Comparison of Information Retrieval Methods

1. Introduction

The aim of this work was to compare methods by using them retrieve a ranked list of documents from a collection corresponding to a given query. The smoothed tf-idf scores were compared with unsmoothed tf-idf scores and unsmoothed BM25 scores. The parameters of all language models were varied and language models were compared with each other.

2. Related work

BM25 is one of the most popular methods used for document retrieval (Fan, Xi, Fox, & Wang, 2004). Therefore it is interesting to compare the method against the Tf-idf method. Previous work has found that for language models, retreival performance is sensitive to the smoothing parameters.

The performance is generally more sensitive to smoothing for verbose queries than for keyword queries.The Jelinek-Mercer tends to perform much better for verbose queries than for keyword queries. As in the given assignment queries are mostly keywords, the Dirichlet and Absolute discounting methods should perform better than the Jelinek Mercer method.

For Jelinek, optimal λ value is generally very small for short queries (around 0.1). The optimal value of μ appears to have a wide range (500-10000) and usually is around 2,000. For absolute discounting, there is little variation in the optimal value for δ and is generally around 0.7 in all cases.

3. Research Questions

1. Is the unsmoothed BM25 a more optimal way of ranking documents compared with the smoothed and unsmoothed tf-idf method?
    a. In which cases is either query performing well?
2. Without using smoothing, what are the optimal model parameters for
    a. Jelinek-Mercer language model
    b. Dirichlet prior learning language model
    c. Absolute discounting
3. Which language model gives the most optimal ranking of documents without smoothing?
    a. Is the language model performance better than BM25 or Tf-idf?

4. Methodology

Initially the queries were processed without removing special characters. Due to this fact, queries 167 and 180 initially returned no matching documents. After the special characters were removed, both of the queries also returned matching documents.

A list was constructed of all of the unique query terms and for each query term a corresponding dictionary was made which contained document numbers for all documents that had the query term in the document and a tf-score. This dictionary was used to calculate evaluation scores per query term for the following methods: Tf-idf, BM25, Jelinek-Mercer language model, Dirichlet prior language model and Absolute Discounting.

The following formulas were used to calculate the scores for each method:

Tf-idf non-smoothed: $tf * \left(\frac{N}{df}\right)$

Tf-idf smoothed: $\log(1 + tf) * \log\left(\frac{N}{df}\right)$

BM25: $\sum IDF(q) * \left[\frac{tf*k1+1}{tf+k1*(1-b+b*\frac{|C|}{avgdl}}\right]$

Jelinek-Mercer: $P(w|d) = \lambda \frac{\text{tf}}{|d|}(1-\lambda)\frac{\text{tf(w;C)}}{|C|}$

Dirichlet $\text{P(w|d)} = \frac{|\text{d}|}{|\text{d}|+\mu}\frac{\text{tf}}{|\text{d}|} + \frac{\mu}{|\text{d}|+\mu}\frac{\text{tf(w;C)}}{|\text{C}|}$

Absolute discounting $P(w|d) = \frac{\max(\text{tf}-\delta)-\delta,0)}{|d|} + P(w|C)$

For BM25, k1 = 1.5 and b = 0.75 were used as the values for hyperparameters since these are the most commonly used values. For Jelinek-Mercer the following λ values were tested: 0.05, 0.1, 0.2, 0.5, 0.9. For Dirichlet, the following μ values were tested: 500, 1000, 1500 and 2000. For absolute discounting, the following δ values were tested: 0.1, 0.5 and 0.9.

After the scores were calculated for every query term, the query term scores were added together. A dictionary was created that had a query number as key and a dictionary that contained the document numbers and the corresponding scores as values.

Initially instead of a dictionary, an array was used and the values corresponding to all documents were stored in the array for every query. This approach proved to need too much computational resource and thus, only documents that hada a score higher than 0 were stored in a dictionary with the corresponding queries.

No smoothing was used for the language models because of technical difficulties. The function used to add the query term scores together into a query score, did not accept negative values. After smoothing, all of the query term values were negative. However, after this problem was discovered, there was not enough time to deliver smoothed results.

Top 1000 results were recorded for all queries, except for queries 57, 78 and 180 as less than 1000 documents returned a score above 0.

A report was constructed and the query scores were evaluated using trec eval. Both precision, recall and NDCG were recorded at 10 because the top 10 results are considered to be the most relevant for user search engines. Recall was also recorded at 1000 because this shows what percentage of all relevant results were ranked. The MAP was recorded for all of the results.

