Building IR systems based on Indrir Report

Pan Yueh-Sheng

May 18, 2021

Abstract

The project is based on Indri as search engine for information retrieval to implement several different retrieval methods including TF-IDF, BM25, Language Models with different Smoothing to assign a score to each document according to its relevance to the query. This report will focus on the difference in Precision of the four models under different Recall rates.

1 Introduction

1.1 Document Corpus

The document corpus uses WT2g data collection to constructs two indexes: with stemming (Porter stemmer), and without stemming. Both indexes contain stopwords.

The document corpus have 247,491 document and 261,742,791 terms. After indexing, the index with stemming includes 1,391,908 unique terms, and the index without stemming includes 1,526,004 unique terms.

1.2 Queries

The query is a set of 50 TREC queries for the corpus, with the standard TREC format having topic title, description and narrative.

1.3 Ranking Functions

There are four different ranking functions to run the set of queries against the WT2g collection, return a the top 1000 ranked list of documents. The four retrieval systems are listed below:

1. Vector space model with BM25 weighting
2. Language modeling with Laplace smoothing
3. Language modeling with Jelinek-Mercer smoothing
4. Vector space model with BM25 weighting and pseudo relevance feedback

1.4 Evaluation

Run all 50 queries and return at top 1,000 documents for each query. The evaluation tool apply the script to see how well the ranking is.

2 Model description

In IndriRunQuery instruction, the parameter file can be set to different retrieval functions. The four functions are discussed below, each model is applied to both stemmed and unstemmed indexes.

2.1 Vector space model with BM25 weighting

Vector space model, terms weighted by Okapi TF (see note) times an IDF value, and inner product similarity between vectors. Set k1 = 2 and b = 0.75.

$$socre(D,Q) = \sum_{n=1}^{n} IDF(q_i) \cdot \frac{f(q_i,D)\cdot(k_1+1)}{f(q_i,D)+k_1\cdot(1-b+b\cdot\frac{|D|}{avgd})}$$

2.2 Language modeling with Laplace smoothing

Language modeling, maximum likelihood estimates with Laplace smoothing. Laplace smoothing (also known as Add-one smoothing) solved the previous problem what MLE encountered if a word was never seen in the training data, then the probability of that sentence is zero:

$$P(w|D) = \frac{count(w,D)+1}{len(D)+|V|}$$

2.3 Language modeling with Jelinek-Mercer smoothing

Language modeling, Jelinek-Mercer smoothing using linear interpolation of the maximum likelihood model with the collection model, set 0.8 of the weight attached to the background probability, which means $\lambda = 0.2$, query likelihood.

$$P_{JM}(w) = \lambda \cdot p(q|M_d) + (1-\lambda)\cdot p(q|M_c)$$

2.4 Vector space model with BM25 weighting and pseudo relevance feedback

Relevance feedback takes the results that are initially returned from a given query in order to gather user feedback, and use information about whether or not those results are relevant to perform a new query.

The steps are listed below:

1. Use vector space model with BM25 weighting firstly
2. Take the terms of the first N documents as potential expansion terms
3. Calculate the score for each potential expansion term
4. Create a new Qlearned with the first m terms
5. Retrieve documents with new query

The model specifies ten documents and ten terms to use for pseudo relevance feedback.

3 Model Evaluation

In this section, Each model uses query parameters in 50 TREC quires as search keywords, and includes stemmed and unstemmed indexing. After generating the results, use qrel file for the WT2g corpus to compare the results to see if the results are correct.

