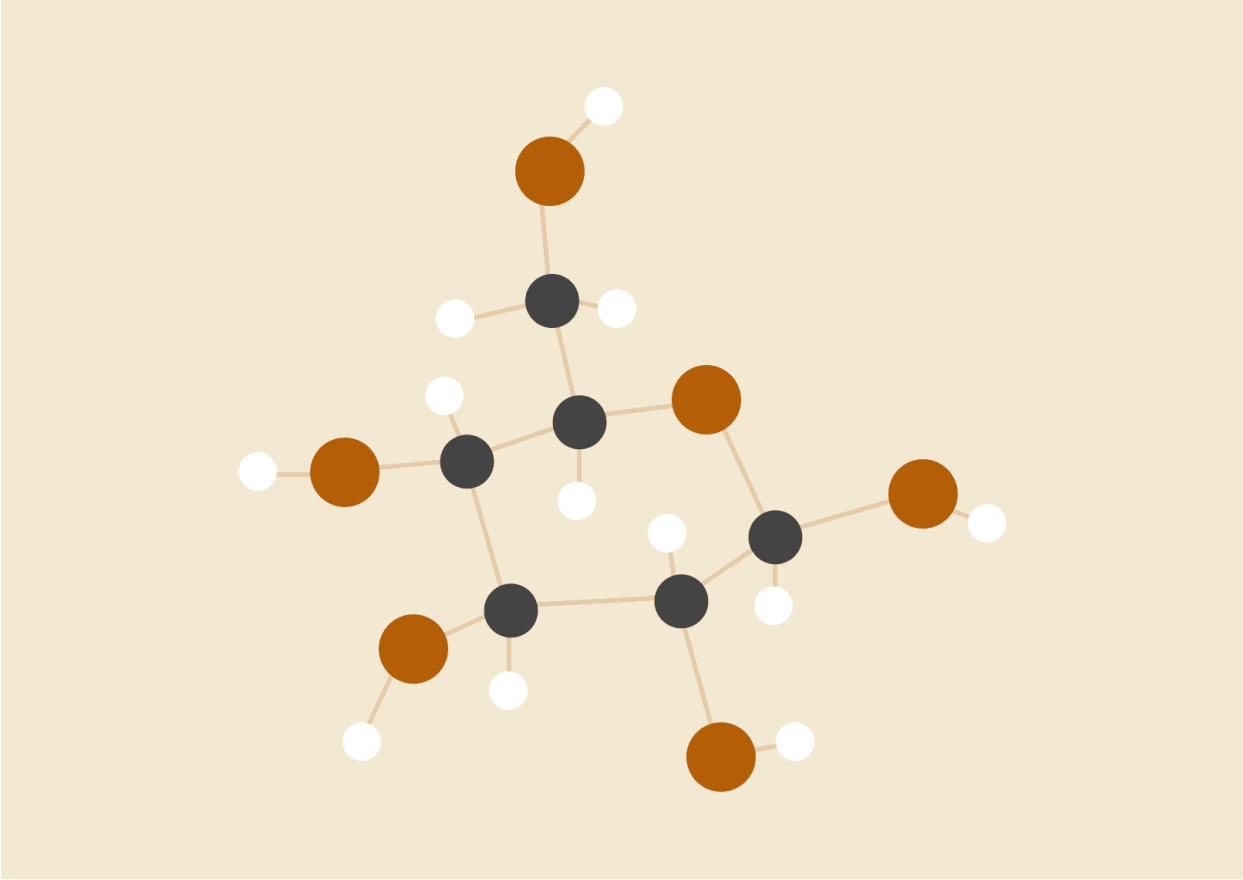
CS6370 Assignment 2

Natural Language Processing



B Midhun Varman EE18B113

Dibyendu Mandal EE18B108

20.03.2023

INTRODUCTION

The goal of the assignment is to build a search engine from scratch, which is an example of an information retrieval system. In the class, we have seen the various modules that serve as the building blocks of a search engine. The first part of the assignment involved building a basic text processing module that implements sentence segmentation, tokenization, stemming/lemmatization and stopword removal. This module involves implementing an Information Retrieval system using the Vector Space Model. The same dataset as in Part 1 (Cranfield dataset) will be used for this purpose.

Questions

Question 1

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).

Answer

Inverted Index

Plant :S1,S2

Typically : S1,S2

Eaters : S1,S2

Meat eaters : S1,S2

Herbivores : S1

Carnivores : S2

Deers : S3

Grass : S3

Eat: S3

Leaves : S3

Question 2

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.

