CS6730 : Natural Language Processing
Assignment #2

E Naveen **(ME16B077)** Pawan Prasad **(ME16B179)**

1. Consider the following table of term frequencies for threedocuments. Give an inverted index representation for the same.

	Doc A	Doc B	Doc C
cat	4	2	1
dog	3	0	3
animal	1	3	3

Figure 1: Term Document Matrix

Ans:

cat → Doc A, Doc B, Doc C
dog → Doc A, Doc C
animal → Doc A, Doc B, Doc C

2. Next, we must proceed on to finding a representation for the text documents. In the class,we saw about the TF-IDF measure. What would be the TF-IDF vector representations for the documents in the above table? State the formula used.

Ans: For each document,

$$tf_i = term\ frequency\ of\ i^{th}\ term$$
$$IDF_i = Inverse\ Document\ frequency\ of\ i^{th}\ term$$

For the document vector, each term weight w_i is given by:

$$\begin{aligned} w_i &= t_{fi} * IDF_i \\ &= t_{fi} * log\left(\frac{D}{df_i}\right) \end{aligned} \tag{1}$$

D = total number of documents
df_i = number of documents in which the i^{th} term occurs atleast once.
The TF-IDF weights are calculated and presented in the table below:

		Term Frequency Tf_i						Weights: $W_i = Tf_i * IDF_i$			
TERMS	Q	Doc A	Doc B	Doc C	df_i	D/df_i	IDF_i	Q	Doc A	Doc B	Doc C
cat	0	4	2	1	3	1	0	0	0	0	0
dog	1	3	0	3	2	1.5	0.1761	0.1761	0.5283	0	0.5283
animal	0	1	3	3	3	1	0	0	0	0	0

Figure 2: calculating term weights

So,

$$\begin{aligned} DocA &= 0 \times cat\ +\ 0.5283 \times dog\ +\ 0 \times animal \\ DocB &= 0 \times cat\ +\ 0 \times dog\ +\ 0 \times animal \\ DocC &= 0 \times cat\ +\ 0.5283 \times dog\ +\ 0 \times animal \end{aligned} \tag{2}$$

3. Suppose the query is "dog", which documents would be retrieved based on the inverted index constructed before?

Ans: The given query is a single word "dog". From inverted index, we observe that this word occurs in documents A and C. Hence, retrieved documents would be A, C.

4. Find the cosine similarity between the query and each of the retrieved documents. Rank them in descending order.

Ans: The TF-IDF representation of the query is calculated as shown in Figure 2. Cosine similarity between query Q and Document D:

$$Cos(\theta) = \frac{Q.D}{|Q||D|}$$
$$Cos(\theta_A) = \frac{Q.A}{|Q||A|} = \frac{0 + (0.1761 * 0.5283) + 0}{0.1761 * 0.5283} = 1$$
$$Cos(\theta_B) = \frac{Q.B}{|Q||B|} = \frac{0 + 0 + 0}{|Q||B|} = 0$$
$$Cos(\theta_C) = \frac{Q.C}{|Q||C|} = \frac{0 + (0.1761 * 0.5283) + 0}{0.1761 * 0.5283} = 1$$

Rank 1 → Doc A and Doc C
Rank 2 → Doc B

We can rank Doc A and Doc C with different ranks if the document order is considered (i.e., Doc A appears before Doc C). Then,
Rank 1 → Doc A
Rank 2 → Doc C
Rank 3 → Doc B

5. Now, you are set to build a real-world retrieval system. Implement an Information Retrieval System for the Cranfield Dataset using the Vector Space Model.

Ans: Refer code for implementation

6. (a) What is the IDF of a term that occurs in every document?(b) Is the IDF of a term always finite? If not, how can the formula for IDF be modified to make it finite?

Ans: Let number of documents = N:
Let number of documents in which term occurs = n
(a)

$$IDF = log\left(\frac{N}{n = N}\right) = log(1) = 0$$

(b) If n = 0, that is, if term does not occur in any document, then $IDF \to \infty$. This situation arises when a term in the query does not occur in any of the documents. Hence, we can add a smoothing constant and modify the IDF formula as follows:

$$IDF = log\left(\frac{N}{n + k}\right)$$

where k is some positive constant. We can take k=1.

