VITAMIN-E-DEFICI = 196
- VITAMINS = 198
- CYSTIC-FIBROSIS (CF) = 51

Con estos términos se logra una precisión de 0,92 y un recall de 0.92. Aunque el término con id 51 no aporta valor dado que tiene un IDF de 0.

Query 3:
De todos los términos de la query, dejando solamente PANCREATIC-EXTRA, se obtiene el mismo recall y una precisión de 1.

Query 4:
Usando el término SWEAT se obtiene recall y precisión de 0.8

Query 5:
Usando el término TASTE-DISORDERS se obtiene recall y precisión de 1.

2. Dados los siguientes documentos, arme la matriz término-documento (TD)

- **Doc 1 = {El software libre ha tenido un papel fundamental en el crecimiento de Internet. Además, Internet ha favorecido la comunicación entre los desarrolladores de software.}**
- **Doc 2 = {La mayor riqueza que tiene un país es la cultura, eso lo hace más libre.}**
- **Doc 3 = {La producción de software es fundamental para nuestro país, como así también lo es la producción de tecnología de hardware y comunicación}**
- **Doc 4 = {La cultura del software libre está en crecimiento. Es fundamental que nuestro país incorpore software libre en el estado.}**

¿Qué documentos se recuperan en cada caso para las siguientes consultas booleanas? (Muestre mediante operaciones con conjuntos cómo se resuelven las consultas)
a) (not software) or (pais and fundamental)
b) producción and (cultura or libre)
c) fundamental or libre or país

a)
Condicion = Conjunto de documentos que la cumplen

Not Software = {d2}
País = {d1}
Fundamental = {d1,d3,d4}
País and Fundamental = {d1} ∩ {d1,d3,d4} = {d1}
(not software) or (pais and fundamental) = {d2} ∪ {d1} = {d2, d1}

Documentos recuperados = d1 y d2.

b)
Producción = {d3}
Cultura = {d2,d4}
Libre = {d1,d2,d4}
Cultura or Libre = {d2,d4} ∪ {d1,d2,d4} = {d1,d2,d4}
Producción and (cultura or libre) = {d3} ∩ {d1,d2,d4} = {}

Documentos recuperados = Ninguno.

c)
Fundamental = {d1,d3,d4}
Libre = {d1,d2,d4}
País = {d2,d3,d4}

Fundamental or Libre or Pais = {d1,d3,d4} ∪ {d1,d2,d4} ∪ {d2,d3,d4} = {d1,d2,d3,d4}

Documentos recuperados = d1, d2, d3, d4. Todos.

3. Utilizando los documentos del ejercicio anterior arme la matriz TD pero calculando wij como la frecuencia del i-ésimo término en el j-ésimo documento. Calcule el ranking para la siguientes consultas utilizando como métrica el producto escalar y luego repita con la métrica del coseno.

a) software
b) país libre
c) producción software país

Producto Escalar con TF					
a) software		b) país libre		c) producción software país	
Ranking	Score	Ranking	Score	Ranking	Score
d1	2	d4	3	d3	4
d4	2	d2	2	d4	3
d3	1	d1	1	d1	2
d2	0	d3	1	d2	1

Coseno con TF					
a) software		b) país libre		c) producción software país	
Ranking	Score	Ranking	Score	Ranking	Score
d4	0,5547001962	d2	0,7071067812	d3	0,7302967433
d1	0,5163977795	d4	0,5883484054	d4	0,4803844614
d3	0,316227766	d3	0,2236067977	d1	0,298142397
d2	0	d1	0,1825741858	d2	0,2886751346

Realizando una comparación, el ranking utilizando como métrica el producto escalar, tanto para la query "a" como para la "b", hay documentos cuyo score es el mismo, por lo que no se puede definir cual es más relevante que el otro. Sin embargo, utilizando la métrica del coseno, si fué posible diferenciar la relevancia de estos documentos. Esto explica la diferencia en el orden de los documentos en los dos primeros rankings.

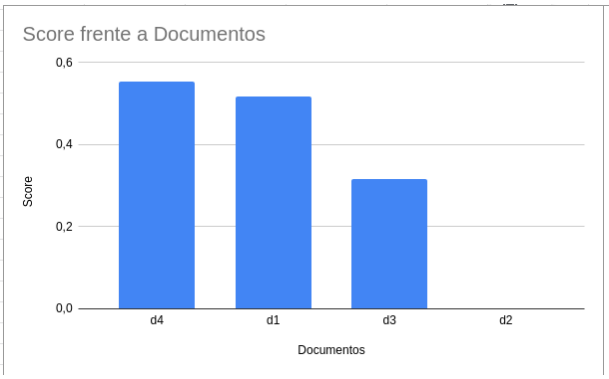
Es decir, para la consulta a) con producto escalar con TF, no era posible diferenciar qué documento era más relevante entre el 1 y el 4, porque ambos tenían score 2, los dos incluyen el término "Software" 2 veces.
Incorporar la longitud normalizada tanto del query como de los documentos, provocó que se penalice el score del documento 1, dado que este es el documento de mayor longitud, y que se penalice ligeramente menos el score del documento 4, dado que este es el segundo más largo. Por este motivo, este último documento pasó a ser el más relevante. Permitiendo darle un orden al ranking, libre de subjetividades. Algo muy similar a esto sucedió en el ranking para la consulta "b" con los documentos 2 (muy corto) y 4 (largo).

Por otro lado, el orden de los resultados de los rankings para la consulta "c", son exactamente iguales.

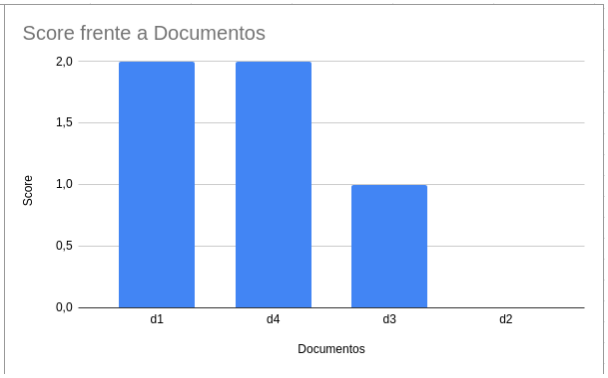
Comparación de histogramas. Coseno y Producto Escalar ambos con TF.

a)

Ranking a) coseno con TF.