Answer

Tf-Idf Formula used is ,

$f_{i,j} \times \log(N/n_i)$

Where ,

$f_{i,j}$ is frequency of word i in document j

N is total no of documents
n_i is no of documents in which word i occurs

Note : Tf-idf is bound to fail since there are 2 words which have representation in all documents which results in its value being 0.

```
IDF of:

           meat: 0.17609125905568124

     Carnivores: 0.47712125471966244

      typically: 0.17609125905568124

     Herbivores: 0.47712125471966244

            eat: 0.47712125471966244

         leaves: 0.47712125471966244

          grass: 0.47712125471966244
```

```
Deers: 0.47712125471966244

eaters: 0.17609125905568124

plant: 0.17609125905568124
```

index	meat	Carnivores	typicallly	Herbivores	eat	leaves	grass	Deers	eaters	plant
0	0.029348543175946873	0.0	0.029348543175946873	0.07952020911994373	0.0	0.0	0.0	0.0	0.058697086351893746	0.029348543175946873
1	0.029348543175946873	0.07952020911994373	0.029348543175946873	0.0	0.0	0.0	0.0	0.0	0.058697086351893746	0.029348543175946873
2	0.0	0.0	0.0	0.0	0.11928031367991561	0.11928031367991561	0.11928031367991561	0.11928031367991561	0.0	0.0

Question 3

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

Query "plant eaters", inverted Index S1 and S2 would be retrieved

Question 4

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

Query "plant eaters", cosine similarity would be high and equal for both S1 and S2 as they are of equal length and same words in same frequency and equal different words, 0 for S3

Order: S1=S2>S3

Question 5

5. Is the ranking given above the best?

Cosine similarity won't work very well because it's only based on whether the rank at position i is the same in vector 1 and vector 2.

For instance the vectors `[3,2,4,1,5]` vs `[2,3,5,1,4]` will have very low similarity because 4 positions out of 5 are different, even though there are only two swaps between (2,3) and (4,5).

Question 6

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.

Implementation done in informationRetrieval.py for further information

Question 7

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?

a. 0(zero) since (N/n_i) will be 1 in this case

b. IDF term's infinity

Idf term can be infinity if we are considering a word which is not present in any documents this can be solved by using the formula as log((N+1)/(n_i +1))

Question 8

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.

Euclidean Distance

$$\|a-b\| = \sqrt{\|a\|^2 + \|b\|^2 - 2a^T b} = \sqrt{2 - 2\cos(\theta_{ab})}$$

One measure we can use is Euclidean distance between 2 vectors

And consider 2 vectors are similar if distance between them is less

One way it is better than cosine similarity is it can differentiate between colinear vectors but it gives some false negative based on magnitude of vectors (eg a document with similar composition but with large size will give more distance compared to smaller even though both are similar)

We can also use Improved cosine similarity

$$ISC(x,y) = \frac{\sum_{i=1}^{m} \sqrt{x_i y_i}}{\sqrt{(\sum_{i=1}^{m} x_i)} \sqrt{(\sum_{i=1}^{m} y_i)}}$$

From Improved sqrtcosine similarity measurement by Sahar Sohangir and Dingding Wang Journal of Big Data volume 4, Article number: 25 (2017)

https://rdcu.be/b3br1

Another measure we can use is spearman coefficient which mitigates by ranking the values before applying Pearson correlation.

The Spearman correlation coefficient is defined as the Pearson correlation coefficient between the rank variables.[3]

For a sample of size *n*, the *n* raw scores X_i, Y_i are converted to ranks $\mathrm{R}(X_i), \mathrm{R}(Y_i)$, and r_s is computed as

$$r_s = \rho_{\mathrm{R}(X),\mathrm{R}(Y)} = \frac{\mathrm{cov}(\mathrm{R}(X), \mathrm{R}(Y))}{\sigma_{\mathrm{R}(X)} \sigma_{\mathrm{R}(Y)}},$$

where

ρ denotes the usual Pearson correlation coefficient, but applied to the rank variables,
$\mathrm{cov}(\mathrm{R}(X), \mathrm{R}(Y))$ is the covariance of the rank variables,
$\sigma_{\mathrm{R}(X)}$ and $\sigma_{\mathrm{R}(Y)}$ are the standard deviations of the rank variables.

