COMPSCI 546: Assignment 2

ppachpute@umass.edu

October 2019

1 Questions

1. Description of the system, design tradeoffs, questions you had and how you resolved them, etc. List the software libraries you used, and for what purpose.

    - Created a Scorer interface that specifies a score method. BM25, Jelinek Mercer and Dirichlet implements this interface and each one of them has its own way of calculating score according to their algorithm.
    - To make the score function consistent accross all the classes, I create a method signature that takes union of parameters required for all the three methods. Hence not all parameters are used in each of the method.
    - Because of this, adding a new score function is very easy. We just have to implement the score function and pass that in the Document at a time model.

2. Do you expect the results for Q6 to be good or bad? Why?

    - I don't expect to see results of the Q6 query to be good.
    - Reason being our retrieval models use unigram probabilities. That is, it treats each query term independently. The query in Q6 "to be or not to be" contains the words which are found in most of the scenes and that too frequently. Moreover, we will try to rank documents based on individual query terms rather than the phrase.
    - Hence our model has no way to predict "hamlet:2.0" with high probability.

3. Do you expect the results for a new query setting the scene to be good or bad? Why?

    - Since the phrase "setting the scene" does not occur in the collection, I assume good result here would mean the document that contains something like "setting the" or variation of the 3 given words. Following response is based on this assumption.

- I don't expect to see results of the new query "setting the scene" to be good.
- Reason being query terms "the" and "scene" occur too frequently in the collection. Therefore, presence of such words as the query terms doesn't really help in retrieving the required document.

4. Look at the top ten results for Q3 for each QL and BM25 (maximum of 30 results in total, union the sets of scene identifiers).

    - Attached judgments.txt in zip file

5. What will have to change in your implementation to support phrase queries or other structured query operators?

    - As of now each word is considered individually (unigram model) therefore when evaluating query we never look at the other query term to calculate score of current query term.
    - One way to answer the phrase queries will be to upgrade our model to n-gram. That is, each inverted index would contain list of postings for 2 words, 3 words etc.
    - Other way to evaluate phrase queries or structured query operators would be to consider query term's position as well when calculating the score of the document instead of just their raw count. Just like we did in case of Dice's coefficient we would have to use actual positions from the posting list.

6. How does your system do? Which method appears to be better? On which queries? Justify your answer.

    - In general I have seen that model is returning relevant results
    - Of all the models I have noticed BM25 is returning better results (However, This is based on randomly collected subset of results)
    - For query "hope dream sleep" BM25 returned the results where these 3 words appeared in relevant context whereas for other models either some of the terms were not present or they appeared in the document but within different context.
    - Dirichlet smoothing is generally more effective than Jelenik-Mercer for shorter queries
    - None of them produced good result for phrase queries like "to be or not to be"