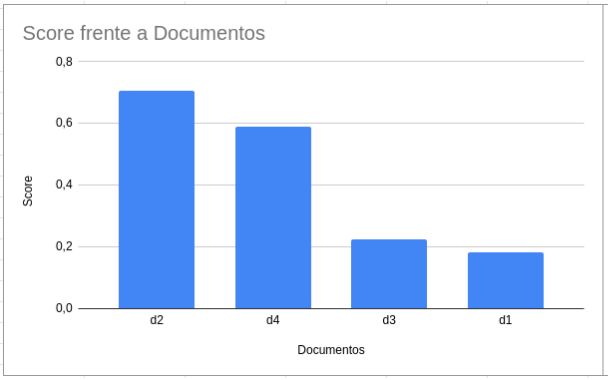


Ranking a) producto escalar con TF.

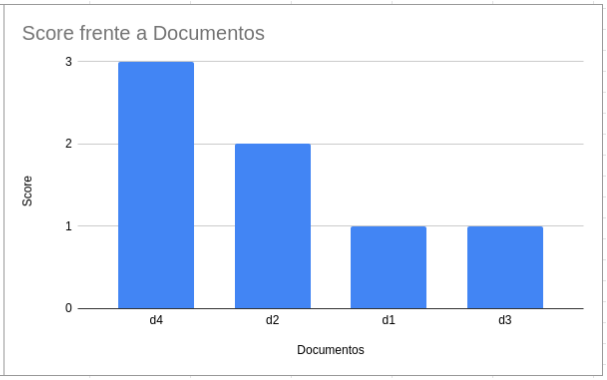


b)

Ranking b) coseno con TF.

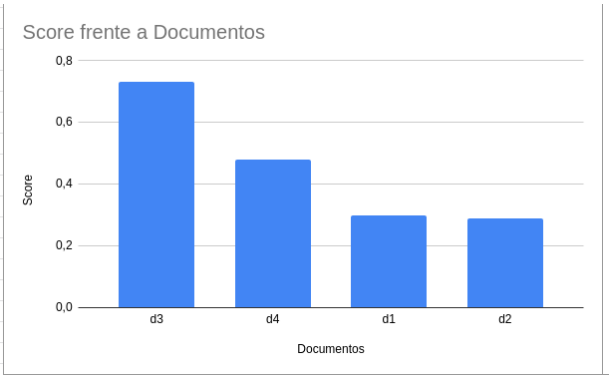


Ranking b) producto escalar con TF.

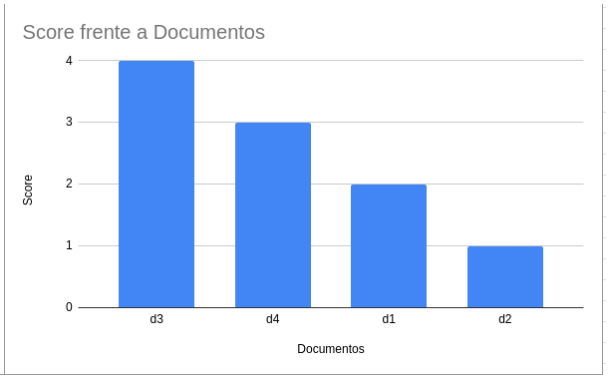


c)

Ranking c) coseno con TF.



Ranking c) producto escalar con TF.



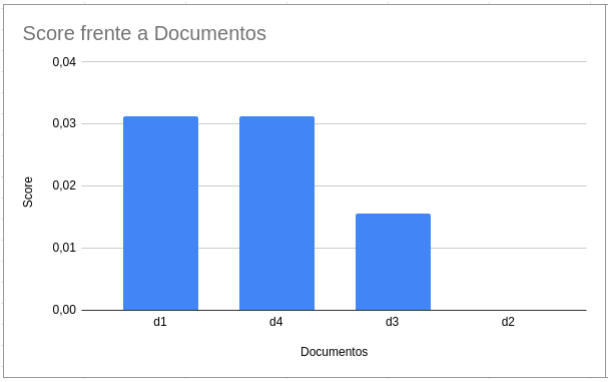
Gráficamente se puede ver mejor lo dicho anteriormente, la métrica del coseno permite mayor granularidad a la hora de determinar la relevancia de un documento.
Si bien son documentos cortos, se puede ver como la métrica del coseno penaliza a los que son más largos, y beneficia a los de longitud menor.

4. Rearme la matriz del ejercicio anterior pero calcule los pesos de acuerdo a TF ∗ IDF . Repita todas las consultas (por ambas métricas). ¿Puede obtener alguna conclusión?

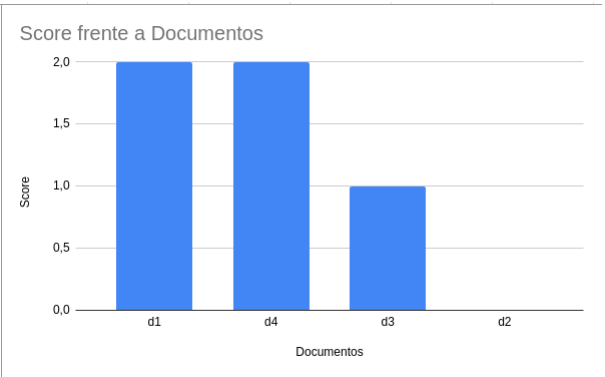
Producto Escalar con TFxIDF					
a) software		b) país libre		c) producción software país	
Ranking	Score	Ranking	Score	Ranking	Score
d1	0,03121937581	d4	0,04682906372	d3	0,7561718421
d4	0,03121937581	d2	0,03121937581	d4	0,04682906372
d3	0,01560968791	d1	0,01560968791	d1	0,03121937581
d2	0	d3	0,01560968791	d2	0,01560968791

Comparación de histogramas. Producto Escalar con TF vs Producto Escalar con TFxIDF

Ranking a) producto escalar con TFxIDF.