5. Results and Analysis

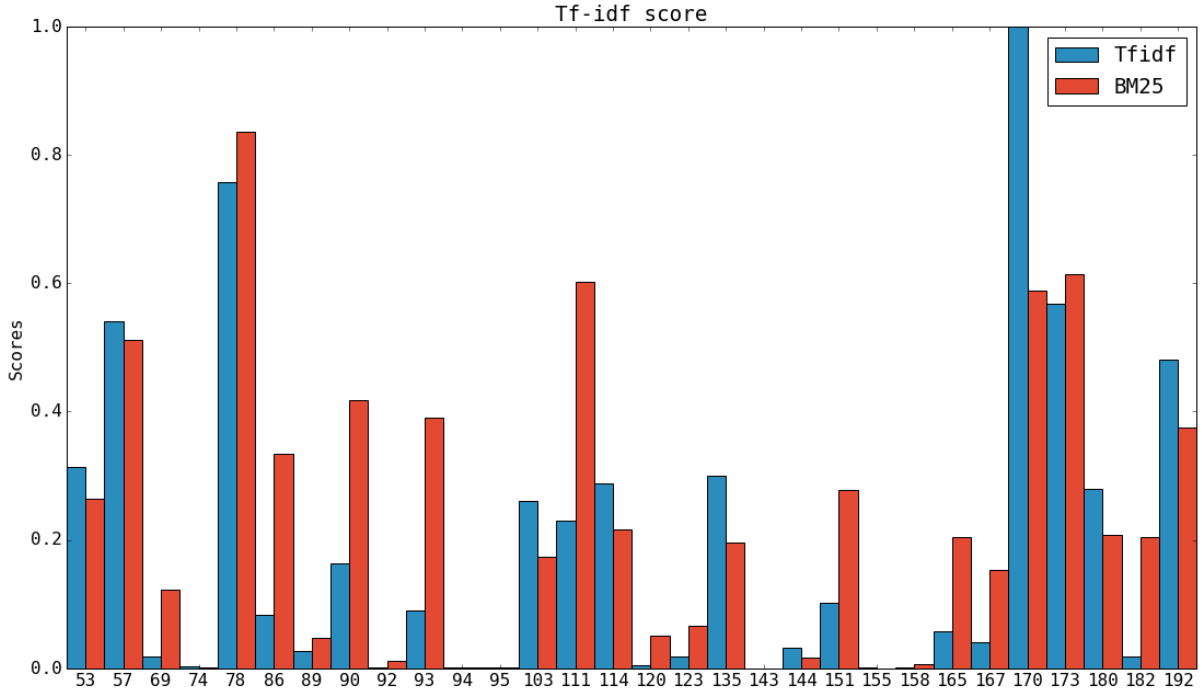
1. Is the unsmoothed BM25 a more optimal way of ranking documents compared with the smoothed and unsmoothed tf-idf method?

Method	P@10	R@10	R@1000	NDCG@10	MAP
Tf-idf: not smoothed	0.2433	0.0629	0.5939	0.2513	0.1896
Tf-idf: smoothed	0.3367	0.0837	0.6460	0.3689	0.2420
BM25: not smoothed	0.3300	0.0865	0.6447	0.3635	0.2291

Table 1. Comparing smoothed tf-idf scores with

As can be seen from Table 1, when the smoothed version of tf-idf was used, the scores across all evaluation measures were higher compared with the non-smoothed version. The unsmoothed BM25 formula gives evaluation scores that are comparable with the smoothed tf-idf scores but the smoothed tf-idf model performs the best out of all three methods.

In which cases is either method performing well?



Graph 1. Comparison of MAP across queries for Tfidf(unsmoothed) and BM25(unsmoothed)

As can be seen from Graph 1, there were considerable differences in how the unsmoothed Tf-idf and BM25 perform across all of the queries. The the BM25 method gave higher results for example in queries 111 (“Non-commercial satellite launches”) where the MAP was 37% higher compared with tf-idf and for query 93 (“What backing does the national assault rifle have”) where the MAP was 30% higher compared with the tf-idf score. This is likely due to the fact that the BM25 takes the document length into account. For query 170 - “The Consequences of Implantation of Silicone Gel Breast Devices” the MAP was 41% higher for the tf-idf method compared with the BM25 method.