Only if all *n* ranks are *distinct integers*, it can be computed using the popular formula

$$r_s = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)},$$

where

$d_i = \mathrm{R}(X_i) - \mathrm{R}(Y_i)$ is the difference between the two ranks of each observation,
n is the number of observations.

Question 9

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

Accuracy = tp / (N), where N is the total number of documents However, 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 ~ N). This is because for a given query, very few documents would be relevant to it, while the rest would be non relevant. For example, if there are 10^4 documents in total and 3 documents relevant to a query, of which 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.

Question 10

10. For what values of α does the Fα -measure give more weightage to recall than to precision?

Fα score for recall Having alpha between 0 and 0.5 gives more weightage to recall

Question 11

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

For systems that return a ranked sequence of documents, it is desirable to also consider the order in which the returned documents are presented. This is not considered in precision but average precision

$$AP@n = \frac{\sum_{k=1}^{n}(P(k) \times rel(k))}{number\ of\ relevant\ documents}$$

Considers it.

Question 12

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

Mean Average Precision vs Average Precision

The mean of all the Average Precision scores calculated for all the queries is the Mean Average Precision it quantifies the performance of the model as a whole, independent of any single query. Here, the mean refers to mean across queries so MAP is better than AP

as MAP gives performance of a system.

Question 13

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

AP vs nDCG

For Cranfield dataset nDCG is more important compared to AP as nDCG unlike AP also considers how relevant a document is .We have this information in qrels.json as 'position' .

AveP is designed for binary (relevant/non-relevant) ratings and doesn't take into account fine-grained numerical ratings for relevance.

In the calculation of AveP, we might want to threshold the fine-grained ratings to make binary relevance predictions, therefore introducing bias in the evaluation metric because of the manually set threshold. Besides, we are also throwing away the fine-grained information.

Question 14

14. Implement the following evaluation metrics for the IR system:

(a) Precision @ k

(b) Recall @ k

(c) F-Score @ k

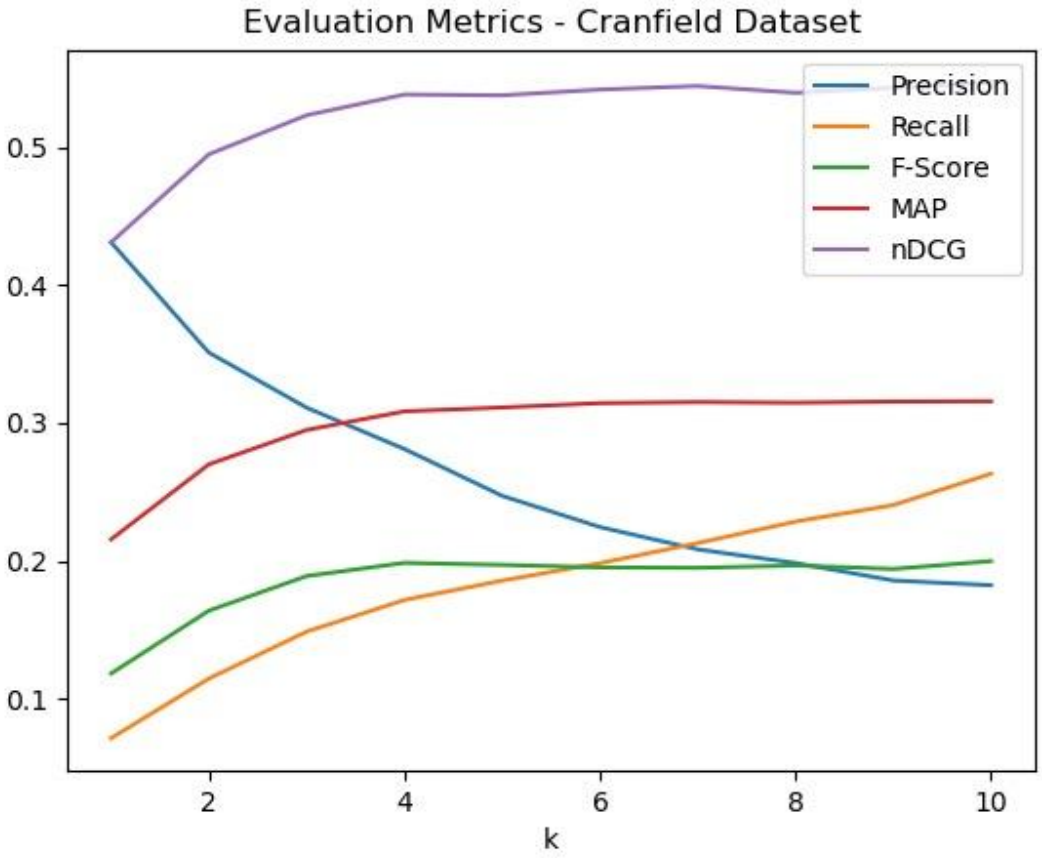
(d) Average Precision @ k

(e) nDCG @ k

Implementation done in evaluation.py

Question 15

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 a 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.