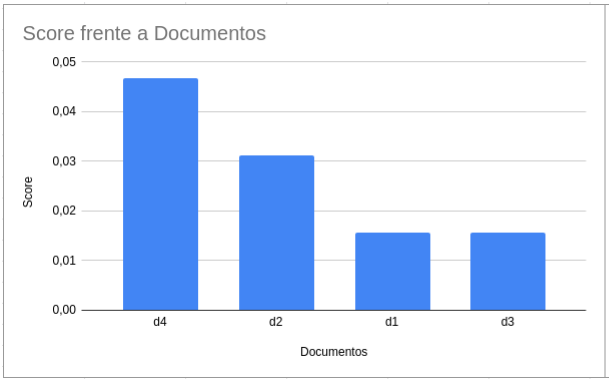


Ranking a) producto escalar con TF.

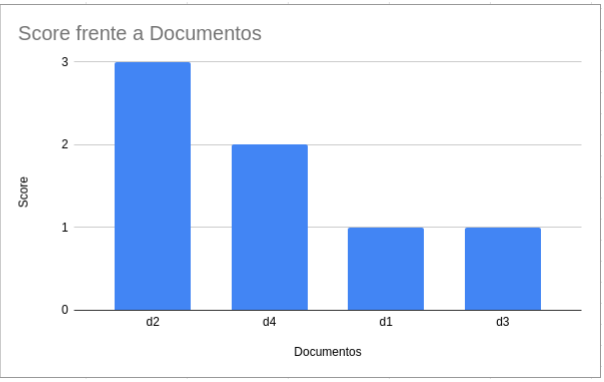


Si bien parecía que incorporar el IDF iba a evitar que esto suceda, tenemos 2 documentos con el mismo score. Dado que se trata del mismo término, los vectores de los documentos son iguales.

Ranking b) producto escalar con TFxIDF.



Ranking b) producto escalar con TF.

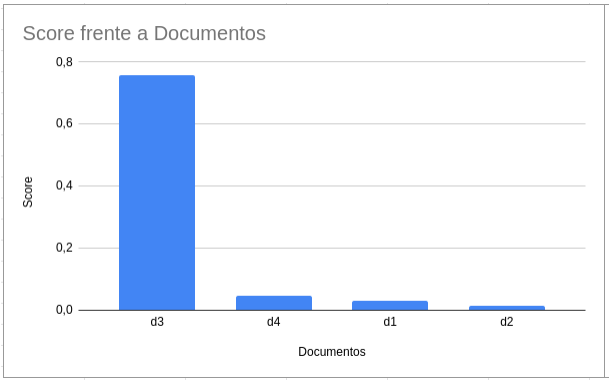


Volvemos a tener dos documentos con el mismo score, si bien el documento 1 tiene solo el término "Libre" y el d2 tiene solo el término "Pais", vuelven a tener el mismo score porque el IDF de estos términos es el mismo.

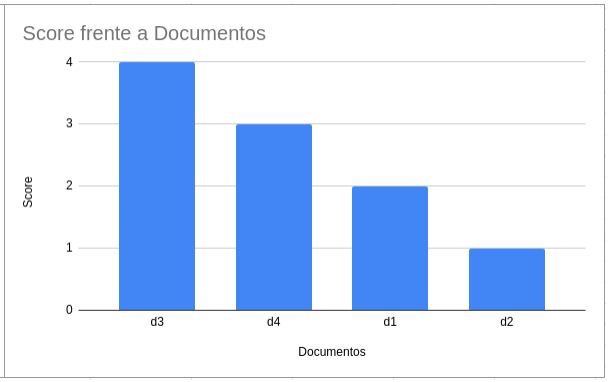
Es decir, incorporar como peso TFxIDF a diferencia de TF, me puede ayudar a diferenciar la relevancia de dos documentos, cuyos términos son distintos pero la cantidad total de términos es la misma, solo si estos tienen IDF distintos.

En el ranking c) hay una diferencia significativa en los histogramas.

Ranking c) producto escalar con TFxIDF.



Ranking c) producto escalar con TF.



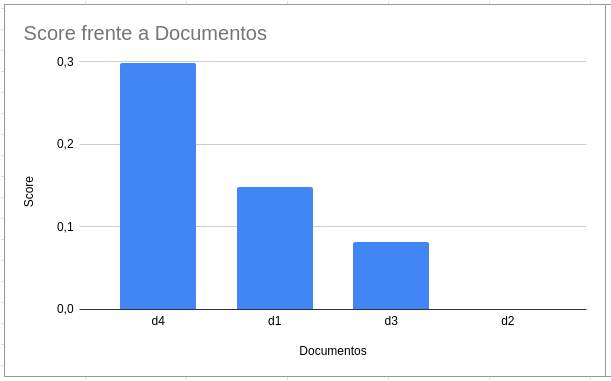
Esto sucede porque de los 3 términos que tiene la query (Producción,Software,Pais), Pais y Software tienen un IDF bajo, 0,12, y si bien el documento 3 los contiene a ambos, son términos que también están presentes en los demás documentos.
Sin embargo, Producción posee un IDF alto, 0,60 y el documento 3 es el único que contiene este término. Por este motivo, al incluir el IDF a la métrica del producto escalar, se ve una valoración significativamente mayor en el documento 3.

