CS6370 – Natural Language Processing

Assignment 2

Teammate 1: R Narendhiran CH18B015
Teammate 2: Akiti Gunavardhan Reddy CH18B035

1.Now that the Cranfield documents are pre-processed, our search engine needs a data structure to facilitate the 'matching' process of a query to its relevant documents. Let's work out a simple example. Consider the following three sentences:
S1 Herbivores are typically plant eaters and not meat eaters
S2 Carnivores are typically meat eaters and not plant eaters
S3 Deers eat grass and leaves
Assuming are, and, not as stop words, arrive at an inverted index representation for the above documents (treat each sentence as a separate document).
Ans. Inverted Index Representation of given documents are:

Herbivore → $S1$
Typically → $S1, S2$
Plant → $S1, S2$
Eater → $S1, S2$
Meat → $S1, S2$
Carnivore → $S2$
Deer → $S3$
Eat → $S3$
Grass → $S3$
Leaf → $S3$

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. Formula of TF-IDF measure
For each document vector, weight corresponding to each word/term(i) w_i is given by:

$$w_i = tf_i * IDF_i$$

$$w_i = tf_i * log(\frac{D}{df_i})$$

where:
tf_i = term frequency of i^{th} term
IDF_i = Inverse Document Frequency of i^{th} term
df_i = number of documents which has i^{th} term
D = Total number of documents in the dataset

Following table shows TF-IDF measures calculated for each document:

	Term Frequency tf$_i$							TF-IDF Vectors			
Terms	Q	S1	S2	S3	df$_i$	D/df$_i$	IDF$_i$	Q	S1	S2	S3
Herbivore	0	1	0	0	1	3	0.477	0	0.477	0	0
Typically	0	1	1	0	2	1.5	0.176	0	0.176	0.176	0
Plant	1	1	1	0	2	1.5	0.176	0.176	0.176	0.176	0
Eater	1	2	2	0	2	1.5	0.176	0.176	0.352	0.352	0
Meat	0	1	1	0	2	1.5	0.176	0	0.176	0.176	0
Carnivore	0	0	1	0	1	3	0.477	0	0	0.477	0
Deer	0	0	0	1	1	3	0.477	0	0	0	0.477
Eat	0	0	0	1	1	3	0.477	0	0	0	0.477
Grass	0	0	0	1	1	3	0.477	0	0	0	0.477
Leaf	0	0	0	1	1	3	0.477	0	0	0	0.477

Hence last three columns shows the TF-IDF vector of each document expressed vertically.

3.Suppose the query is ”plant eaters”, which documents would be retrieved based on the inverted index constructed before?
Ans. Given query ”plant eaters”, when reduced gives (after lemmatization, stopword removal and all) ”plant” and ”eater”, where ”plant” is in ”S1” and ”S2” and ”eater” is in ”S1” and ”S2”.
Hence based on inverted index construction we retrieve both ”S1” and ”S2” documents.

4.Find the cosine similarity between the query and each of the retrieved documents. Rank them in descending order.
Ans. From Part 2 we obtained the query document vectors as shown in the below table:

	Herbivore	Typically	Plant	Eater	Meat	Carnivore	Deer	Eat	Grass	Leaf
Q	0	0	0.176	0.176	0	0	0	0	0	0
S1	0.477	0.176	0.176	0.352	0.176	0	0	0	0	0
S2	0	0	0.176	0.352	0.176	0.477	0	0	0	0
S3	0	0	0	0	0	0	0.477	0.477	0.477	0.477

Lets rank them based on cosine similarity i.e

$$cos(S_i) = \frac{Q * S_i}{|Q| * |S_i|}$$

where, S_i represent the ith document and $cos(S_i)$ is cosine of angle between document S_i and the query vector.

	$\|Q\|$	$\|S_i\|$	Q*S$_i$	$cos(S_i) = \frac{Q*S_i}{\|Q\|*\|S_i\|}$
S1	0.249	0.666	0.093	0.56
S2	0.249	0.666	0.093	0.56
S3	0.249	0.954	0	0

Based on cosine values we can rank the documents as:

Rank 1 → $S1, S2$
Rank 2 → $S3$

5.Is the ranking given above the best?
Ans. The topic of Query is plant eaters in the sense we need documents related to Herbivores which are ”S1” and ”S3”, but we got document ”S2” which describes carnivores as higher rank than ”S3”. Hence we cannot conclude that the above ranking is the best.

