Assignment Part2

M.Nikhil(EE17B138),Anirudh(EE17B043)
25th February 2020

1. Inverted Index
   Cat : docA, docB, docC
   Dog : docA, docB
   Animal : docB, docC, docA
2. Tf-idf
   Formula used is
   $f_{i,j} \times \log(\frac{N}{n_i})$
   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 whic word i occurs
   Note : Tf-idf is bound to fail since there are 2 words which have representation in all documents which results in it's value being 0.

   docA = 0(cat) + 0.528(dog) + 0(animal)

   docB = 0(cat) + 0(dog) + 0(animal)

   docC = 0(cat) + 0.528(dog) + 0(animal)

   Q(cat) = 0

   Q(dog) = 0.176(dog)

   Q(animal) = 0

3. Query "dog", inverted Index
   docA and docC would be retrieved
4. Query "dog", cosine similarity
   1 for both docA and docC
   0 for docB
   Order: docC=docA>docB
5. Implement an Information Retrieval System for the Cranfield Dataset using the Vector Space Model.
   Implementation done in informationRetrieval.py for further information read README.md in the folder
6. Question on IDF
   a. What is the IDF of a term that occurs in every document?
      0(zero) since $\frac{N}{n_i}$ 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

$$\log\left(\frac{1+N}{1+n_i}\right)$$

7. Euclidian Distance

One measure we can use is Euclidian distance between 2 vectors

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

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 sqrt-cosine similarity measurement by Sahar Sohangir and Dingding Wang Journal of Big Data volume 4, Article number: 25 (2017)

https://rdcu.be/b3br1

8. Accuracy

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.

9. F_{alpha} -score for recall

Having alpha between 0 and 0.5 gives more weightage to recall

10. Precision vs average precision

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

11. 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.

12. 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'

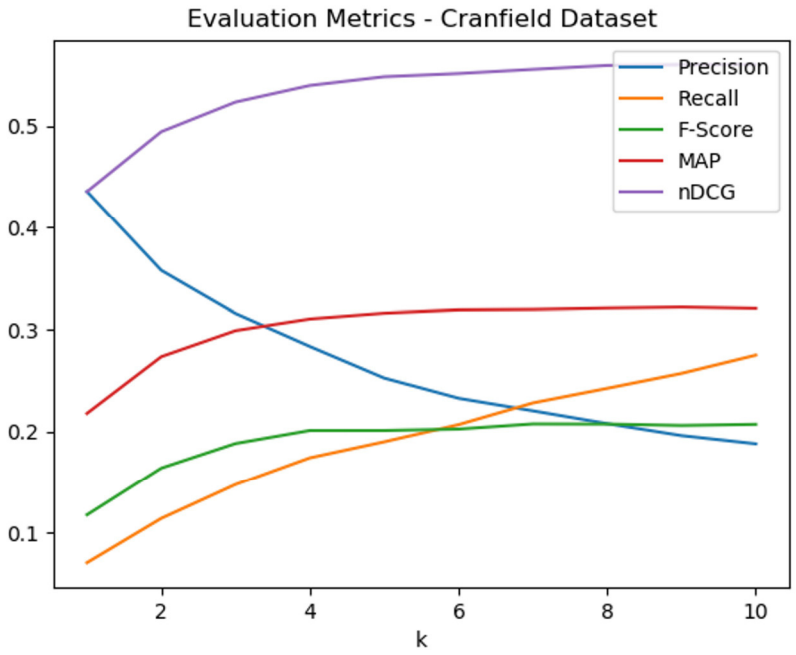
13. Implementation

Implementation done in evaluation.py and some functions needed for it are written in util.py for further

information read README.md in the folder

Note: For calculating nDCG scores relevance order is [4,3,2,1,0] with 4 being most relevant and 0 being not relevant.

14. Evaluation Metrics graph



Observations

- From this graph we can see that except precision every metric improves as we increase documents retrieved(k)
- We can also see that after 6 there isn't much improvement in F-score, AP, nDCG so we can say that for reasonable performance 6 documents are enough after which it is difficult to get better metrics

15. Queries for which performance is not expected

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

a. For query "what chemical kinetic system is applicable to hypersonic aerodynamic problems ."(query_num = 5) one of the expected answer(according to qrels.json) is document titled "chemical kinetics of high temperature air ."
(doc_ID =552) this document 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 re-entry problem ."(doc_ID=1379) is best result.

b. 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.

16. Shortcomings in Vector Space Model

- 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.

These are some shortcomings which we found.

17. Giving importance to Titles of documents

First we can calculate document vector(using Tf-idf method) $\vec{a}$ then calulate title vector in same way $\vec{b}$ then use the following vector as final vector

$$3\frac{\vec{b}}{|\vec{b}|}+\frac{\vec{a}}{|\vec{a}|}$$

This vector gives similarity similarity scores in which title component contributes 3 times of document component

18. 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)

19. Implicit Feedback

One way for implicit feedback is recording which documents user is opening from our search results which can be considered as relevant documents in calculating AP, precision, recall, F-score.