Coseno con TFxIDF					
a) software		b) país libre		c) producción software país	
Ranking	Score	Ranking	Score	Ranking	Score
d4	0,2987009834	d4	0,3168202363	d3	0,7925334618
d1	0,1490052933	d2	0,2538915793	d4	0,08921640843
d3	0,08216266207	d3	0,05809777551	d2	0,0357478662
d2	0	d1	0,05268132668	d1	0,02967006661

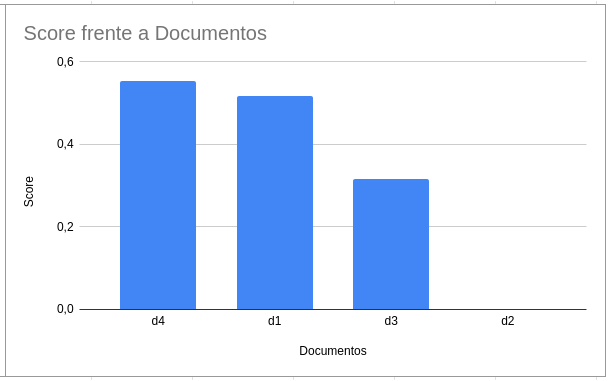
(Continúa en la otra página)

Comparación de histogramas. Coseno con TF vs Coseno con TFxIDF

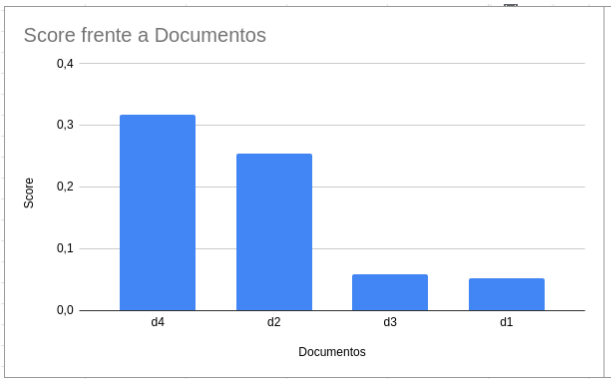
Ranking a) coseno con TFxIDF.



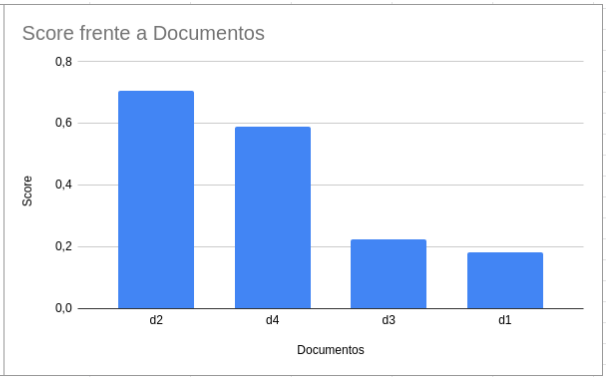
Ranking a) coseno con TF.



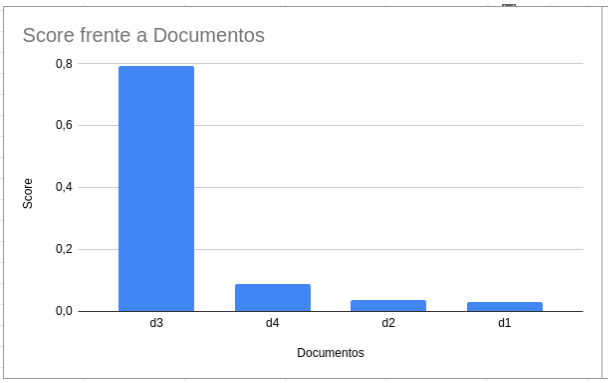
Ranking b) coseno con TFxIDF.



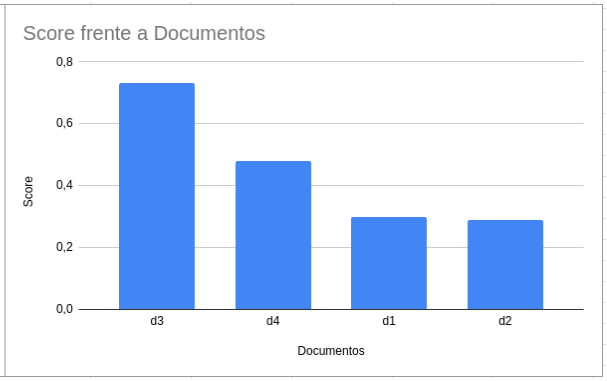
Ranking b) coseno con TF.



Ranking c) coseno con TFxIDF.



Ranking c) coseno con TF.



A modo de conclusión, podemos ver que cuantas más herramientas incorporamos al modelo, mayor capacidad tenemos para determinar la relevancia de los documentos.

- Si no usamos TFxIDF como peso, no valoramos de forma correcta los términos poco comunes.
- Si no usamos la métrica del coseno, no tenemos en cuenta las longitudes de los documentos, por lo que no se penalizan documentos largos.

Lo comprobamos viendo que la métrica del coseno con TFxIDF incorpora todas las ventajas que analizamos en casos anteriores, valora documentos con términos con IDF alto como el

caso de la consulta "c". Y además penaliza documentos largos y beneficia los de menor longitud, pudiendo diferenciar los casos de las consultas "a" y "b" que tenían el mismo score.

5. Utilizando Terrier indexe la colección Wiki-Small . Tome 5 necesidades de información y – de forma manual – derive una consulta (query). Para cada una, pruebe la recuperación por los modelos basados en T F ∗ IDF y BM25. ¿Cómo se comportan los rankings? Calcule el coeficiente de correlación para los primeros 10, 25 y 50 resultados. ¿Qué conclusiones obtiene?

Necesidades de información:

- INFORMATION RETRIEVAL VECTORIAL MODEL
- INDUSTRIAL REVOLUTION
- ALAN TURING
- SECOND WORLD WAR
- KUBERNETES CLUSTER

Calculé el coeficiente de correlación de Spearman para los rankings, y el resultado fue:

	Query 1	Query 2	Query 3	Query 4	Query 5
Primeros 10	61	6	0	0	0
Primeros 25	752	34	0	20	0
Primeros 50	1499	317	0	111	0

Por lo tanto, para esta colección en particular, y los procesos de tokenización, normalización, stemming aplicados por Terrier, los rankings se comportan de igual forma, tanto para el modelo BM25 como para TF_IDF, en las querys 3 y 5, de forma muy similar en las querys 2 y 4. Para la query 1, conforme avanzamos en el ranking, las diferencias van siendo cada vez mayores, es decir, los documentos con mayor score, los clasifica de forma similar, pero los que tienen scores más bajos, difieren bastante en la posición que poseen en el ranking.

6. Escriba un pequeño programa que lea un directorio con documentos de texto y arme una estructura de datos en memoria para soportar la recuperación. Luego, debe permitir ingresar un query y devolver un ranking de los documentos relevantes utilizando el modelo vectorial. Se debe soportar la ponderación de los términos de la consulta. Implemente las versiones sugeridas en MIR [1].