7. Can you think of any other similarity/distance measure that can be used to compare vectors other than cosine similarity. Justify why it is a better or worse choice than cosine similarity for IR.

Ans: Euclidean distance is another measure that can be used to compare vectors as well. But this would be a relatively worse choice when compared to cosine similarity. This is because it is skewed as

it incorporates the magnitudes of the vectors and in practical situations the query length is much lesser than the document lengths creating an imbalance.

8. Why is accuracy not used as a metric to evaluate information retrieval systems?

Ans:

$$Accuracy = \left(\frac{TP}{N}\right)$$

where, TP = True positives, N = total number of documents.
This is a skewed measure due to the inherent class imbalance in IR, i.e, the number of true negatives is almost the same as the total number of documents ($TN \approx N$). This is because for a given query, very few documents would be relevant to it, while the rest would be irrelevant. Consider 10^4 documents in total and 3 documents relevant to a query; Suppose the IR system retrieves none, the accuracy would still be $(10^4 - 3)/10^4$ which is a high value even though the performance of the IR system was poor.

9. For what values of alpha does the F_α -measure give more weightage to recall than to precision?

Ans: For $\alpha < 0.5$, more weight is given to recall than precision

10. What is a shortcoming of Precision @ K metric that is addressed by Average Precision @k?

Ans: Precision @ k tells us the ratio of relevant documents to retrieved documents @ k. Total number of relevant documents for a query has a strong influence on Precision @ k. Average Precision @ k is a measure of recall as well and adjusts for the size of the set of relevant documents. It is the area under the Precision-Recall curve @ k.

11. What is Mean Average Precision (MAP) @ k? How is it different from Average Precision(AP) @ k ?

Ans: mAP is average precision (AP) averaged across a set of queries Q. If the set of relevant documents for a query q_j is $\{d_1, d_2, ..., d_{m_j}\}$ and R_{jk} is the set of ranked retrieval results from the top result until we get document d_k, then AP is given by:

$$AP = \frac{1}{m_j} \sum_{k=1}^{m_j} Precision\,(R_{jk})$$

Now, mAP is calculated by averaging AP over a set of queries $q_j \in Q$:

$$mAP = \frac{1}{|Q|} \sum_{j=1}^{|Q|} \frac{1}{m_j} \sum_{k=1}^{m_j} Precision\,(R_{jk})$$

Therefore, for a single query $|Q| = 1$ and $mAP = AP$. Moreover, AP approximates the area under the un-interpolated precision-recall curve whereas mAP is the average area under the precision-recall curve for a set of queries Q [1].

12. For Cranfield dataset, which of the following two evaluation measures is more appropriate and why? (a) AP (b) nDCG

Ans: nDCG is more appropriate for the Information Retrieval system using Cranfield Dataset.
This is because average precision does not take into account the rankings of the retrieved documents whereas in nDCG evaluation, both ranking and relevance of the documents are considered. Moreover, we can use scaled relevance {1,2,3,4} instead of binary values {0,1}.

$$DCG_k = \sum_{i=1}^{k} \frac{rel_i}{log_2\,(i+1)}$$

Therefore, relevant documents appearing lower in a search result list are penalized - the relevance value is reduced logarithmically proportional to the position of the result [2]. Thus using this measure, we assert that:

- Highly relevant documents are more useful when appearing earlier in a search engine result list (have higher ranks).
- Highly relevant documents are more useful than marginally relevant documents, which are in turn more useful than non-relevant documents.

