Homework - 4

INFO-533 Web Based Information Retrieval and Search

Declaration

“I have done this assignment completely on my own. I have not copied it, nor have I given my solution to anyone else. I understand that if I am involved in plagiarism or cheating I will have to sign an official form that I have cheated and that this form will be stored in my official university record. I also understand that I will receive a grade of 0 for the involved assignment and my grade will be reduced by one level (e.g., from A to A- or from B+ to B) for my first offense, and that I will receive a grade of “F” for the course for any additional offense of any kind.”

- Dorbala Sankarshana Sharma

1.

Perfect or almost perfect recall can be achieved using a search system, although in most situations, this may not be desired or practicable.

The proportion of relevant items the search engine successfully discovered out of all the relevant items in the dataset is known as recall. High recall is frequently exchanged for low accuracy, or the proportion of retrieved items that are actually relevant. A search algorithm that returns every item in the dataset as relevant is essentially ineffective due to its flawless recall and low accuracy. Furthermore, in order to achieve high recall, the system must retrieve a lot of irrelevant things with relevant ones, which can be tedious and annoying for users who must inspect through a lot of irrelevant results in order to locate what they are searching for.

Other measures which include the relevance and rank of the retrieved items, may also suffer as a result of this. Therefore, while it is technically feasible to create a search system that achieves perfect or almost perfect recall, it is typically not practicable or desirable to do so, and depending on the particular use case and requirements of the system, a balance between recall and precision needs to be found.

D1 higher term frequenc longer document
D2 repeat term give higher term frequenc
D3 term frequenc import term document
D4 invers document frequenc import term corpu

Term Frequency

Terms	T1	T2	T3	T4	T5	T6
Documents						
D1	higher	term	frequenc	longer	document	
D2	repeat	term	give	higher	frequenc	
D3	term	frequenc	import	document		
D4	invers	document	frequenc	import	term	corpu

Terms	T1	T2	T3	T4	T5	T6
Documents						
D1	1	1	1	1	1	0
D2	1	2	1	1	1	0
D3	2	1	1	1	0	0
D4	1	1	1	1	1	1

Normalized TF

Terms	T1	T2	T3	T4	T5	T6
Documents						
D1	1/5	1/5	1/5	1/5	1/5	0
D2	1/6	2/6	1/6	1/6	1/6	0
D3	2/5	1/5	1/5	1/5	0	0
D4	1/6	1/6	1/6	1/6	1/6	1/6

Terms	T1	T2	T3	T4	T5	T6
Documents						
D1	0.2	0.2	0.2	0.2	0.2	0
D2	0.16667	0.33333	0.33333	0.33333	0.33333	0
D3	0.4	0.2	0.2	0.2	0	0
D4	0.16667	0.16667	0.16667	0.16667	0.16667	0.16667

$$IDF(\,term) \;=\; 1 \;+\; \log\left(\frac{Total\ No\,.\ Of\ \ Documents}{No\,.\ of\ \ documents\ with\ term\ in\ it}\right)$$

Terms	TF D1	TF D2	TF D3	TF D4	IDF
corpu	0	0	0	1	1.602
document	1	0	1	1	1.125
frequenc	1	1	1	1	1
give	0	1	0	0	1.602
higher	1	1	0	0	1.301
import	0	0	1	1	1.301
invers	0	0	0	1	1.602
longer	1	0	0	0	1.602
repeat	0	1	0	0	1.602
term	1	2	2	1	1

Terms	TF-IDF D1	TF-IDF D2	TF-IDF D3	TF-IDF D4
corpu	0	0	0	1.602
document	1.125	0	1.125	1.125
frequenc	1	1	1	1
give	0	1.602	0	0
higher	1.301	1.301	0	0
import	0	0	1.301	1.301
invers	0	0	0	1.602
longer	1.602	0	0	0
repeat	0	1.602	0	0
term	1	2	2	1

2. Query = “invers”

Terms	corpu	document	frequenc	give	higher	import	invers	longer	repeat	term
Query	0	0	0	0	0	0	1	0	0	0
TF D1	0	1	1	0	1	0	0	1	0	1
TF D2	0	0	1	1	1	0	0	0	1	2
TF D3	0	1	1	0	0	1	0	0	0	2
TF D4	1	1	1	0	0	1	1	0	0	1

$$CosSim(Q,D) = \frac{Q.D}{|Q||D|}$$

$$CosSim(Q,D1) = \frac{1 \cdot 0}{\left|\sqrt{1^2}\right|\left|\sqrt{1.125^2+1^2+1.301^2+1.602^2+1^2}\right|} = 0$$

