Proximity Enabled Search (Extra Credit Task)

1. Introduction

The goal of this task is to design and build our own proximity enabled search information retrieval systems, evaluate and compare their performance levels in terms of retrieval effectiveness. To do this task various assumptions were given to us.

- Terms should appear in the matching document in the same order as the query.
- Allow a maximum proximity of three terms, that is, adjacent query terms can be separated by no more than 3 terms in the matching document.
- Documents with terms appearing closer to each other are deemed better matches.

2. Implementation

The proximity-enabled retrieval model (Proximity Model) is built on top of the BM25 model.
The idea is to have 2 scores for every query term.

- Base score (from BM25)
- Proximity score

Algorithm

The base scores of every query term, computed from BM25, gives us an idea about how significant every query term is. We compute proximity scores for every occurrence of each query term by using a scoring technique as illustrated below.

- Make a dictionary where query terms (q) are keys and whose values are lists of adjacent query terms (qadj) as it appears in the query.

Note - A query term can have multiple adjacent query terms. For example, for a query "a boy a girl", both "boy" and "girl" will appear in its adjacent list

- Scan every query term in the document from left to right
  - If qadj appears within a specified window,then
    - **Offset of qadj:** Position difference of q and qadj
    - **Update proximity_score _q:** proximity_score_q + (window-size - offset) * base_score_qadj * base_score_q
    - **Update proximity_score_qadj**: proximity_score_qadj" + proximity_score_q + (window-size - offset) * base_score_q * base_score_qadj
  - If none of the qadj appears in the window of q then update proximity_score_q as "proximity_score_q- (window-size * base_score_q).

Explanation

We are considering 3 important factors when scoring.

- Proximity score of a query term is increased only based on the base score (its significance) of that query term and its adjacent query term. This means that a phrase “a machine” will garner less proximity score than the phrase “turing machine” (assuming that “a” has less base score than “turing”)
- The lengthier the phrase the higher the proximity score, on the precondition that both phrases have the same set of query terms. For example, “turing machine turing machine” will have more score than just “turing machine”. This is because the query term “machine” appearing in the 4th position will be rewarded by both “turing” term occurrences.
- If a query term appears alone, we penalize it.

3. Experimental Results:

For experimenting, a fabricated document containing text -

“dummy new york city dummy dummy dummy dummy dummy dummy dummy dummy dummy dummy dummy city new york dummy dummy dummy dummy dummy dummy dummy dummy dummy dummy dummy dummy dummy dummy dummy dummy dummy city york new dummy dummy dummy dummy dummy dummy dummy dummy dummy dummy dummy dummy dummy dummy dummy dummy new city york dummy dummy dummy dummy dummy dummy dummy dummy dummy dummy dummy york new city dummy dummy dummy dummy dummy dummy dummy york city new dummy dummy dummy dummy dummy dummy dummy dummy dummy dummy dummy dummy dummy new york city new york city new york city dummy dummy dummy dummy dummy dummy dummy dummy dummy icity york new york city dummy dummy dummy dummy ”

was added to the CACM corpus and tested with the query “new york city”. The proximity model scored the query terms as follows,

Position	Term	Base Score	Proximity Score	Comments
* 0025	new	5.0588	282.6972	The highest score “new york city” phrase can get is 951.47
* 0026	york	13.9704	951.4780	
* 0027	city	11.9678	951.4780	
* 0039	city	11.9678	-59.8389	The term “city” is penalized as it appeared out of order. However, the phrase “new york” was still scored, but not as high as the first set
* 0040	new	5.0588	282.6972	
* 0041	york	13.9704	282.6972	
* 0058	city	11.9678	-59.8389	None of terms appear in query order. So, all terms are penalized
* 0059	york	13.9704	-69.8522	
* 0060	new	5.0588	-25.2942	
* 0077	new	5.0588	212.0229	“new” and “york” are not penalized as they appear within specified window. “city” is penalized
* 0078	city	11.9678	-59.8389	
* 0079	york	13.9704	212.0229	
* 0091	york	13.9704	501.5856	“city” and “york” are not penalized as they appear within specified window. “new” is penalized
* 0092	new	5.0588	-25.2942	
* 0093	city	11.9678	501.5856	
* 0101	york	13.9704	668.7808	“york” and “city” appears closer than the last set and hence have got higher score
* 0102	city	11.9678	668.7808	
* 0103	new	5.0588	-25.2942	
* 0117	new	5.0588	282.6972	Single query term matches more than 1 occurrence of adj. term. For example, “york” at position 118 matches “city” at 119 and 122
* 0118	york	13.9704	951.4780	
* 0119	city	11.9678	951.4780	
* 0120	new	5.0588	282.6972	
* 0121	york	13.9704	951.4780	
* 0122	city	11.9678	951.4780	
* 0123	new	5.0588	282.6972	
* 0124	york	13.9704	951.4780	
* 0125	city	11.9678	951.4780	
* 0136	york	13.9704	334.3904	Similar to last set
* 0137	new	5.0588	282.6972	
* 0138	york	13.9704	951.4780	
* 0139	city	11.9678	1285.8684	

4. Results

The global measures Mean Average Precision (MAP) and Mean Reciprocal Rank (MRR) for runs with and without stopping results are shown below.