13. Implement the following evaluation metrics for the IR system:(a) Precision @ k(b) Recall @ k(c) F-Score @ k(d) Average Precision @ k(e) nDCG @ k

Ans: Refer code for implementation

14. Assume that for a given query, the set of relevant documents is as listed in *cran_qrels.json*. **Any document with a relevance score of** $1-4$ **is considered as relevant. For each query in the Cranfield dataset, find the Precision, Recall, F-score, Average Precision and nDCG scores for** $k = 1-10$**. Average each measure over all queries and plot it as function of k. Code for plotting is part of the given template. You are expected to use the same.Report the graph with your observations based on it.**

Ans: We observe the following from the plots:

- Recall monotonically increases with k
- Fscore ($\alpha = 0.5$) takes into account both precision and recall resulting in the observed trend. For example, consider $k = 1$: Since recall is very low, Fscore is low as well

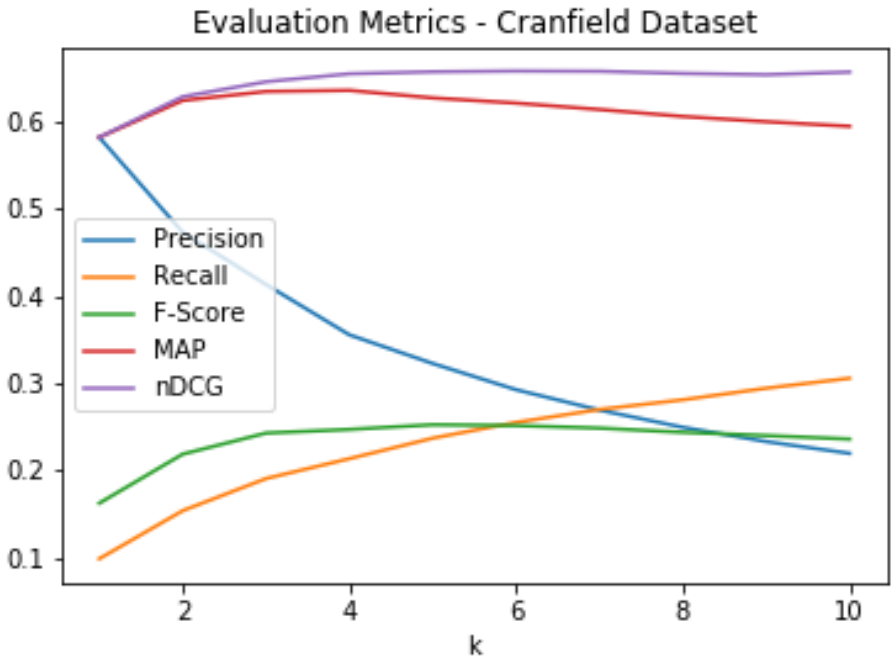

Figure 3: Graph of Evaluation metrics vs k (1-10)

The respective metric values for k (1-10) corresponding to the plot are given in the table below:

k	precision	recall	Fscore	mAP	nDCG
1	0.582	0.099	0.162	0.582	0.582
2	0.473	0.154	0.219	0.624	0.629
3	0.413	0.191	0.243	0.635	0.646
4	0.356	0.213	0.247	0.636	0.655
5	0.323	0.237	0.252	0.627	0.657
6	0.293	0.255	0.251	0.621	0.658
7	0.269	0.270	0.249	0.614	0.658
8	0.249	0.281	0.244	0.606	0.655
9	0.233	0.294	0.240	0.600	0.654
10	0.220	0.306	0.236	0.595	0.657

Figure 4: Evaluation metrics VS k used for plots (rounded to 3 decimal places here)

15. Analyse the results of your search engine. Are there some queries for which the search engine's performance is not as expected? Report your observations.

Ans: For analysing the search engine we tabulated the results using the evaluation metrics - Precision, Recall, F-score, nDCG and Average Precision for each of the queries in cran_queries.json file at k = 10 value. For the report of the tabulated values please refer to [3]. We filtered the queries based on their F-score (greater than 0.6) & nDCG measure (greater than 0.85) and the corresponding queries for which documents retrieved match better are summarized in the below table:

Query ID	Queries
3	what problems of heat conduction .....
20	has anyone formally determined the influence of joule ....
92	given complete freedom in the design of an airplane ....
101	why does the incremental theory and the deformation theory ....
120	are previous analyses of circumferential thermal buckling ....
130	what are the flutter characteristics of the exposed skin ....
185	experimental studies on panel flutter ....