Similarly the dot product between the query Q and Documents D2 and D3 is 0 (because there are no common terms between the query and the document) hence

$$CosSim(Q, D2) = 0$$

$$CosSim(Q, D3) = 0$$

$$CosSim(Q, D4) = \frac{1 \cdot 0}{\left|\sqrt{1^2}\right|\left|\sqrt{1.125^2 + 1^2 + 1.301^2 + 1.602^2 + 1^2 + 1.602^2}\right|} = 0.314$$

3. Query = “repeat higher cours”

Terms	corpu	document	frequenc	give	higher	import	invers	longer	repeat	term	cours
Query	0	0	0	0	1	0	0	0	1	0	1
TF D1	0	1	1	0	1	0	0	1	0	1	0
TF D2	0	0	1	1	1	0	0	0	1	2	0
TF D3	0	1	1	0	0	1	0	0	0	2	0
TF D4	1	1	1	0	0	1	1	0	0	1	0

$$CosSim(Q, D) = \frac{Q.D}{|Q||D|}$$

$$CosSim(Q, D1) = \frac{1 \cdot 1}{\left|\sqrt{1^2 + 1^2 + 1^2}\right|\left|\sqrt{1.125^2 + 1^2 + 1.301^2 + 1.602^2 + 1^2}\right|} = 0.364$$

$$CosSim(Q, D2) = \frac{1 \cdot 1 + 1.1}{\left|\sqrt{1^2 + 1^2 + 1^2}\right|\left|\sqrt{1.602^2 + 1^2 + 1.301^2 + 1.602^2 + 2^2}\right|} = 0.581$$

The dot product between the query Q and Documents D3 and D4 is 0 (because there are no common terms between the query and the document) hence

$$CosSim(Q, D3) = 0$$

$$CosSim(Q, D4) = 0$$

4.

Alphabetizing the index can help with faster and more efficient lookups of terms in the index. By sorting the index alphabetically, we can easily locate terms and their corresponding postings. This can significantly speed up the process of finding documents containing a given term, especially when dealing with large collections of documents.

In addition, an alphabetized index can help with tasks such as range queries, where we need to find all terms that fall within a certain range (e.g., terms starting with a certain letter or set of

letters). With an alphabetized index, we can easily locate all terms that match a particular prefix, making it easier to perform range queries efficiently.

Overall, an alphabetized index can improve the efficiency and speed of searching and querying a collection of documents, which is important for many applications such as search engines, recommender systems, and information retrieval systems.

5. Query = “inverse document frequency is importance of a term in a corpus”

In the given corpus, the term "importance" in document D4 has been stemmed to "import". Suppose we perform a query for **"role of an importer"**. Because of stemming, the query term "importer" would be stemmed to "import", which is also present in document D4. Thus, the query for "role of an importer"." would likely return document D4. This may not be desirable if we are interested in finding documents that specifically mention the term "importance" and not just any term that contains the substring "import". In this case, stemming can lead to a false positive result, where a document is returned even though it may not be relevant to the query. This is an example of a situation where stemming can be bad, as it can potentially reduce the precision of the search results.

6.

a. Query = “role import”

Terms	**TF D1**	**TF D4**
corpu	0	1
document	1	1
frequenc	1	1
give	0	0
higher	1	0
import	0	1
invers	0	1
longer	1	0
repeat	0	0
term	1	1
cours	0	0

$$CosSim(Q,D) = \frac{Q.D}{|Q||D|}$$

As the dot products between Query and Documents 1 is 0, the cosine similarity is also 0.

$$CosSim(Q,D1) = \frac{0}{|Q|\left|D1\right|} = 0$$

$$CosSim(Q,D4) = \frac{1.1}{\left|\sqrt{1^2+1^2}\right|\left|\sqrt{1^2+1^2+1^2+1^2+1^2++1^2}\right|} = 0.288$$

b.

