Web Searching and Mining Project

Professor : Ming Feng Tsai
Student : Shu Hong Liu

May 21, 2019

Contents

List of Figures

1 Introduction

I used WT2G datasets which is part of WT10G from **TREC Web Corpus**. Since 1992, a series of annual benchmarking evaluation exercises , called **TREC (Text REtrieval Conference)**, have launched in the USA. TREC experiments were designed to allow large-scale laboratory testing, compare the effectiveness and performance of different information retrieval techniques. TREC has become the standard in the IR field. So, here I used WT2G that is a smaller datasets from TREC.

In WT2G, we can find that there are 247491 documents and about 1,500,000 unique words. That is a big challenge for me, because I studied in Statistic and the number of documents is too big for me to do something. Fortunately, professor Tsai recommend Indri toolkit to do information retrieval more easily than I thought before.

2 Method

Here I am going to introduce the methods that I used.

2.1 Vector Space Model

In **vector space model**, the most important thing is **tf**, **idf** and **the measurement of distance**.
So, I used **Okapi** TF times IDF value and inner product similarity to calculate the distance between two vectors.

$$OkapiTF = \frac{tf}{tf + k1 \times ((1-b) + b \times \frac{doclen}{avgdoclen})}$$

What does k1 and b mean? In this formula, we can see that when k1 is increasing, Okapi Tf will be decreasing. So, if we set k1 a bigger number, that means we care IDF more. After that, b is a parameter to control document length / average document length. Why we need to control document length / average document length? If the document length is too big, it may have a bigger term ferquency easily. So, we need to punish this situation. When b is increasing, the measurement of punishment also is increasing, so Okapi TF will be decreasing.

At first, I set

$$k1 = 2$$

and

$$b = 0.75$$

to see what is going on. The tuning parameter is important that will change result a lot, I want to do many trials to find the best tuning parameters in this project.

2.2 Language model with Laplace smoothing

In language model, we have many smoothing methods. I used **Laplace smoothing** here, and **Jelinek-Mercer smoothing** will be the next method. In this method, I used maximum likelihood estimates with Laplace smoothing only, query likelihood.

I calculate each term's likelihood in each documents and multiply the term's likelihood which appears in query, and give each document a value. Because I assumed that the term's appearance is unigram, I used the multiplication. So, the document's value is bigger then it would be retrieved sooner.

The likelihood definitoin:

$$\rho_i = \frac{m_i + 1}{n + k}$$

where m = term frequency
n = number of terms in document (doc length)
k = number of unique terms in corpus.

2.3 Language model with Jelinek-Mercer smoothing

In **Jenlinek-Mercer smoothing**, we still use term frequency, but we need to consider a term appeared in one document and appeared in all documents. It is like local and global, we need to consider both situation. There will be a parameter to control which situation we care more. In this time, I take the parameter 0.8, that is to say I care the global situation more.

In this way, we don't be afraid of the query term didn't appear in some documents that get a term frequency equal to 0 to cause this document's value equal to 0.

$$\rho_i = \lambda \times P + (1 - \lambda) \times Q$$

where P is the estimated probability from document (max likelihood =

$$\frac{m_i}{n}$$

)
and Q is the estimated probability from corpus (background probability = cf/terms in the corpus)

3 Trials

Before using methods that I mentioned above, I construct two indexes, (1) **without stemming**, and (2) **with stemming**. In each index, I used all of methods above, so there will be 6 runs.

3.1 WT2G without stemming

In this section, I constructed index without stemming, that is to say I barely did nothing. The reason I did nothing is that I wanted to know the baseline, just depend on the methods, how far that we can go. Let's see the results. I will show (1) Precision-Recall plot and (2) Precision at docs plot to display how well does our model perform.