Both of the methods achieved a high MAP for query 78 (“Greenpeace”) and had a low MAP for queries like 74(“Conflicting Policy”), 94 (“Computer-aided crime”) and 155 (“Right Wing Christian Fundamentalism in U.S.”). For queries 94 and 95, no results were found because “computer-aided” was not assigned a word id. For query 155, the “U.S” was not assigned a word id, this might have been a possible reason why the returned documents were not considered relevant.

The number of times a query appears in a document is not the only indicator of relevance. For example, for query 53 („Leveraged Buyouts“), document AP890202-0297 received a high tf-idf score and the document had a 4.2 % overlap with the query but it was likely marked as irrelevant. Document AP880928-0249 was marked relevant but was not selected among the top 1000 documents based on the tf-idf score. When examining the document, it can be seen that even though only one query word appears once („buyout“), there are several words in the document that are related with the same topic. For example, “share”, “economic”, “bidding”, “shareholders”, “companies”.

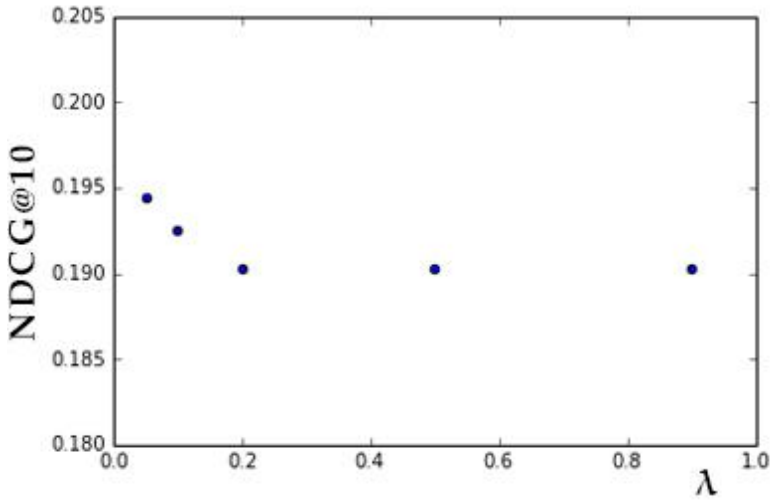
2. What are the optimal model parameters for the language models

a) Jelinek-Mercer language model

	P@10	R@10	R@1000	NDCG@10	MAP
$\lambda = 0.9$	0.1733	0.0550	0.4678	0.1903	0.1282
$\lambda = 0.5$	0.1733	0.0550	0.4681	0.1903	0.1283
$\lambda = 0.2$	0.1733	0.0550	0.4681	0.1903	0.1283
$\lambda = 0.1$	0.1767	0.0552	0.4681	0.1925	0.1283
$\lambda = 0.05$	0.1767	0.0552	0.4660	0.1944	0.1282

Table 2. Comparing the evaluation scores between varying levels of λ

As can be seen from Table 2, the λ value that obtained the highest NDCG score for the unsmoothed Jelinek model was 0.05. As slightly higher results were obtained when the λ value was small, it indicates that emphasizing the importance of the collection probability gave higher evaluation scores. As can be seen from Graph 2, the NDCG scores for λ at values 0.2, 0.5 and 0.9 were similar.



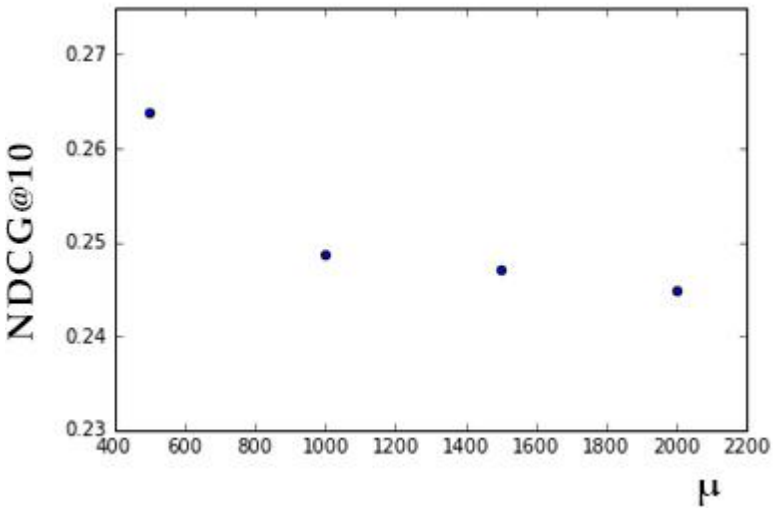
Graph 2. Comparing the evaluation scores between varying levels of λ.

b) Dirichlet prior smoothing

	P@10	R@10	R@1000	NDCG@10	MAP
μ = 2000	0.2400	0.0636	0.5581	0.2448	0.1826
μ = 1500	0.2433	0.0650	0.5554	0.2471	0.1832
μ = 1000	0.2400	0.0659	0.5543	0.2487	0.1843
μ = 500	0.2500	0.0684	0.5433	0.2638	0.1828

Table 3. Comparing the evaluation scores between varying levels of μ.

