CAIM

(Laboratorio)

Práctica 3

Documentación

Curso 2021/2022 Q1

- Jesús Benítez Díaz
- Javier Rivera Hernández

UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

Índice

Desarrollo

Para el desarrollo de esta práctica implementaremos la regla de Rocchio. Tal y como nos sugerían en el enunciado hemos operado con diccionarios. Una vez calculado los TfIdF de los *k* documentos más relevantes, iremos añadiendo términos a la query inicial.

En cuanto dificultades, destacamos algunos problemas a la hora de operar con los diccionarios, ocupando la mayor parte del tiempo de la implementación. De resto no ha habido problema para entender la regla de Rocchio.

Para fusionar los vectores de los tf-idf de cada documento, estos estarán implementados con diccionarios en vez de con listas ordenadas.

Esto se debe a que con listas ordenadas para cada elemento de la lista más pequeña deberíamos hacer una búsqueda dicotómica O(log n) para encontrar el elemento en la otra lista o la posición donde se debería colocar, y O(n) para insertarlo si hiciese falta; con lo cual tenemos una complejidad de $O(n^2 \log n)$ en el peor caso.

Por contra, si están implementados con diccionarios, hace falta hacer una consulta al diccionario por cada palabra del diccionario más pequeño con coste O(n) por operación en el peor caso (O(1) en el caso medio) y lo mismo para insertarlo; por lo tanto tenemos un coste medio de O(n), $O(n^3)$ en el caso peor, que no se da a no ser que la función de hash sea muy inadecuada o que los inputs sean raros (no es el caso puesto que son strings).

Experimentación

Para la experimentación, hemos trabajado con diferentes queries, así como hemos ido modificando los parámetros α,β,k y Nrounds. Hemos trabajado con el índice de news, creado a partir de los documentos 20 news group.

Para determinar el recall y el precision de una búsqueda, puesto que en una práctica de dos semanas no nos da tiempo a implementar un sistema complejo que seleccione qué documentos son a priori relevantes para la búsqueda, lo describiremos de manera cualitativa.

2.1 Primer experimento

Nuestra primera query será "*science^2 planet*". Fijamos alpha = 0.7 y beta = 0.3.
Tendremos nrounds = 5 y sacaremos 4 palabras por query (R = 4).

Queremos observar que las querys generadas por el ‘pseudorelevance feedback’ tienen sentido observado los términos que añade están relacionados con el campo semántico de la query original. Inicialmente probamos con una k = 3:

['science^0.3676469874710612', 'planet^0.2470012631841603',
'habitable^0.15465835700001299', 'fs7^0.10441408906123485']
1 Documents

En este caso, para una k muy pequeña, no solo nos saca un único documento, si no que nos devuelve un término aleatorio del documento, el cual no tiene relevancia.

Por lo tanto, decidimos observar que sucede al aumentar la k a 9 para que tome más documentos a la hora de hacer promedio del tf-idf de los k documentos más relevantes.

['science^0.4719610899841361', 'planet^0.26422582088821056',
'habitable^0.06137133680375913', 'earth^0.016903230935515254']
1 Documents

Seguimos obteniendo un único documento, pero en este caso obtenemos una serie de términos adicionales que se adecuan bastante bien a la temática de los originales.

Por último, probamos con k = 50:

['science^0.5130020440306373', 'planet^0.2613686760456023',
'space^0.014519156990567977', 'earth^0.010465662909505449']
11 Documents

Como podemos observar, para k = 50, además de obtener unos términos coherentes, nos devuelve un gran número de textos.

Esperamos encontrar un recall bajo para el resto de experimentos, puesto que nuestro conjunto de textos a priori relevantes es muy grande y poco acotado a la búsqueda.

2.2 Segundo experimento

En este segundo experimento veremos como cambia el recall y la precisión al usar la regla de Rocchio. Trabajaremos con la query “science^2 particle”

Para la query de “science^2 particle” sin usar la regla de Rocchio nos devuelve 10 documentos,de los cuales no todos pertenecen al tema en cuestión. Esto quiere decir que

obtenemos un mayor recall, pero perdemos mucha precisión. Sin embargo, si usamos la regla de Rocchio en el experimento anterior, obtenemos 1 solo texto con temática de ciencia; por lo tanto nuestra precision aumenta y disminuye nuestro recall ya que el número de textos de ciencia devueltos es menor.