6.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

7.(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. IDF means inverse document frequency of the word across a set of documents. This means, how common or rare a word is in the entire document set. The closer it is to 0, the more common a word is. This metric can be calculated by taking the total number of documents, dividing it by the number of documents that contain a word, and calculating the logarithm.

$$IDF_i = log(\frac{D}{df_i})$$

where:
IDF_i = Inverse Document Frequency of i^{th} term
df_i = number of documents which has i^{th} term
D = Total number of documents in the dataset

Given df_i = D as the word occurs in all the documents implies,

$$IDF_i = log(\frac{D}{df_i = D}) = log(1) = 0$$

(b) Is the IDF of a term always finite? If not, how can the formula for IDF be modified to make it finite?
Ans. For the given formula above if $df_i = 0$ i.e, query word does not occur in any document then the expression evaluates to $\log(\frac{D}{0}) \rightarrow \infty$. But this can be averted by adding a smoothing constant and modifying the IDF formula as,

$$IDF_i = log(\frac{D}{df_i + k})$$

where k is some constant usually taken as 1.

8.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. We can use Euclidean distance as another measure which measures the distance between the two vectors. Lesser the distance more relevant the document. But when compared to cosine similarity measure this performs poorly because its skewed and takes magnitude into consideration, so even if two vectors point in same direction we assume to get zero(highly related) as they are same but because of magnitude consideration it does not, where as cosine measure gives that they are highly related.

9.Why is accuracy not used as a metric to evaluate information retrieval systems?
Ans. Accuracy is defined as,

$$Accuracy = \frac{TruePositives + TrueNegatives}{TotalNumberofdocuments}$$

This measure is skewed because the number of true negatives is almost equal to total number of documents i.e, $TN \approx N$ due to inherent class imbalance in IR. This is because for a given query, very few documents would be relevant to it, while the rest would be irrelevant.
Consider 10^3 documents in total and 3 documents relevant to a query; Suppose the IR system retrieves one, the accuracy would still be $(10^3 - 1)/10^3$ which is a high value even though the performance of the IR system was poor.

10.For what values of α does the Fα-measure give more weightage to recall than to precision?
Ans. Fα-measure metric is given by,

$$F_{\alpha} = \frac{PR}{(1-\alpha)P + (\alpha)R} = \frac{R}{(1-\alpha) + (\alpha)R/P}$$

$$F_{\alpha=0} = R$$
$$F_{\alpha=1} = P$$

where P is Precision and R is Recall

For Recall to have high weightage the denominator containing Precision term should be minimised hence α should be less than .5 (i.e, $0 < \alpha < .5$)

11.What is a shortcoming of Precision @ K metric that is addressed by Average Precision @ k?
Ans. Precision @ K and Average Precision @ K are defined as:

$$Precision@k = \frac{Truepositives@K}{Truepositives@K + Falsepositives@K}$$

$$AveragePrecision@k = \frac{\sum_{n=1}^{K} P(K) * rel(K)}{Numberofrelevantdocuments}$$

where: rel(k) is an indicator function which is 1 when the item at rank K is relevant and P(k) is the Precision@k metric
As precision depends only on the number of relevant documents in top K results, a shortcoming is that it doesn't consider the position of the relevant items. Which is addressed by Average Precision, as it is a metric that evaluates whether all of the ground-truth relevant items selected by the model are ranked higher or not.

Precision is single-value metric based on the whole list of documents returned by the system. For systems that return a ranked sequence of documents, it is desirable to also consider the order in which the returned documents are presented. By computing a precision and recall at every position in the ranked sequence of documents, one can plot a precision-recall curve, plotting precision p(r) as a function of recall r.That is the area under the precision-recall curve.

12.What is Mean Average Precision (MAP) @ k? How is it different from Average Precision (AP) @ k ?
Ans. Mean Average Precision (mAP) is average precision (AP) averaged across a set of queries Q. If the set of relevant documents for a query q_j is $d_1, d_2, ..., d_{mj}$ 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:

$$AP = \frac{\sum_{n=1}^{m_j} P(R_{jk})}{m_j}$$

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

$$mAP = \frac{1}{|Q|}\sum_{j=1}^{|Q|}\frac{\sum_{n=1}^{m_j} P(R_{jk})}{m_j}$$

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].

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

Ans. For Information Retrieval system using Cranfield dataset nDCG is more appropriate measure. Though AP(Average Precision) takes relevance into picture, it just considers whether a document is relevant or not, but nDCG takes magnitude of relavance and ranking as well into picture.

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

$$nDCG_k = \frac{DCG_k}{IDCG_k}$$

Because of which more relevant documents appearing lower in the search results are penalized - the relevance value is logarithmically proportional to the position of the result. Thus we ensure using this measure that[2]:

- Highly relevant documents get higher rank in the search results.
- Relevant documents are ranked by there order of relevance and rank higher than non-relevant documents.