- **No Stopping:**

MAP 0.353415616857

MRR 0.346153846154

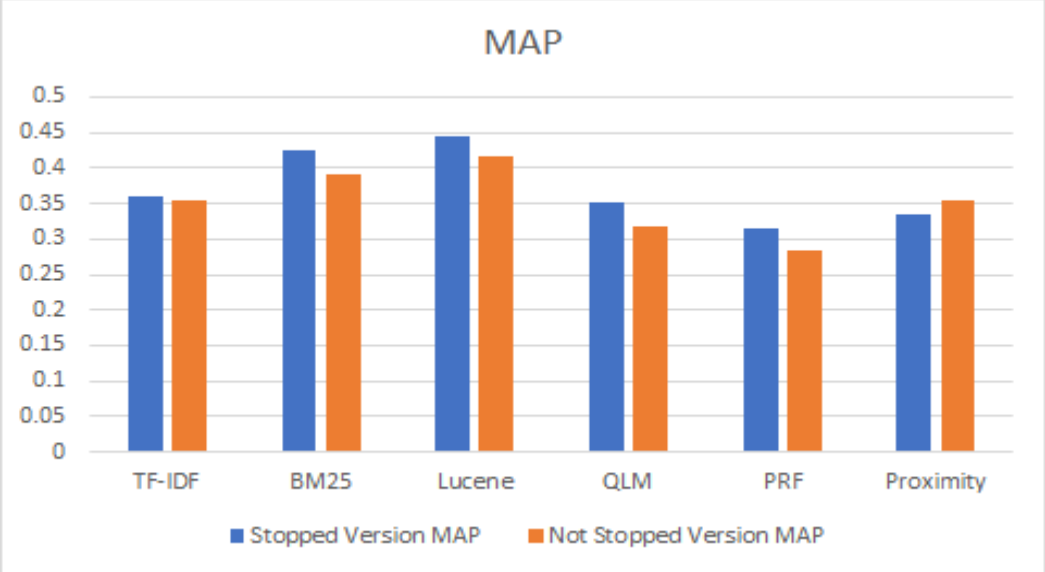
- **With Stopping:**

MAP 0.335843356878

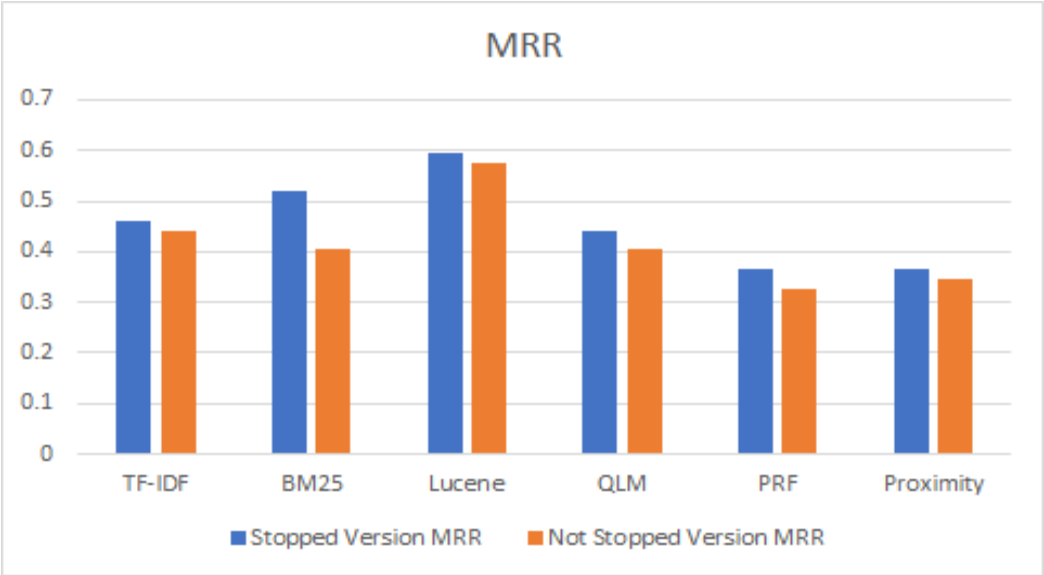
MRR 0.365384615385

5. Comparison

For calculating the effectiveness, we have compared proximity search engine model with the other models that we have implemented in our project. So, the Global measures for Proximity enabled search, like Mean Average Precision (MAP) and Mean Reciprocal Rank (MRR), has been evaluated.



MAP Comparison



MRR Comparison

When we compare the results of MAP and MRR we can easily see that the results of proximity search are much better. The value of MAP for Proximity search is much better than PRF. The map values proximity search is also near about to tf-idf but MRR values of proximity search is comparatively low as compared to tf-idf. We can also see that for non-stopped there is higher MAP value but there is lower MRR value. So, we can conclude that stopping results in favor of MRR but not in favor of MAP. This can be because of this that the relevant documents are appearing in top ranked documents but all the relevant documents for that query are not appearing in the result set. When we compare the proximity model with all the other models we may find that the Lucene Model and BM25 model are still good significantly. Thus, the working of our proximity model is solely based on the algorithm we have used and the corpus.