As can be seen from Table 3, the μ value that obtained the highest NDCG score for the unsmoothed model was 500. As slightly higher results were obtained when the μ value was small, it indicates that emphasizing weighting of terms gave better results. As can be seen from Graph 3, the NDCG scores for μ values 1000,1400 and 2000 were similar.



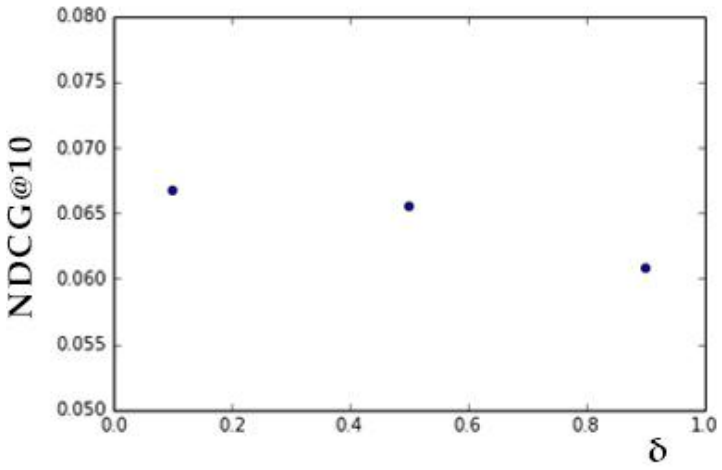
Graph 3. Comparing the evaluation scores between varying levels of μ.

c) Absolute discounting smoothing

	P@10	R@10	R@1000	NDCG@10	MAP
δ = 0.1	0.0833	0.0076	0.3500	0.0855	0.0668
δ = 0.5	0.0833	0.0076	0.3529	0.0811	0.0656
δ = 0.9	0.0833	0.0076	0.3441	0.0792	0.0609

Table 4. Comparing the evaluation scores between varying levels of δ

As can be seen from Table 4, the δ value that obtained the highest NDCG score for the unsmoothed model was 0.1. As slightly higher results were obtained when the δ value was small, it indicates that not emphasizing the collection probability gave higher evaluation scores. As can be seen from Graph 4, the NDCG scores for δ values 0.1, 0.5 and 0.9 were very similar.



Graph 4. Comparing the evaluation scores between varying levels of δ.

3. Which language model gives the most optimal ranking of documents without smoothing?

When the language models were not smoothed, the Dirichlet method gave the highest NDCG@10 score: 0.2638 (Table 3). The Jelinek-Mercer method achieved a a score of 0.1944 (Table 2) and the absolute discounted smoothing achieved a score of 0.0855 (Table 3).

a) Is the language model performance better than BM25 or Tf-idf?

As can be seen in Table 1 and in Table 3, the smoothed Tf-idf method achieved higher evaluation scores across all evaluation metrics. This is likely due to the fact that the Dirichlet scores were not smoothed.

6. Conclusion

It was found that a smoothed version of Tf-idf achieved the highest precision, recall and NDCG at 10 results. In addition, it reached the highest MAP score and the highest score for recall at 1000. The unsmoothed BM25 method reached higher scores than the unsmoothed Tf-idf method but lower than the smoothed Tf-idf method.

From the unsmoothed language models, the Dirichlet model achieved the highest evaluation metrics. These results were lower than the results gathered with the unsmoothed Tf-idf, smoothed Tf-idf and unsmoothed BM25 methods.

References

Fan, W., Xi, W., Fox, E. A., & Wang, L. (2004). Can We Get A Better Retrieval Function From Machine? In *TREC*. Citeseer. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.80.6430&rep=rep1&type=pdf

Zhai, C., & Lafferty, J. (2004). A study of smoothing methods for language models applied to information retrieval. ACM Transactions on Information Systems (TOIS), 22(2), 179-214.