Terms	TF D1	TF D4	IDF	Terms	TF-IDF D1	TF-IDF D4
corpu	0	1	1.602	corpu	0	1.602
document	1	1	1.125	document	1.125	1.125
frequenc	1	1	1	frequenc	1	1
give	0	0	1.602	give	0	0
higher	1	0	1.301	higher	1.301	0
import	0	1	1.301	import	0	1.301
invers	0	1	1.602	invers	0	1.602
longer	1	0	1.602	longer	1.602	0
repeat	0	0	1.602	repeat	0	0
term	1	1	1	term	1	1

As the dot products between Query and Documents 1 is 0, the cosine similarity is also 0.

$$CosSim(Q,D4) = \frac{1.1}{\left|\sqrt{1^2+1^2}\right|\left|\sqrt{1.125^2+1^2+1^2+1.301^2+1.602^2+1.602^2}\right|} = 0.222$$

c.

Terms	TF D1	TF D2	TF D3	TF D4	IDF	Terms	TF-IDF D1	TF-IDF D2	TF-IDF D3	TF-IDF D4
corpu	0	0	0	1	1.602	corpu	0	0	0	1.602
docume	1	0	1	1	1.125	docume	1.125	0	1.125	1.125

nt						nt				
frequenc	1	1	1	1	1	frequenc	1	1	1	1
give	0	1	0	0	1.602	give	0	1.602	0	0
higher	1	1	0	0	1.301	higher	1.301	1.301	0	0
import	0	0	1	1	1.301	import	0	0	1.301	1.301
invers	0	0	0	1	1.602	invers	0	0	0	1.602
longer	1	0	0	0	1.602	longer	1.602	0	0	0
repeat	0	1	0	0	1.602	repeat	0	1.602	0	0
term	1	2	2	1	1	term	1	2	2	1

$$|Q| = \sqrt{2}$$

$$|D1| = \sqrt{1.125^2 + 1^2 + 1^2 + 1.602^2 + 1.301^2} = 2.743$$

$$|D2| = \sqrt{1.602^2 + 1^2 + 2^2 + 1.602^2 + 1.301^2} = 3.438$$

$$|D3| = \sqrt{1.125^2 + 1^2 + 2^2 + 1.301^2} = 2.821$$

$$|D4| = \sqrt{1.125^2 + 1^2 + 1^2 + 1.301^2 + 1.602^2 + 1.602^2} = 3.176$$

$$CosSim(Q, D1) = \frac{0}{|Q||D1|} = 0$$

$$CosSim(Q, D2) = \frac{1.0}{\left|\sqrt{1^2 + 1^2}\right|\left|\sqrt{1^2 + 2^2 + 1.301^2 + 1.602^2 + 1.602^2}\right|} = 0$$

$$CosSim(Q, D3) = \frac{1.1}{\left|\sqrt{1^2 + 1^2}\right|\left|\sqrt{1^2 + 1.125^2 + 1.301^2 + 2^2}\right|} = 0.250$$

$$CosSim(Q, D4) = \frac{1.1}{\left|\sqrt{1^2 + 1^2}\right|\left|\sqrt{1.125^2 + 1^2 + 1^2 + 1.301^2 + 1.602^2 + 1.602^2}\right|} = 0.222$$

D3>D4>D2=D1

We can also achieve this by just comparing the most common terms between query and documents. Only documents D3 and D4 have common terms with query. Based on this we can say that D3 and D4 will have higher rank.

d.

A problem with using natural (raw) frequency is that it may result in giving too much weight to terms that are commonly used across all documents, while underweighting terms that are specific to a particular document. This can lead to misleading results in document ranking.
For example, consider a corpus of documents that includes a large number of news articles about politics, and a small number of articles about technology. If we use natural frequency to calculate the TFIDF scores for each term, the term "government" may appear in many of the documents, and therefore have a high raw frequency. However, this does not necessarily mean that it is a highly relevant term for document ranking. Conversely, a term like "machine learning" may only appear in a few documents, but may be highly relevant for ranking the documents that do contain it. By using raw frequency alone, we may overlook this important term and rank the documents incorrectly. To avoid this problem, it is often better to use more sophisticated methods for calculating term frequency, such as logarithmic term frequency or sublinear scaling. These methods can help to downweight frequently occurring terms and give more weight to rare terms that may be more important for ranking.