3.1 Model Comparison (R-Precision, Confusion Matrix)

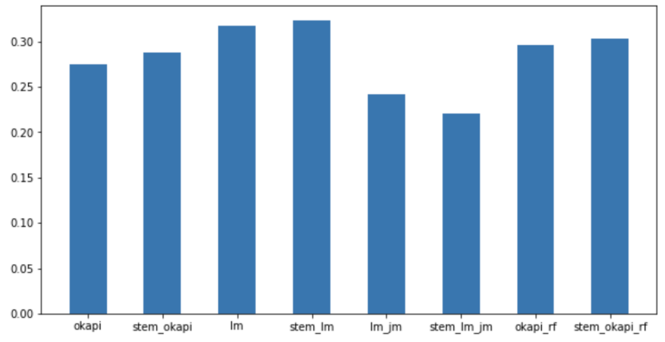

Figure 1: R-Precision in 8 Models

Model	Precision	Recall	F1 Score
Okapi	0.029	0.619	0.056
Stemmed Okapi	0.034	0.723	0.064
Language Model	0.034	0.714	0.064
Stemmed Language Model	**0.039**	**0.828**	**0.074**
Language Model JM	0.03	0.629	0.057
Stemmed Language Model JM	0.032	0.695	0.062
Okapi RF	0.034	0.756	0.066
Stemmed Okapi RF	0.036	0.781	0.068

Table 1: Precision, Recall, F1 Score in 8 Results

The Figure 1 and Table 1 shows that Stemmed Language Model with Laplace smoothing surpassed other models in R-Precison, Precision, Recall, F1 score, and Stemmed Okapi Relevance Feedback ranked second in all three categories of confusion matrix.

3.2 Average precision (non-interpolated)

The Average precision with non-interpolated is the average rate when the number of documents is 5, 10, 15, 20, 30, 100, 200, 500, 1000 respectively.

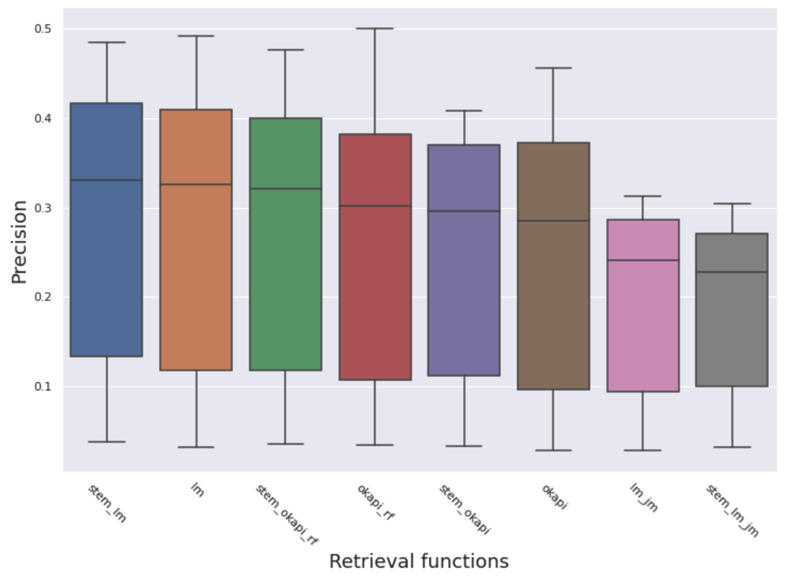

Figure 2: The precision of models under different number of document.

3.3 Average precision (11-point interpolated)

The average precision is measured for the 11 recall values 0.0 to 1.0

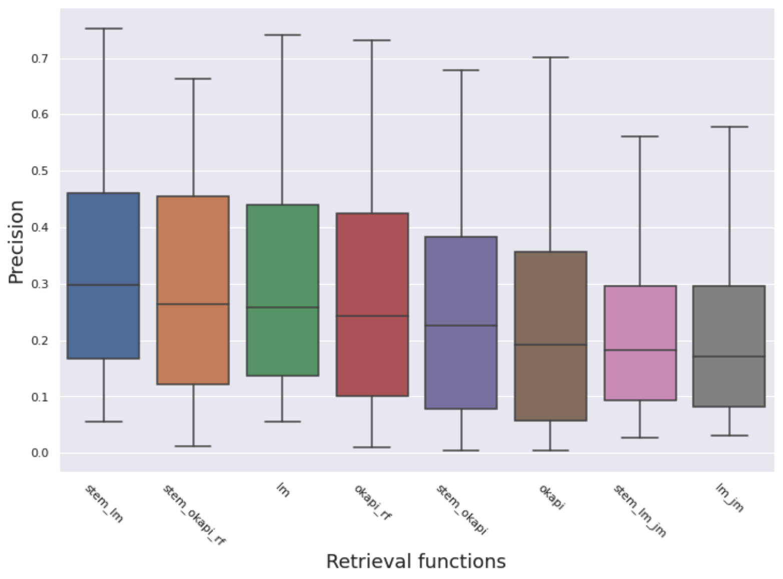

Figure 3: The precision (11-point interpolated) of different models under different number of document.