Observations

- From this graph we can see that except precision every metric improves as we increase documents retrieved(k)
- Precision decreases monotonically with k. This is as expected for a well functioning IR system as the density of relevant documents is expected to decrease with increase in retrieved documents. It is important to note that monotonicity is not a necessity for precision, rather indicative of our IR system's performance.
- Recall monotonically increases with k as expected. As the total number of relevant documents is a constant, the number of relevant documents retrieved can only increase with k.
- We can also see that after 6 there isn't much improvement in Fscore, AP, nDCG so we can say that for reasonable performance 6 documents are enough after which

it is difficult to get better metrics

Question 16

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

Answer:

Queries for which performance is not expected

These are some examples for which results are not good as expected

For query "what chemical kinetic system is applicable to hypersonic aerodynamic problems ."one of the expected answer for this(according to qrels.json) is document titled "chemical kinetics of high temperature air ."

The document with id = 552 does not show up in our search engine but most of our results are related to keywords 'hypersonic' 'aerodynamic' eg document titled "hypersonic flight and the reentry problem ."(doc_ID=1379) is best result.

Similarly for query "basic dynamic characteristics of structures continuous over many spans ."(query_num = 102) expected answers(documents titled "free vibrations of continuous skin stringer panels ."(doc_ID=728) and "stresses in continuous skin stiffener panels under random loading ."(doc_ID=729)) are missing top result is "a characteristic type of instability in the large deflections of elastic plates ."(doc_ID=1379) here also keywords like 'characteristic' and 'structures' .

From these examples we can infer that our model mostly matched some keywords which is not the answer for these queries

Question 17

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

Shortcomings in Vector Space Model

- It implicitly assumes that the terms are orthogonal to each other. This is often not the case.
- Syntactical (grammar) contexts are not taken into account.
- Synonyms are not assumed as same Only some keywords which occur frequently are given importance over others as mentioned in previous examples

- Dependencies between words in both query and document are not taken care of as it is just a bag of words model.
- We are assuming words are independent dimensions which is not true since words also have relations between them.
- Every time we add a new term to the term space, we need to recalculate all the vectors.
- Very long documents lead to difficulties in calculating the similarity measure. i.e., the issue of small dot products and high dimensionality.
- High latency, i.e., it is very calculation intensive and takes a lot of processing time.

Question 18

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?

We can use the top down knowledge that the documents are not very long. In such a case, the contribution from the title can be included by adding the TF-IDF representation of the title to the TF-IDF representation of the document.

In such a case, the contribution from the title can be included by adding the TF-IDF representation of the title to the TF-IDF representation of the document. If we want the title to have three times the contribution of the text in the body, we can add 3 times the TF-IDF representation of the title to the representation of the document, therefore giving more weight to the title

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.

Question 19

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

Answer:

Bigrams over unigrams

One advantage of using bigrams over unigrams is some of the dependencies between words are captured over unigrams Which may increase precision But a disadvantage of using bigrams is no of dimensions will be very and most will be zero which might result reduced recall(now document should have same order of words as query if it is to be recommended) .There is also the issue of orthogonality. Our simple vector space model assumes that dimensions are orthogonal to each other. Even though this issue exists even with unigrams, it is sort of easier to interpret unigrams as orthogonal building blocks that make up a sentence.

Question 20

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.

Implicit Feedback include:

Recording which documents user is opening from our search results which can be considered as relevant documents in calculating AP, precision, recall, Fscore. This includes clickthrough, search query history

• Reading Time Observation of the amount of time the user spends on each result. It seems reasonable that a user would spend more time on more relevant results

REFERENCES

1. Stanford NLP article on Inverted Index
2. Wikipedia article on Evaluation Metrics
3. Cornell University CS6740 lecture notes
4. Medium article on Relevance Feedback
5. Medium article on Mean Average Precision
6. Medium article on Similarity Metrics