Implementing User Relevance Feedback

21-10-2022

He Chen

Daniel Muñoz

Grupo 12

2 We will, we will Rocchio you
What is the difference in computational cost of merging ordered vectors or using dictionaries for this operation? Discuss it in your report.

La unión de dos vectores es O(n * log(n)) para ordenar los vectores y O(n) para hacer el merge. Así que el coste es de O(n * log(n)) donde m es el número de vectores a fusionar. En cambio, para la unión de dos map(diccionario) se hace en coste O(n) donde n es el tamaño del vector más grande. Para implementar la función de Rocchio, vamos haciendo merge en cada iteración, por lo tanto, usando diccionario conseguimos una mejor eficiencia.

3 Experimenting

Para la query inicial, hemos escogido la siguiente query: ***computer hardware^2***, y los parámetros iniciales: alpha = 1, beta = 1, R = 5, k = 5, nrounds = 5. En los experimentos iremos modificando una de estas variables para ver su efecto en el resultado, pero manteniendo las otras variables a los valores iniciales.
La fuente de documentos usados para el experimento es 20_newsgroups.

Influencia Alpha
Como hemos analizado anteriormente, aumentando la alpha, el peso resultante de los términos de la query original (computer, hardware) incrementa. Cuando alpha = 0, desaparecen los términos de la query original (tienen peso 0), reaparece para alpha = 0.5 y a media que lo vamos aumentando, los pesos de los términos computer y hardware aumenta(hardware aumenta más que computer porque le hemos dado más peso).

alpha	query
0	['paschal^0.2303118230616868', 'tampa^0.16325347217990988', 'reanable^0.12951703880265988', 'proccessing^0.12951703880265988', 'tscs^0.12045617631204437']
0.5	['hardware^0.07415454508656098', 'c^0.03180001981811068', 'paschal^0.014394488941355425', 'computer^0.011008642815216393', 'tampa^0.010203342011244367']
1	['hardware^2.186472721384976', 'c^1.008800317089771', 'paschal^0.2303118230616868', 'computer^0.1761382850434623', 'tampa^0.16325347217990988']
2	['hardware^66.98356354215962', 'c^32.14080507343633', 'paschal^3.6849891689869887', 'computer^2.8182125606953967', 'tampa^2.612055554878558']

Influencia Beta

Justo al contrario que alpha, beta pondera la query nueva, aumentado su valor, los términos de la query nueva cogen más peso como se puede ver con beta = 2, estas nuevas palabras serían “paschal” y “tampa” que son las que más peso tienen y “hardwar”. Lo que no logramos entender es que cuando beta = 0, no damos importancia a la query nueva, por lo que los únicos términos con pesos serían hardware y computer, sin embargo nos sale el término c. Pensamos que la razón puede estar en el preproceso (tokenization, stemming…) de los datos.

Beta	Query final
0	[('hardware', 2.0), ('c', 1.0), ('moreov', 0.0), ('pc', 0.0), ('which', 0.0)]
0.1	[('hardware', 2.0), ('hardwar', 1.4590668467924088), ('paschal', 1.2709155629512345), ('c', 1.1259424523544814), ('comput', 0.4569822005413236)]
0.5	[('paschal', 10.167324503609876), ('tampa', 5.405229546822753), ('comput', 5.1772034111960075), ('hardwar', 5.1123674265023595), ('hardware', 2.0)]
1	[('paschal', 22.87648013312222), ('tampa', 16.73047240683233), ('tscs', 10.635294504186803), ('hardwar', 9.21071156179764), ('comput', 9.164435088613521)]
2	[('paschal', 10.167324503609876), ('tampa', 5.405229546822753), ('comput', 5.1772034111960075), ('hardwar', 5.1123674265023595), ('hardware', 2.0)]

influencia R

Aumentando R incluimos más términos en la query (el valor de R nos indica el número de términos en la query como se puede observar en la tabla), por lo tanto aumentamos el recall. Esto significa que la nueva query tiene que apuntar a más términos, disminuyendo la precisión. Esto lo podemos ver en el experimento, ya que con R = 3, las tres palabras de la query tienen relación, ya que la nueva palabra es paschal y al ser un lenguaje de programación, consideramos que tiene relación con computer y hardware (c es la stemmización de computer y hardware era la palabra de la query). En cambio con R = 10, se puede ver que hay mas de 1 palabra que no esta relacionada, como “renable”, que intuimos que se refiere a la palabra reenable, antes de ser pasada por los filtros y no le vemos relación con hardware o computer, otros ejemplo serían la palabra wilde, tampa, zeurich y tilk que por sí solas no tienen relación.