3.1.1 Vector Spcae Model

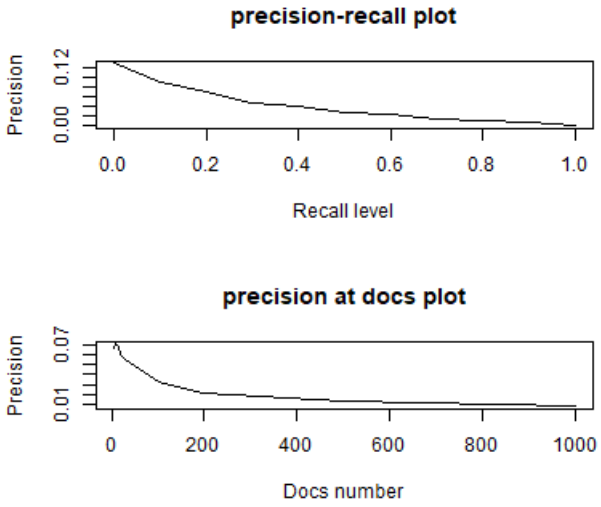
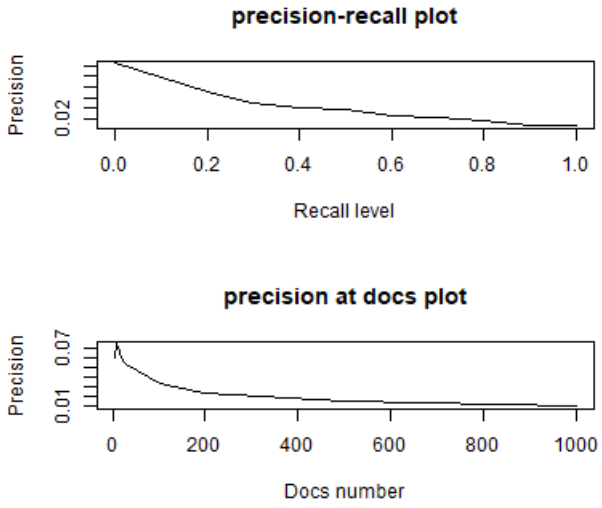


Figure 1: Vector Space Model without stemming

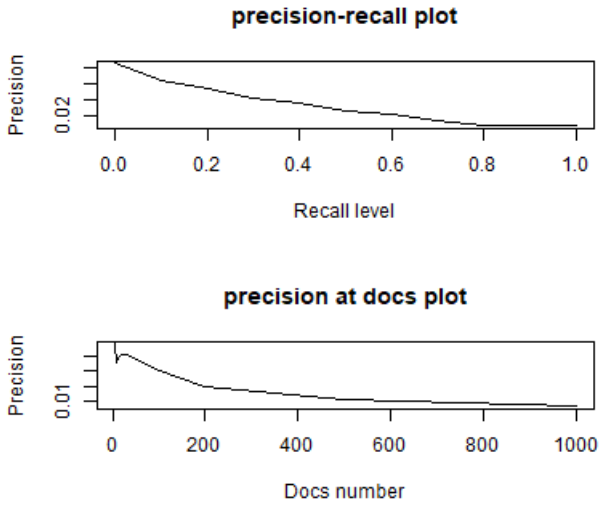
3.1.2 Language model with Laplace smoothing

Figure 2: Language model with Laplace smoothing without stemming



3.1.3 Language model with Jelinek-Mercer smoothing

Figure 3: language model with Jelinek-Mercer smoothing without stemming



3.1.4 Compare

In figure1, figure2 and figure3, we can see that the overall precision is not high enough. However, three methods still have different points. Let's compare them to see which one is better.

