Assignment 1 - Part 2

CS6370: Natural Language Processing

JUL-NOV 2021

INSTRUCTOR: DR. Sutanu Chakraborty

PRAGALBH VASHISHTHA - MM19B012

SAARTHAK MARATHE - ME17B162

Answer 1:-

	herbivore	typically	plant	eater	meat	carnivore	deer	eat	grass	leaf
S1	1	1	1	2	1	0	0	0	0	0
S2	0	1	1	2	1	1	0	0	0	0
S3	0	0	0	0	0	0	1	1	1	1

Answer 2:-

$$tf - idf_{t,d} = tf_{t,d} * idf_{t}$$

- The above equation assigns a weight to the term t in the document d
- $idf_{t} = log\frac{N+1}{df+1}$ which is the inverse document freq of t and df being the no of docs in the collection that contains the term t
- N = total no of documents

- $tf_{t,d}$ = term freq of t in the document d

Terms	Counts (tf)			df	N/df	idf	Weights (tf*idf)		
	S1	S2	S3				S1	S2	S3
herbivore	1	0	0	1	3	0.693	0.693	0	0
typically	1	1	0	2	1.5	0.288	0.288	0.288	0
plant	1	1	0	2	1.5	0.288	0.288	0.288	0
eater	2	2	0	2	1.5	0.575	0.575	0.575	0
meat	1	1	0	2	1.5	0.288	0.288	0.288	0
carnivore	0	1	0	1	3	0.693	0	0.693	0
deer	0	0	1	1	3	0.693	0	0	0.693
eat	0	0	1	1	3	0.693	0	0	0.693
grass	0	0	1	1	3	0.693	0	0	0.693
leaf	0	0	1	1	3	0.693	0	0	0.693

Answer 3:-

Following the previous deductions, Doc 1, Doc 2 will be retrieved for queries: plant, eater

Answer 4:-

Using the weights retrieved in the question 2’s table

	herbivore	typically	plant	eater	meat	carnivore	deer	eat	grass	leaf
S1	0.693	0.288	0.288	0.575	0.288	0	0	0	0	0
S2	0	0.288	0.288	0.575	0.288	0.693	0	0	0	0
QUERY (Q)	0	0	0.288	0.288	0	0	0	0	0	0

Calculating the cosine similarity:

$$cossim(S1, Q) = \frac{S1.Q}{|S1||Q|} = 0.593$$

$$cossim(S2, Q) = \frac{S2.Q}{|S2||Q|} = 0.593$$

Each of the S1,Q,S2 are represented as vectors in the above two equations. We don't take S3 into consideration as it is not retrieved for the given query.

Answer 5:-

No, the ranking system above isn’t the best because it tends to rank both the documents simultaneously. Ideally, the third sentence which is relevant should be retrieved as well but the given system does not pick it up. Thus, it is not the best ranking system for the given case.

Answer 6:-

Python code files attached with this pdf.

Answer 7:-

a)The idf of a term t_i that appears in every document of the corpus, idf_i is equal to 0. This is because idf is a term frequency measure which gives a larger weight to terms which are less common in the corpus.

b) The idf is not always finite. For example any word (say from a dictionary) that does not occur in the corpus has infinite idf.

To make the idf finite a small smoothing factor can be used. For example: idf_t = $log\frac{N+\alpha}{n+\alpha}$. N is the number of documents total and n is the document frequency of the term. ɑ. Is the smoothing factor which we have used as 1.

Answer 8:-

The following are other distance measures that can be used to compare vectors:-

- Euclidean distance:-
    - Defined as the L_2 norm between the vectors. Unlike cosine similarity, it is not scale invariant
    - Due to large magnitude of documents compared to queries, the distance between a document na d a query will be high and is thus useless for retrieval
- Manhattan distance:-
    - It is defined as the L_1 norm between the two vectors.
    - Its is not scale invariant and also will be high due to large magnitude of documents
- Jaccard Similarity Index
    - It is the ratio of the intersection of non-zero dimensions between vectors with the union of non-zero dimensions between vectors. Higher dimensions shared imply higher similarity
    - However the relative frequency as seen in idf is ignored here, hence it performs suboptimally compared to Cosine similarity
- Hence cosine-similarity is the best for our Information retrieval applications.

Answer 9:-

Accuracy is defined as :- $\frac{True\ positives + true\ negatives}{total\ number\ of\ documents}$.

During information retrieval, the number of true negatives will be large, it will be of similar magnitude to the total number of documents. Hence the accuracy is bound to be close to one even for bad IR systems, even zero positives may lead to a high accuracy. Hence it is not an effective measure.