The Figure 2 and Figure 3 show that if the index has been stemming, the median results are generally better than without stemming, but the type of model is still the main factor that affects the results. In addition, the larger the full range also means that the accuracy of the front document is significantly improved. Compared to the unstemmed index, the stemmed index is slightly inferior to the unstemmed index in both the highest and lowest values. Furthermore, the Stemmed Language Model has shown good results, while the Language Model and Stemmed Okapi Relevance Feedback have their own advantages and disadvantages in Interpolated Recall-Precision Averages. The former has a higher Precision peak, while the latter is in the median, better than the former.

So I want to know: which of these two models is better? The following will compare the two:

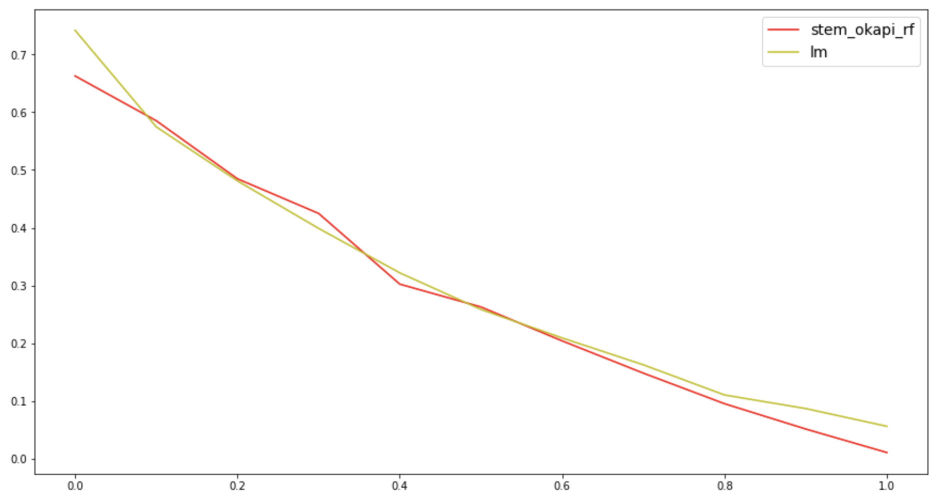

Figure 4: The precision (11-point interpolated) in Language Model and Stemmed Okapi Relevance Feedback

It can be seen from the Figure 4 that the Language Model with Laplace smoothing is better than Stemmed Okapi Relevance Feedback in most situations, so the Language Model can perform well whether it is more than a large number of related documents search or users who need to obtain some related documents more quickly.

Is it possible to improve the performance of Okapi model by changing the values of k and b? The general value range of k1,b is k1∈[1.2,2.0], b=0.75. Set the value of k1 to 1.2 and compare it with the Language Model and the results are as follows:

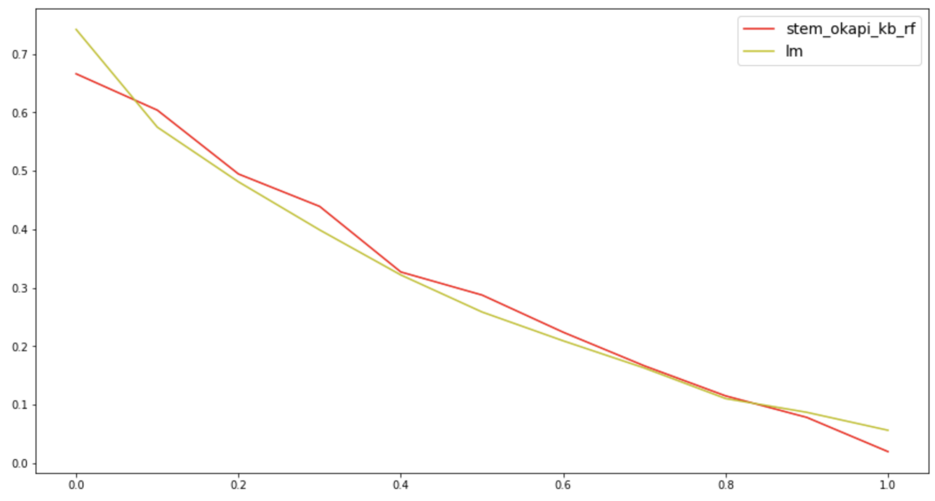

Figure 5: The precision (11-point interpolated) in Language Model and Stemmed Okapi Relevance Feedback with k1 = 1.2