Para resolver este ejercicio, cree los siguientes módulos en python:

- **normalizer.py**
  Pasa a minúsculas una palabra pasada como argumento, elimina acentos, caracteres no alfanuméricos, y realiza un proceso de stemming con PorterStemmer.
- **tokenizer.py**
  Implementa la lógica necesaria para que dado un archivo html, se extraiga texto de este, extrae términos usando el **normalizer** como módulo complementario. Crea los archivos parciales de vocabulario, índice invertido y vector de documentos.

- **indexer.py**
  Recorre el directorio pasado como argumento, crea una cola para paralelizar el procesamiento de los archivos de la colección, llama a los threads que consumen de esta cola el nombre del archivo que deben procesar, y llaman al **tokenizer**.
- **merger.py**
  Módulo encargado de unificar el vocabulario, índice invertido y vector de documentos de cada uno de los threads.
- **model.py**
  Implementa la lógica necesaria para dar soporte al modelo vectorial.
- **menu.py**
  Implementa la lógica necesaria para solicitar los inputs al usuario y llamar a las funciones correspondientes.
- **exporter.py**
  Guarda los archivos en disco luego de crear el índice, para evitar tener que volver a correr el indexer en caso de cerrar el programa, la extensión de estos archivos es (.pkl). Además exporta archivos .txt para poder usar como debug.

Se implementó la ponderación de los términos de la consulta, dependiendo de la frecuencia, el peso del término en el vector de querys.

```
Ingrese la query
casa perro
Términos de la query: {'casa': 1, 'perro': 1}
Vector de query: {'casa': 7.097217943927458}, Norma de la query: 7.097217943927458
```

```
Ingrese la query
casa casa perro
Términos de la query: {'casa': 2, 'perro': 1}
Vector de query: {'casa': 12.016634551580228}, Norma de la query: 14.194435887854915
```

Nota: "Perro" no fue incluido en el vector de querys porque tiene un IDF de 0, entonces al calcular el peso del término, el resultado fué 0 también, por lo que no se agrega al vector.

Para ejecutar el programa, se debe correr el archivo, con los argumentos. menu.py <path_corpus> <0 usar indice de disco, 1 volver a indexar> <path archivo palabras vacias>.

7. Indexe la colección del ejercicio 5 con su software. Ejecute las consultas y compare los resultados con los obtenidos con Terrier. ¿Son consistentes?

Realicé un programa para traducir los doc_id usados por Terrier, a los doc_id que usé yo en mi índice, para poder comparar los resultados.

Comprobando con un script, confirmé que los documentos recuperados son, en la mayoría de los casos, los mismos, es decir, los modelos son consistentes en cuanto al recall. Pero a simple vista parecía que variaba considerablemente el ranking de estos.
Para confirmar esto último, utilicé el mismo programa del ejercicio 5, calculé el coeficiente de correlación de Spearman, y los resultados fueron, en la mayoría de los casos, superiores a 8.000, es decir, definitivamente los programas rankean de manera completamente distinta a los mismos documentos, a pesar de estar usando ambos TF_IDF.

8. Se requiere evaluar la performance en la recuperación de un sistema. Para una consulta q1, dicho sistema entregó la siguiente salida.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	N	N	R	R	N	N	N	N	R	N	N	N	R	N

Los documentos identificados como R son los relevantes, mientras que las N's corresponden a documentos no relevantes a q1. Suponga – además – que existen en el corpus otros 6 documentos relevantes a q1 que el sistema no recuperó. A partir de esta salida calcule las siguientes medidas:
a) Recall y Precisión para cada posición j
b) Precisión promedio
c) Precisión al 50 % de Recall
d) Precisión interpolada al 50 % de Recall
e) Precisión-R
Finalmente, realice las gráficas interpolada y sin interpolar. Luego, interprete brevemente los resultados y brinde una explicación.

a)

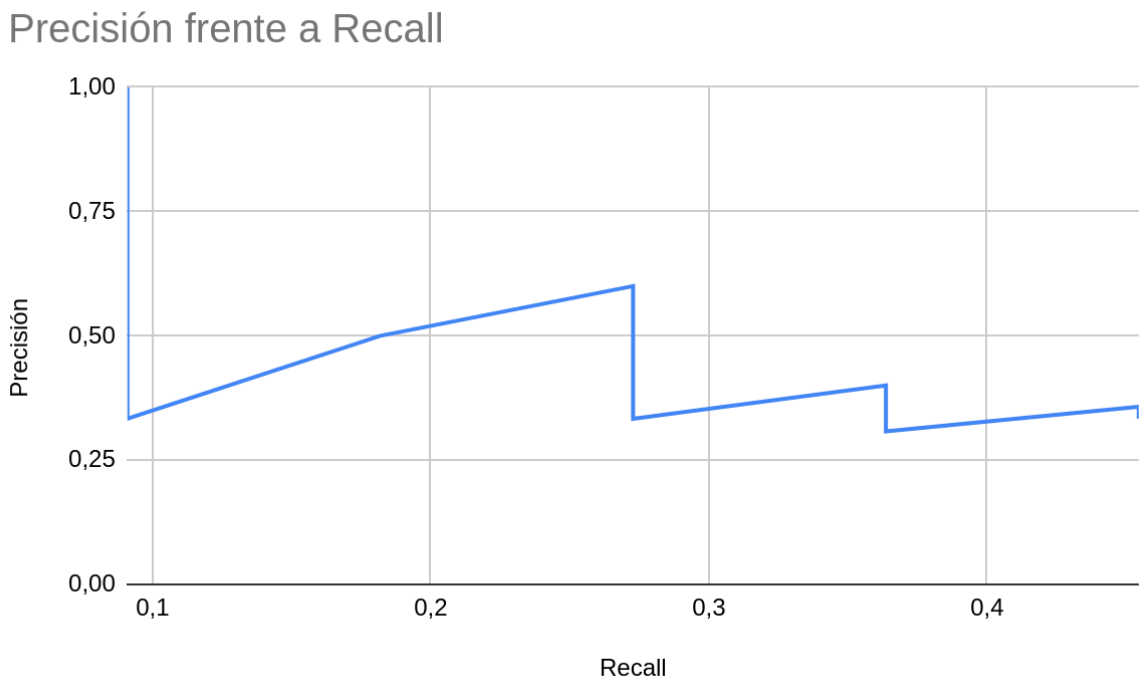
	Recall	Precisión
R	0,09090909091	1
N	0,09090909091	0,5
N	0,09090909091	0,3333333333
R	0,1818181818	0,5
R	0,2727272727	0,6
N	0,2727272727	0,5
N	0,2727272727	0,4285714286
N	0,2727272727	0,375
N	0,2727272727	0,3333333333
R	0,3636363636	0,4
N	0,3636363636	0,3636363636
N	0,3636363636	0,3333333333
N	0,3636363636	0,3076923077
R	0,4545454545	0,3571428571
N	0,4545454545	0,3333333333