Figure 5: Queries having precision > 0.60 and nDCG > 0.85

Yes, there are some queries for which the search engine's performance is not as expected. There are **41** *such queries* in fact that report **0** *F-score, nDCG and Average Precision values.* This is an implication of no Relevant documents in the top 10 Retrieved documents. The query IDs of such queries are enlisted below. Based on an overlook at these queries it is observed that they correspond to questions that require a much higher level of granularity for the information retrieved and are queries that are a bit too specific.

Query_IDs	Query_IDs	Query_IDs	Query_IDs	Query_IDs	Query_IDs
9	62	85	116	152	205
12	63	87	119	153	207
19	66	88	133	154	216
22	74	95	141	167	217
28	76	98	143	180	218
44	78	110	150	181	219
61	79	115	151	204	

Figure 6: Queries for which IR system has the worst performance

16. Do you find any shortcoming(s) in using a Vector Space Model for IR? If yes, report them.

Ans: We observe the following shortcomings:

- **Computationally intensive:** Calculating TF and IDF is the bottleneck of operations. Even after optimizing our code, it takes around 19 seconds to execute.
- Lack of **model flexibility:** Each time we add a new term into the term space we need to recalculate all vectors.
- Words in documents are assumed to be independent (especially since we are using only uni-grams). Since this can result in irrelevant documents being retrieved, this reduces precision.
- Documents with similar content but different vocabularies result in small cosine similarity. Since such documents are also relevant but are not retrieved, this reduces recall.
- **Keyword spam:** We only need to repeat a keyword many times in a document to increase its weight (via *tf* term). Thus the system can be easily deceived.

17. While working with the Cranfield dataset, we ignored the titles of the documents. But,titles can sometimes be extremely informative in information retrieval, sometimes even more than the body. State a way to include the title while representing the document as a vector. What if we want to weigh the contribution of the title three times that of the document?

Ans: The simplest way would be to append the title to the beginning of the document and calculate the vector representation of the documents as usual. But this would not give any special preference to the title in vector representation except for a slight increase in the term frequency of those terms in the title. So a better way would be to create a separate TF-IDF representation for the titles and TF-IDF representation for the body of the document separately and simply add the two representations together.

If we want to weigh the contribution of the title three times that of the document, we could then multiply the vector representation of the title by a factor of 3 before adding it with vector representation of the document.

18. Suppose we use bigrams instead of unigrams to index the documents, what would be its advantage(s) and/or disadvantage(s)?

Ans: One advantage is we can reduce false positives since using bi-grams captures co-occurrences and collocations. For example, consider the query containing the collocation "heavy rain". In case of bi-grams, our IR system will most likely **not** retrieve documents in the context of "heavy duty" since it has captured the necessity that heavy and rain occur together.

A big disadvantage is increased computational complexity when calculating bigram term frequencies, since we have to update our term-frequency matrix conditioned upon pair-wise occurrence of words in both query and documents.

19. In the Cranfield dataset, we have relevance judgements given by the domain experts.In the absence of such relevance judgements, can you think of a way in which we can get relevance feedback from the user himself/herself? Ideally, we would like to keep the feedback process to be non-intrusive to the user. Hence, think of an 'implicit' way of recording feedback from the users.

Ans: Relevance feedback could be gained based on user's behaviour once documents are retrieved for a query such as when the searcher chooses to view a specific document by clicking on it. In this the document that was clicked would be assigned a higher relevance value relative to the remaining documents [4]. On the other hand, when the searcher does not select any document meaning there is no relevant documents from those retrieved, this could be treated as negative feedback and the relevance values for all these retrieved documents are reduced. The time a user spends on viewing each of the document

although very difficult to implement could also be used, possibly as a supplement to the above proposed method.

REFERENCES:

[**1**] Evaluation of results:
https://nlp.stanford.edu/IR-book/html/htmledition/evaluation-of-ranked-retrieval-results-1.html
[**2**] nDCG metric: https://en.wikipedia.org/wiki/Discounted_cumulative_gain
[**3**] Tabulated results based on evaluation metrics for all queries in Google Sheets:
https://docs.google.com/spreadsheets/d/1jdAMYJBbfR4uuamEGDVGYdSBDbpnTU2_PJINpmHXNpg/edit?usp=sharing
[**4**] Relevance Feedback article: https://en.wikipedia.org/wiki/Relevance_feedback