R	query
3	['hardware^2.186472721384976', 'c^1.008800317089771', 'paschal^0.2303118230616868']
4	['hardware^2.186472721384976', 'c^1.008800317089771', 'paschal^0.2303118230616868', 'computer^0.1761382850434623']
5	['hardware^2.186472721384976', 'c^1.008800317089771', 'paschal^0.2303118230616868', 'computer^0.1761382850434623', 'tampa^0.16325347217990988']

6	['hardware^3.047890869051953', 'paschal^1.151559115308434', 'c^1.142672624800129', 'tampa^0.8162673608995494', 'reanable^0.6475851940132994', 'computer^0.4187317480938211']
7	['hardware^2.186472721384976', 'c^1.008800317089771', 'paschal^0.2303118230616868', 'computer^0.1761382850434623', 'tampa^0.16325347217990988', 'proccessing^0.12951703880265988', 'reanable^0.12951703880265988']
10	['hardware^2.9962400972277163', 'paschal^1.151559115308434', 'c^1.1044732189355442', 'tampa^0.8162673608995494', 'reanable^0.6475851940132994', 'proccessing^0.6475851940132994', 'wilde^0.6149135930876399', 'zuerich^0.6149135930876399', 'tik^0.6149135930876399', 'computer^0.5535110394262035']

influencia K

Se puede apreciar que mientras aumentamos el valor de k, cada vez más palabras de la query tienen relación con la query inicial. Por ejemplo, con K = 1 podemos observar que hay algunas palabras que no tienen mucho sentido como ‘gsfc’ ‘eskimo’ y ‘nanao’. En cambio, con k = 50 la precisión es buena, ya que todas las palabras están relacionadas con las de la query. Por lo tanto, aumentando la K, aumentamos la precisión.

K	Query
1	[('sys', 19.441010817049396), ('nanao', 18.973526416351795), ('comp', 18.283337356191414), ('eskimo', 17.243385576524012), ('gsfc', 15.73570675106281)]
2	[('shopper', 15.578151238045798), ('hardwar', 13.942820254097343), ('ubvm', 10.924411598194832), ('comput', 10.412252670561802), ('hardware', 2.0)]
5	[('paschal', 22.87648013312222), ('tampa', 16.73047240683233), ('tscs', 10.635294504186803), ('hardwar', 9.21071156179764), ('comput', 9.164435088613521)]
10	[('hardwar', 10.198579490334694), ('paschal', 8.896408940658642), ('comput', 8.614348618160825), ('tampa', 3.603486364548502), ('hardware', 2.0)]
15	[('hardwar', 10.669071177338367), ('comput', 7.743664263652544), ('paschal', 4.236385209837448), ('hardware', 2.0), ('c', 1.4697154400167138)]
50	[('hardwar', 10.078418638949676), ('softwar', 4.759341557225371), ('comput', 3.8351256117210655), ('hardware', 2.0), ('c', 1.601643219719735)]

influencia nrounds

Este valor lo tenemos que adaptar experimentalmente. Con pocas iteraciones nos salen términos muy imprecisos. En cambio tampoco tiene sentido hacer muchas iteraciones, ya que como podemos ver para nrounds 10, 50 y 100, las query no han cambiado, lo

único que hacemos es aumentar sus pesos sin modificar el orden en el que están posicionados.

nrounds	**Query**
1	[('paschal', 2.541831125902469), ('hardwar', 2.2139508524687903), ('comput', 2.0477430252104876), ('hardware', 2.0), ('tampa', 1.801743182274251)]
5	[('paschal', 22.87648013312222), ('tampa', 16.73047240683233), ('tscs', 10.635294504186803), ('hardwar', 9.21071156179764), ('comput', 9.164435088613521)]
10	[('paschal', 48.29479139214691), ('tampa', 36.38256847013443), ('tscs', 23.929412634420306), ('comput', 17.961440308551577), ('hardwar', 15.649759460962596)]
50	[('paschal', 251.6412814643441), ('tampa', 204.7258913653325), ('tscs', 130.2823576762882), ('comput', 67.00210655204341), ('hardwar', 46.83097566558034)]
100	[('paschal', 505.8243940545903), ('tampa', 415.1550449843301), ('tscs', 263.2235389786237), ('comput', 127.21832970322464), ('hardwar', 84.8568490858459)]