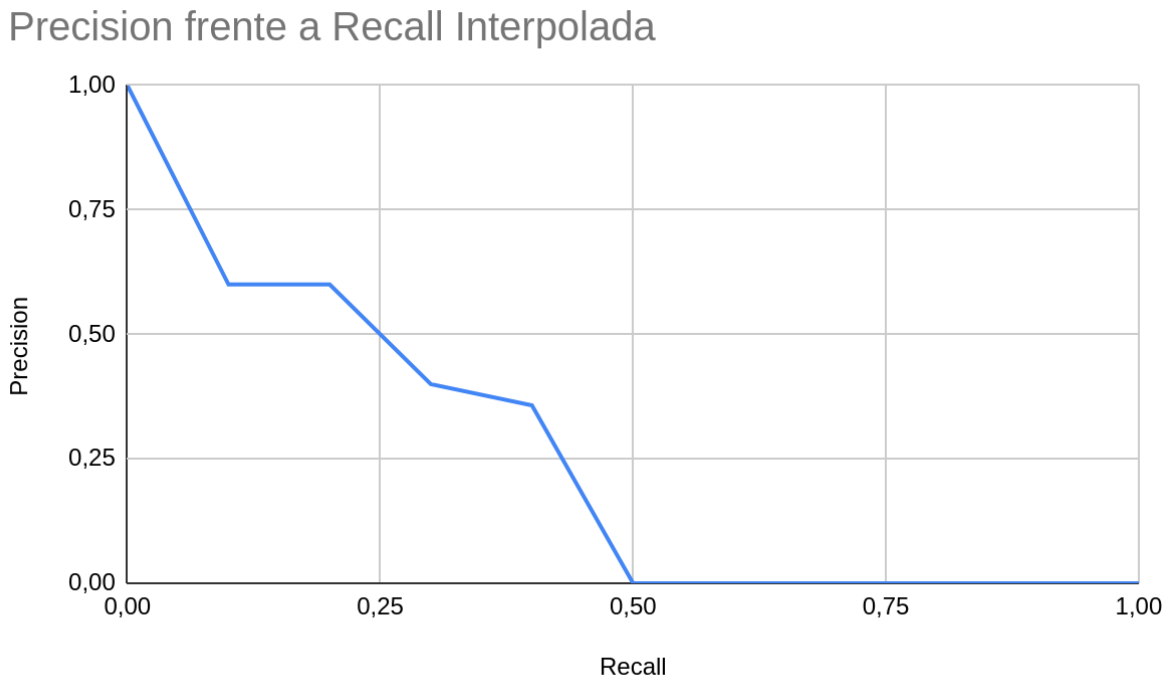
b) Precisión promedio = 0,4443584194

c) No se alcanza un 50% de recall.

d) Precisión interpolada al 50 % de Recall = 0

e) Precisión-R = 0,3636363636
Precisión en la posición R, donde R es la cantidad de relevantes para la query. P@11





9. Utilizando la colección de prueba CISI y Terrier se debe realizar la evaluación del sistema. Para ello, es necesario construir un índice con los documentos de la colección y luego ejecutar las consultas, las cuales se deben armar a partir de los

términos que considere de las necesidades de información. Los resultados deben ser comparados contra los juicios de relevancia de la colección utilizando el software trece val6 . Realizar el análisis y escribir un reporte indicando los resultados obtenidos, junto con la gráfica de R–P en los 11 puntos standard. Realice dos experimentos: en el primero, no considere la frecuencia de los términos en el query mientras que en el segundo lo debe tener en cuenta.

En primer lugar, pre procesé la colección CISI para transformar esta al formato válido leído por Terrier.

- Pre Procesé el archivo de documentos
- Pre Procesé el archivo de relevantes. Me llamó la atención que la query 112 no posee documentos relevantes.
- Por último, derive las querys de las necesidades de información, tokenizando y luego extrayendo términos. Generé dos archivos de querys, uno que tiene términos repetidos en la query, y otro que sólo tiene términos unívocos.

Indexé la colección con Terrier, ejecuté las querys utilizando el modelo TF_IDF, y evalué los resultados, obteniendo:
Se recuperaron 64583 documentos
En total, eran relevantes 3114 pero se recuperaron 2673 de estos.

Precisión frente a recall interpolada.

Recall	Precisión
0,00	0,6615
0,10	0,4291
0,20	0,3366
0,30	0,2545
0,40	0,2042
0,50	0,1718
0,60	0,1432
0,70	0,1014
0,80	0,0711
0,90	0,0368
1,00	0,0115