Answer 10:-

$F_{\alpha} = \frac{Precision*Recall}{(1-\alpha)Precision+\alpha Recall}$ as α → 1, F_{α} → Precision.

as α → 0, F_{α} ⟶ Recall. For values of as α ∈ [0,0.5) weightage is more for recall.

Answer 11:-

Precision @k (P@k) counts the number of relevant results in the top k retrieved documents. This however fails to capture the relative rank of the documents among the top k.

Average precision is defined as $AP@k = \frac{\sum_{j=1}^{k}(P@j*relv(j))}{number\ of\ relevant\ docs}$ where relv(j) =1 if document is relevant 0 otherwise.

Hence, it is a much better measure as it captures the positions of relevant documents.

Answer 12:-

The Mean Average Precision@k (MAP@k) over Q (total no of queries) is:

$$MAP@k = \frac{\sum_{q=1}^{Q} AP@k_q}{Q}$$

MAP for a set of queries is the mean of the average precision scores for each query for a set of queries but whereas AP@k is defined for every single query that is inputted. Therefore, MAP@k is a better indicative of the performance of an IR system.

Answer 13:-

nDCG is better than AP for the cranfield dataset. Reasons are entailed below:-

Human relevance judgements include ranking how relevant a document is to it's query. nDCG does that by placing higher relevance documents higherwhile AP only mentions whether a query is relevant.

AP is designed for only binary results and doesn't take into account fine-grained numerical ratings.

While calculating AP, the fine- ratings are manually thresholded to make binary relevance predictions, therefore introducing bias in the evaluation metric. ALso the fine grained information is lost.

Answer 14:-

Python code files attached with this pdf.

Answer 15:-

The graph is attached in the output folder - eval_plot.png

- F-score increases initially with k and then tends to flatten out. F-score is generally used to compare various models to break the precision-recall trade-off. The flattening can indicate that good number of documents have been retrieved
- Recall is increasing monotonically with k as the number of retrievals can only increase with the number of documents
- MAP increases monotonically with k
- Precision initially increases and then goes through a decrease and then increases again to slightly decrease after that

- nDCG follows a similar pattern as that of precision but it tends to be more monotonous than precision in terms of increase trend

Answer 16:-

There are few queries for which the search engine’s performance is not as expected. There are few terms with special characters like ‘-’ in the index. One example could be ‘-rise’ which does not match when we put ‘rise’ in the query. This hampers the overall performance of the engine.

Answer 17:-

Shortcomings in using a vector space model for IR:

- High computational intensity and has high latency
- Need to recalculate vectors for each addition of new term
- Difficulty in modelling of sequences with terms occurring in documents
- Assumes orthogonality of all the terms which is many times not true. This creates a problem with sentences with diff vocabularies and similar content
- Difficulty in processing long documents and causes problems in calculating cosine similarity with high dimensionality

Answer 18:-

When a dataset similar to Cranfield dataset is being used, we can include the title by adding the TF-IDF representation of the same while using the top down knowledge of the document. For the title to have more contribution during the information retrieval (let's say 5 times), we can multiply the TF-IDF representation of the title accordingly (5 times here). For this to work, the document needs to be smaller in size so that the worth of the title’s representation does not go unnoticed compared to the rest of the document. Directly adding the weighted representation of the title to the representation of the document will mean the title makes very little contribution. Higher weights might put over-emphasis on the titles. Hence, striking a proper balance becomes tough. It becomes even tougher for a dataset with varying lengths of documents. We have to look for more novel techniques in such a case.

Answer 19:-

Advantages:

- Vector space representations are better at modeling sequence.
- As bigrams model context and sequencing better, the precision is higher than that obtained while using monograms.

Disadvantages:

- The recall is lower than that obtained while using monograms.
- Vector space is very high dimensional. Latency will increase tremendously from the unigram model
- Simple vector space model assumes that dimensions are orthogonal. Issue exists even with unigrams, it is sort of easier to interpret unigrams as orthogonal building blocks that make up a sentence. Issue blows up multifold with our bigram model as multiple bigrams contain the same term (unigram)

Answer 20:-

- Based on the user's query history we can infer the user's satisfaction or dissatisfaction with the results. If the query is changed/reformed after the results, it shows dissatisfaction and if the results of the query are continued to be explored it shows positive implications
- When the person engages with a search result (e.g., by clicking on it), the search engine treats the engagement as implicit positive feedback.
- Observation of the amount of time the user spends on each result. More time spent on a certain result tells about the positive reinforcement of the content provided and the indirect implications of the same