From the results in the Figure 5, it can be found that when k1 is set to 1.2, the model effect has been improved, which means that the adjustment factor for term frequency can improve the effect in this case, but when the recall value is close to 0 and 1, Language Model with Laplace smoothing still has its advantage.

If the Language Model is added with Relevance Feedback, can there be more significant effects?

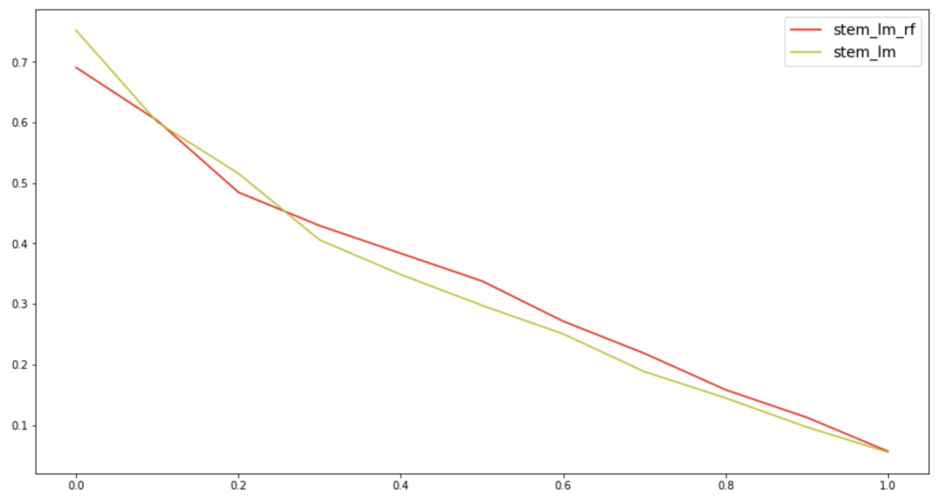

Figure 6: The precision (11-point interpolated) in Stemmed Language Model and Stemmed Language Model Relevance Feedback

The R-Precision value of the Stemmed Language Model of Relevance Feedback is 0.3520, while the value without Relevance Feedback is 0.3235. From the 11-point interpolated graph of Figure 6, the highest precision value of the Language Model without Relevance Feedback is still better than the Relevance Feedback model. These results show that although the Relevance Feedback model can increase the Precision value, it will decrease its Precision when the recall is around 0. Therefore, the Relevance Feedback model is more suitable for reading a large number of related documents.

3.4 Discussion on Language Model Smoothing

In the previous boxplot results, the result of using Language Model with Jelinek-Mercer smoothing is much lower than Language Model with Laplace smoothing, regardless of the highest value, lowest value and median. But can it be inferred that the smoothing method of Laplace smoothing performs better? The Precision change of the original model is as follows:

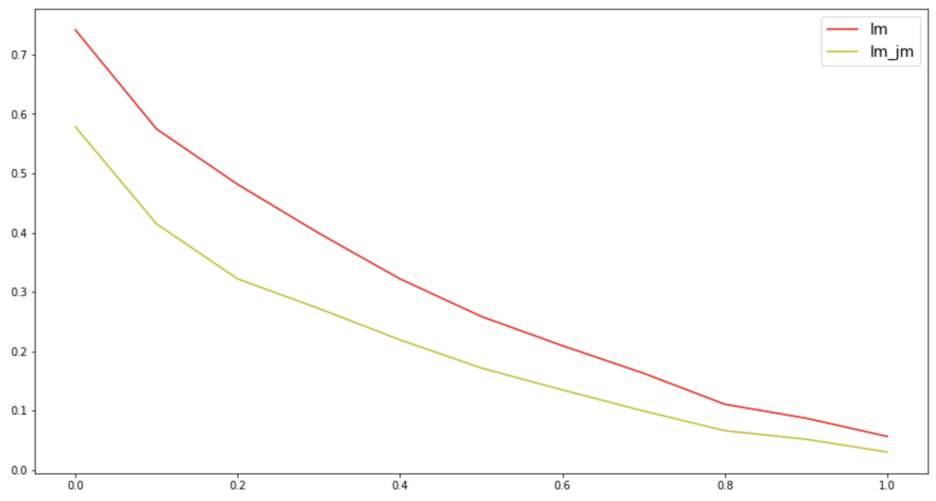

Figure 7: The precision (11-point interpolated) in Language Model with Laplace smoothing and Language Model with Jelinek-Mercer smoothing

I tried to change the λ value of Jelinek-Mercer smoothing, from 0.1 to 0.9 every 0.1 interval. The change in R-precision is as follows:

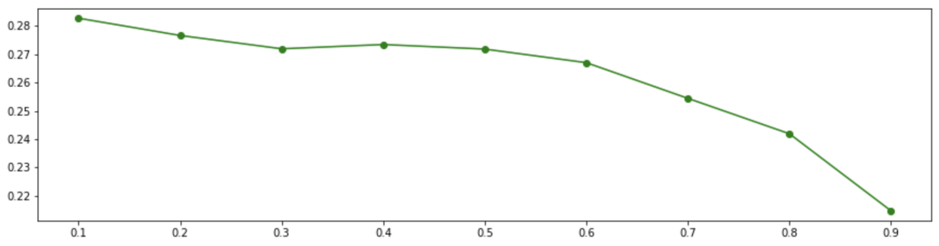

Figure 8: The change in R-precision with Jelinek-Mercer smoothing

According to the Figure 8, R-precision has a better performance when λ is 0.1, but the increase is not much, the gap between the largest and smallest is 0.06. The following Figure compares the λ value of Language Model with Jelinek-Mercer smoothing to 0.1 with Language Model with Laplace smoothing:

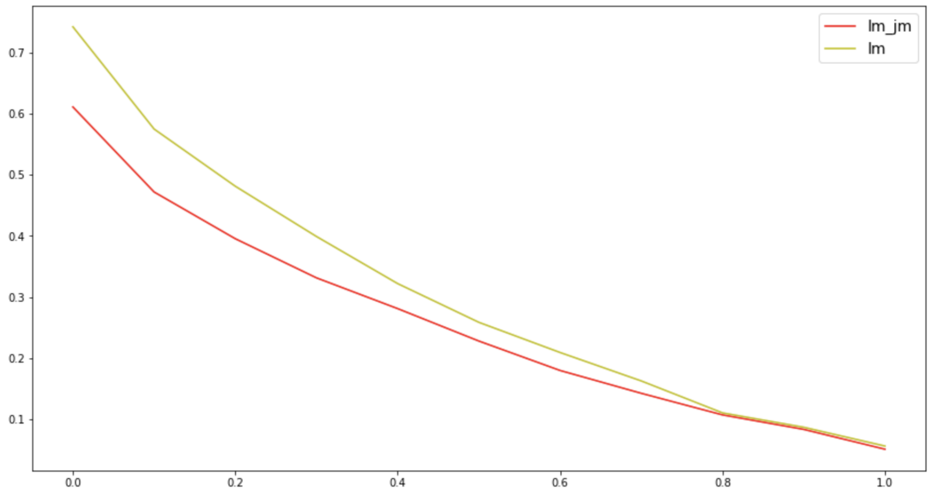

Figure 9: The change in R-precision with Jelinek-Mercer smoothing

It can be inferred that Laplace smoothing is indeed more suitable in this set of data.

4 Conclusion

From the above experimental results, it is found that in the WT2g document collection, Language Models performs better than Probabilistic Models for Information Retrieval on topic finding tasks. Both stemmed index and Relevance Feedback can help the model to improve accuracy to varying degrees.

In this project, I learned how to build an open source package and understand its power: when searching for keywords in many corpus, it only takes a short time to get relevant results.

If you are interested in the code and original information, you can visit my Github repo, I will put trec eval and retrivel result here.