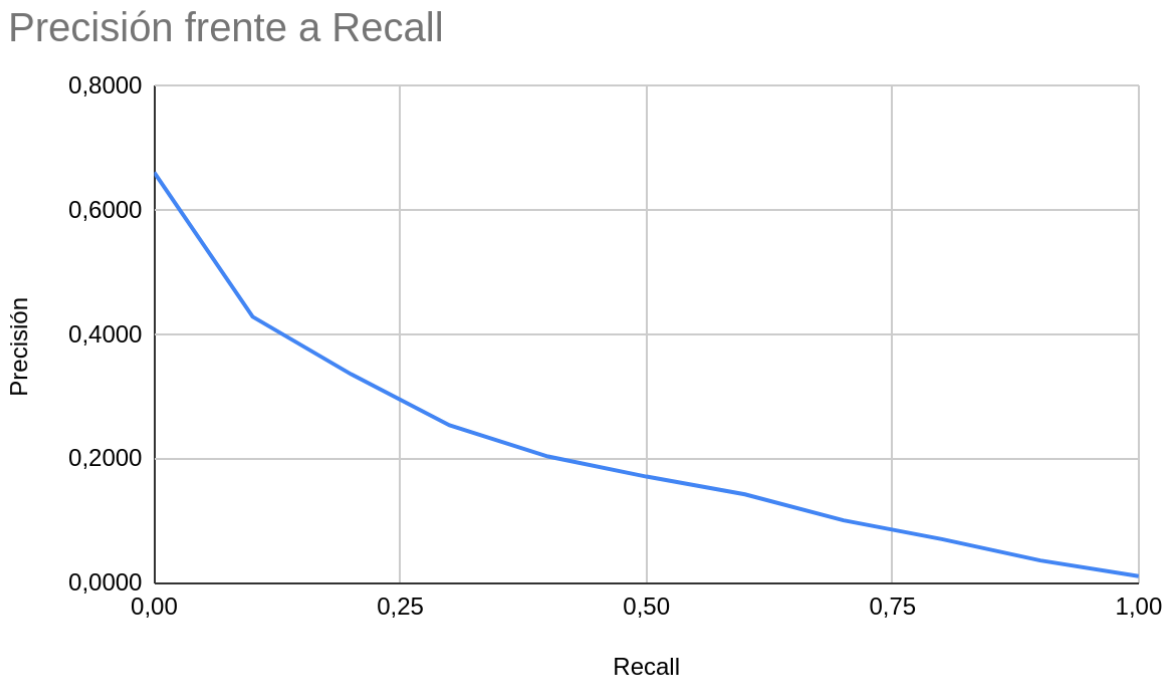
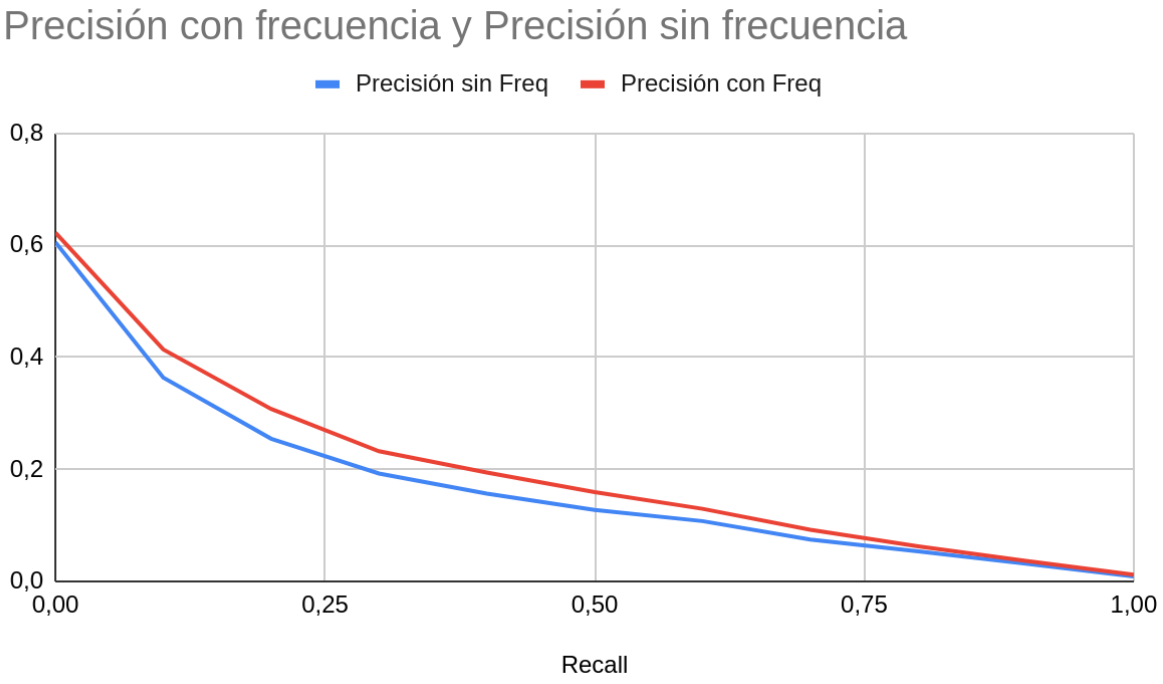


Tabla de P@X

P@X	Precisión
P_5	0,4
P_10	0,3421
P_15	0,3026
P_20	0,2664
P_30	0,225
P_100	0,1451
P_200	0,1036
P_500	0,0598
P_1000	0,0352

Modifique el programa que procesa los archivos de la colección CISI, para tener en cuenta la frecuencia, volví a correr las consultas, y para el modelo TF_IDF no hubo cambios en los resultados, mientras que para DPH, resultó ser mejor utilizar la frecuencia de los términos en la query.



10. Dadas las salidas de tres sistemas de recuperación de información para 3 consultas cualquiera y los juicios de relevancia creados por asesores humanos, calcule para cada sistema:
a) La precisión media
b) La precisión media a intervalos de Recall de 20 %
c) P@5, P@10, P@20
Luego, exponga un escenario posible y medidas complementarias para decidir qué sistema utilizar.

a)

SRI A				
		Query 1	Query 2	Query 3
Average Precision		0,724025974	0,6178571429	0,328488082

Mean Average Precision	0,5567903996

SRI B				
		Query 1	Query 2	Query 3
Average Precision		0,2291353383	0,2875505275	0,3486185154