2.3 Tercer experimento

Para este tercer experimento trabajaremos con una única query, variando los parámetros y así ver cómo afecta al resultado. Partiremos de la query “hockey^2 people”.

En primer lugar partiremos de estos parámetros:

- Alpha = 0.8
- Beta = 0.2
- Nrounds = 5
- R = 4
- k = 100

2.3.1 Variando R

- R = 2
['hockey^0.592913708196543', 'people^0.2874209913907146']
86 Documents
- R = 3
['hockey^0.5739214536669183', 'people^0.2779409071340859',
'game^0.0291469324725832 7']
57 Documents
-R = 5
['hockey^0.567974479869534', 'people^0.28279058730009865',
'tie^0.005927464200544974', 'breaker^0.005517430878650706',
'game^0.005031531764031431']
2 Documents
-R = 7
['hockey^0.5651287264415449', 'people^0.28110841910179707',
'game^0.004710528187892561', 'tie^0.004081923484039208',
'breaker^0.004000360701127222', 'team^0.0036678214868839892',
'espn^0.003536840268568001']
0 Documents

Se aprecia con claridad que al aumentar el número de términos que añadimos a la query, nos devuelve menos documentos, llegando incluso a no encontrar ninguno.

Las palabras que añade son coherentes con la query inicial, eso hace que la concreción de los resultados sea mayor mientras no genere una query con demasiados términos que hace que sea imposible encontrar documentos que la cumplan por ser demasiado específica.

2.3.2 Variando Alpha y Beta

A priori esperamos que al aumentar el valor de beta, reduzca la importancia de la query inicial y prioricemos incorporar nuevos términos a la query.

-Alpha 0.9 y Beta = 0.2
['hockey^0.6208708145173463', 'people^0.3093541757310375',
'game^0.00394770820732752', 'tie^0.003580230063189515']
6 Documents
-Alpha 0.8 y Beta = 0.2
['hockey^0.5702389281392725', 'people^0.2828651549355953',
'game^0.008236116942484021', 'tie^0.0074787233797496445']
6 Documents
-Alpha 0.6 y Beta = 0.4
['hockey^0.46160967433765976', 'people^0.2258402281317076',
'game^0.018169176527205903', 'tie^0.016551933439754984']
6 Documents
-Alpha = 0.5 y Beta = 0.5
['hockey^0.4025621153837198', 'people^0.19471539297490578',
'game^0.024056950565810385', 'tie^0.0219645560191632']
6 Documents

Efectivamente nuestra hipótesis se cumple, a medida que aumentamos beta, el peso de los nuevos términos es cada vez mayor.

2.3.3 Variando nrounds

A priori creemos que aumentar el número de iteraciones hará que los valores de la query converjan a un valor que no cambiará a pesar de que se aumente el número de iteraciones.

Obtuvimos los siguientes resultados:

-nrounds = 2:

['hockey^0.7870795457116107', 'people^0.3917495091502884',
'game^0.00638726822220063', 'tie^0.005534418702354685']
81 Documents
-nround = 5:
['hockey^0.5702389281392725', 'people^0.2828651549355953',
'game^0.008236116942484021', 'tie^0.0074787233797496445']
6 Documents
-nrounds = 10:
['hockey^0.560394253938797', 'people^0.2763037510600535',
'game^0.014419335370623761', 'tie^0.013430480710660976']
6 Documents
-nrounds = 20:
['hockey^0.5467982729721478', 'people^0.26632017864434',
'game^0.02643408779701819', 'tie^0.02499121598577075']
6 Documents
nrounds = 100:
['hockey^0.45869683838656583', 'people^0.20162289212818338',
'game^0.10430475545354491', 'tie^0.09991926942728763']
6 Documents
nrounds = 200:
['hockey^0.38704053736145333', 'game^0.16763997841587186',
'tie^0.16086115054462152', 'people^0.1490020890736349']
6 Documents

En 500 nrounds no se obtiene un número de documentos por que la ejecución no finaliza.

nrounds = 500:
['hockey^0.3592303934921091', 'game^0.19222067316820185',
'tie^0.18451297818986373', 'people^0.1285797105454069']
(No result in nº docs)

Con las pruebas realizadas, debemos refutar la hipótesis inicial, puesto que no parece que los valores converjan. Quizás si se pudiese explorar un mayor número de nrounds si se podría observar ese comportamiento, pero puesto que a partir de 500 la ejecución no finaliza, no se puede determinar.