14.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

15.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 to 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 to 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. Observations:

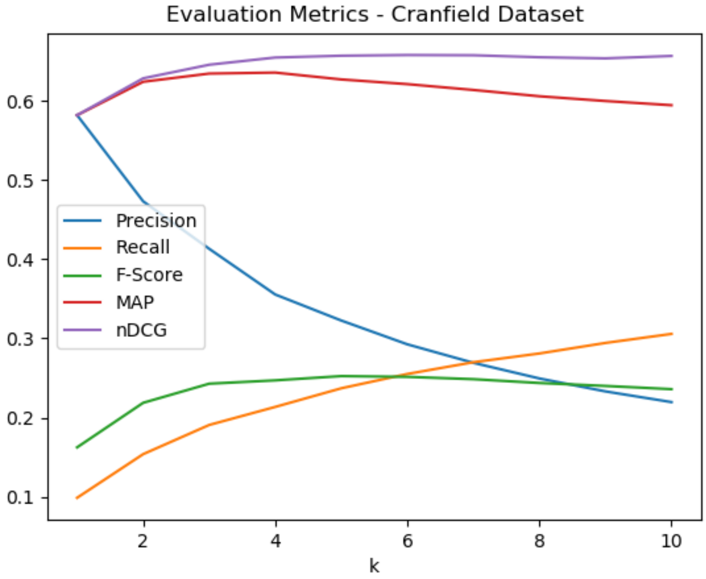

Figure 1: Evaluation Metrics vs k

We observe the following from the plots:

• Recall monotonically increases with k, as per the definition (TP/all relevant docs) denominator remains constant but as K increases the numerator might increase.

• 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.

• nDCG we notice that after k = 5 it saturates around .65 hinting that on average for all queries there are no more relevant documents retrieved from k = 5 till k = 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
3	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

Table 1: Evaluation metrics VS k used for plots

16.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	Query
3	what problems of heat conduction...
20	has anyone formally determined influence of joule...
92	given complete freedom in the design of airplane...
101	why does the incremental theory and deformation theory...
120	are the previous analyses of circumferential thermal buckling...

Table 2: Few 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	

Table 3: Queries for which IR system has the worst performance

17.Do you find any shortcoming(s) in using a Vector Space Model for IR? If yes, report them.
Ans. Using Vector Space Models for IR has its own shares of advantages and limitations.
Few shortcomings of Vector Space Models are:
- It makes the consideration of all words impractical: since each word is a dimension, considering all words would imply expensive computations in a very high-dimensional space[3].
- Lack of model flexibility: Each time we add a new term into the term space we need to recalculate all vectors.
- It assumes that all words are independent. which can result in retrieval of irrelevant documents and hence reduce precision.
- Semantic sensitivity; documents with similar context but different term vocabulary won't be associated, resulting in a "false negative match". Which can result in non retrieval of relevant documents and hence reduce recall
- The order in which the terms appear in the document is lost in the vector space representation.
- This system can easily be deceived by adding a keyword many number of times[4].

18.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. One simple way to go around this is assume the title of the document as one of the sentence of the documents and add it to the list to compute the TF-IDF measure, but this just results in slight increase in frequency of terms in the title and does not give much preference. To give special preference we can add the title sentence higher number of times.
Other method is to calculate TF-IDF vector measure for the titles of all documents and simply add it to their document TF-IDF vectors. Measure of relevance of title can be taken care by scaling the title TF-IDF appropriately.
If we want to weigh the title contribution three times that of the document then in the method suggested above we could scale the title TF-IDF measure 3 times and add it to document TF-IDF vector.

19.Suppose we use bigrams instead of unigrams to index the documents, what would be its advantage(s) and/or disadvantage(s)?
Ans. Advantage of using Bigrams is that it captures co-occurrences and collocations which can reduce false positives of the retrieved results. For example, consider a query containing the collocation "foot ball" in such case our bigram based IR model will most likely not retrieve documents containing "basket ball" as it has captured the necessity that foot and ball occur together.
But a big disadvantage of using bigrams is its computational complexity when calculating bigram frequency as the matric is conditioned on pair wise occurence of terms in both query and documents. For example if 100 documents have 10^5 terms then unigram matrix is 100*10^5 but bigram matrix is 100*(10^5C_2) and for this matrix we have to find the term frequency.

20.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. An implicit feedback can be deduced from user response to our retrieval results. The result that user selects will get higher relevance, and remaining results above this should be reduced in relevance accordingly. But if user does not select any of the document it can be treated as negative feedback for all results retrieved and their relevance score should be penalized accordingly.

Other signals such as time spent of the document selected, page browsing and scrolling actions can also be taken into account though they might be difficult to implement[5].

REFERENCES:

[1] Evaluation of results (mAP vs AP):

`https://nlp.stanford.edu/IR-book/html/htmledition/evaluation-of-ranked-retrieval-results-1.html`

[2] nDCG measure information:

`https://en.wikipedia.org/wiki/Discounted_cumulative_gain`

[3] Vector Space Models:

`https://www.sciencedirect.com/topics/computer-science/vector-space-models`

[4] Shortcomings of Vector Space Models:

`https://en.wikipedia.org/wiki/Vector_space_model`

[5] Relevance Feedback article:

`https://en.wikipedia.org/wiki/Relevance_feedback`