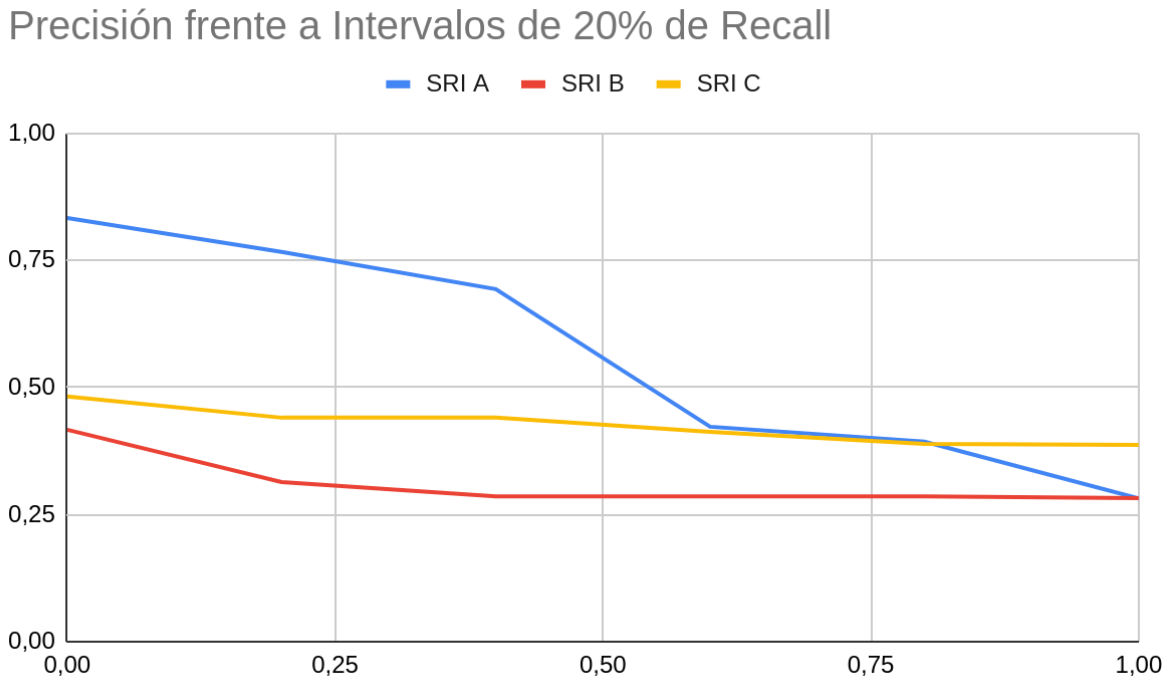
Mean Average Precision	0,2884347937

SRI C				
		Query 1	Query 2	Query 3
Average Precision		0,4952380952	0,3320516612	0,2918911289

Mean Average Precision	0,3730602951

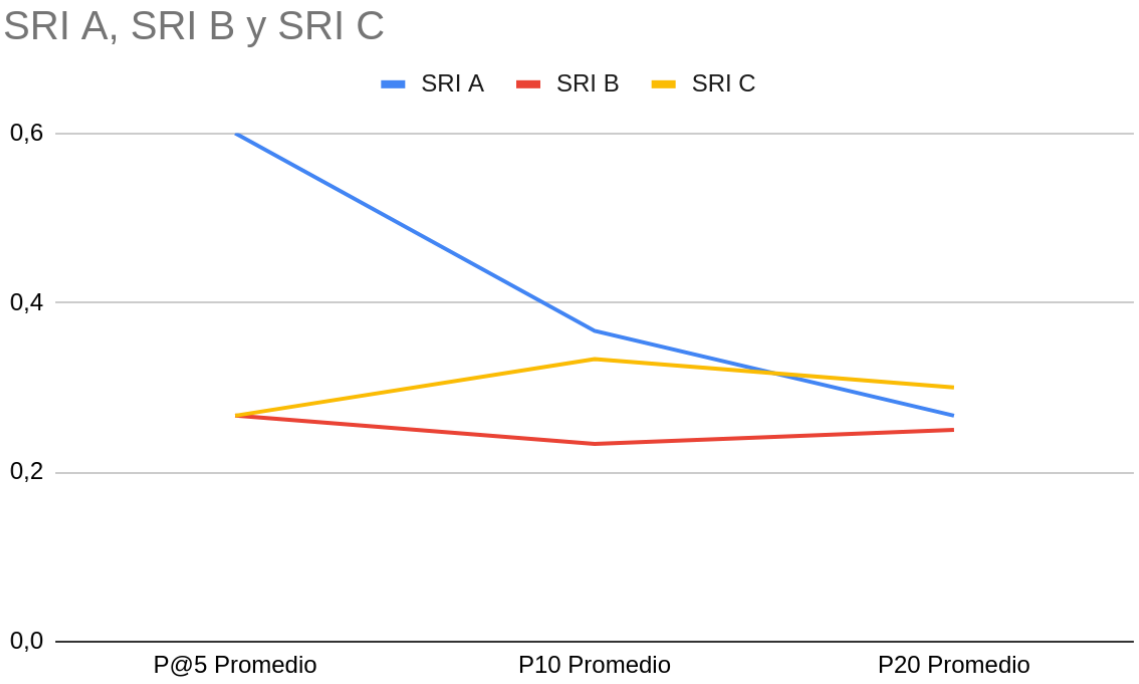
b)

Intervalos	SRI A	SRI B	SRI C
0,00	0,8333333333	0,4166666667	0,4821428571
0,20	0,7666666667	0,3138888889	0,4404761905
0,40	0,6933333333	0,2858187135	0,4404761905
0,60	0,4226984127	0,2858187135	0,4126984127
0,80	0,3933333333	0,2858187135	0,3888888889
1,00	0,2814646465	0,2820048309	0,3866959064



c)

	P@5	P10	P20
SRI A	0,6	0,3666666667	0,2666666667
SRI B	0,2666666667	0,2333333333	0,25
SRI C	0,2666666667	0,3333333333	0,3



El SRI A demuestra tener la mayor precisión para los primeros resultados, lo que sería preferible para escenarios de navegación web, ya que los 10 Blue Links, posiblemente serán resultados relevantes. Sin embargo, la precisión de este sistema cae drásticamente a medida avanzamos en el ranking, teniendo mejor desempeño el sistema C, seguido por el B.
Por lo que, en un escenario de investigación, utilizaría el sistema C, dado que la precisión se mantiene constante a lo largo de todos los intervalos de recall.