Figure 4: Three methods Precision - Recall plot without stemming

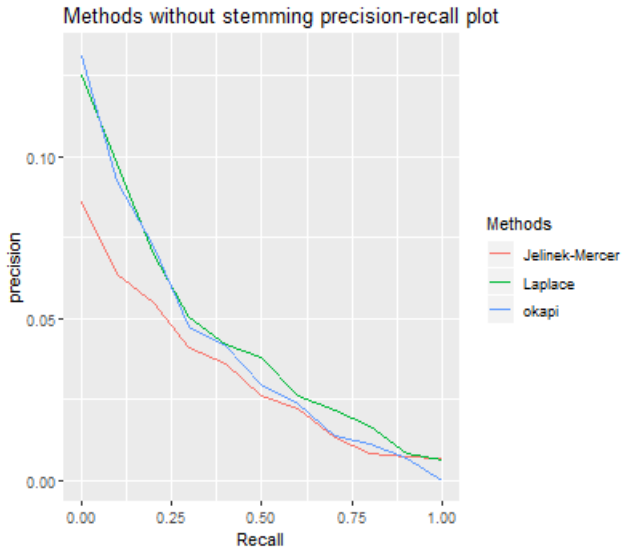
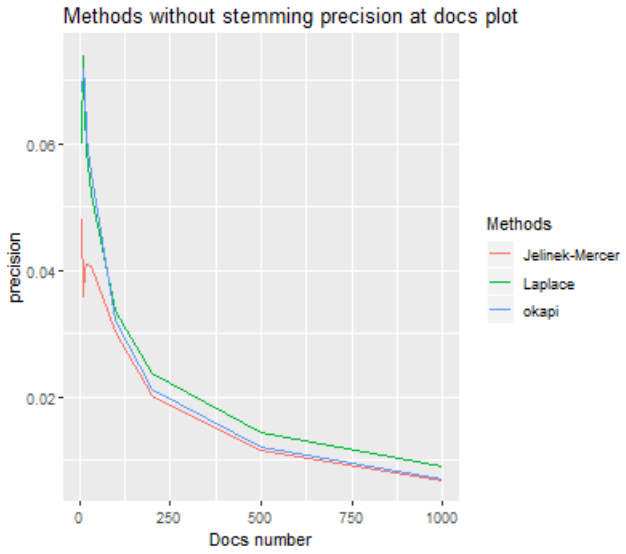


Figure 5: Three methods Precision at docs plot without stemming

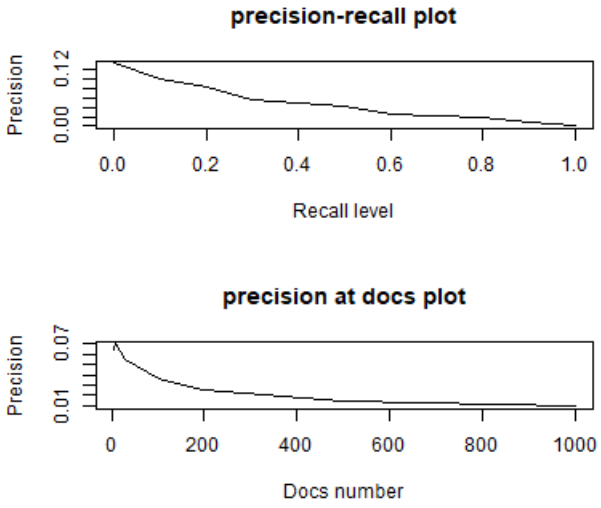


By figure4 and figure5, we can see that Jelinek-Mercer smoothing method is the worse method in this trial. And, we can separate the results of Vector space model and Laplace smoothing, because the two methods are very similar. Both of them consider term frequency and do some smoothing, but the Jelinek-Mercer smoothing also consider the global term frequency, maybe that's the reason it can't perform well.

3.2 WT2G with stemming

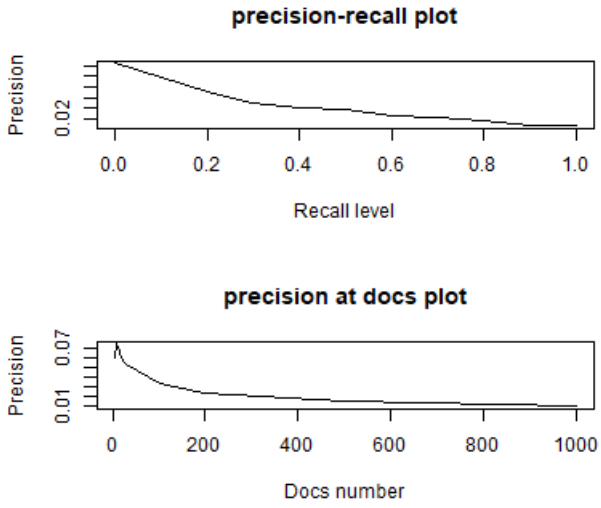
3.2.1 Vector Spcae model

Figure 6: Vector Space Model with stemming



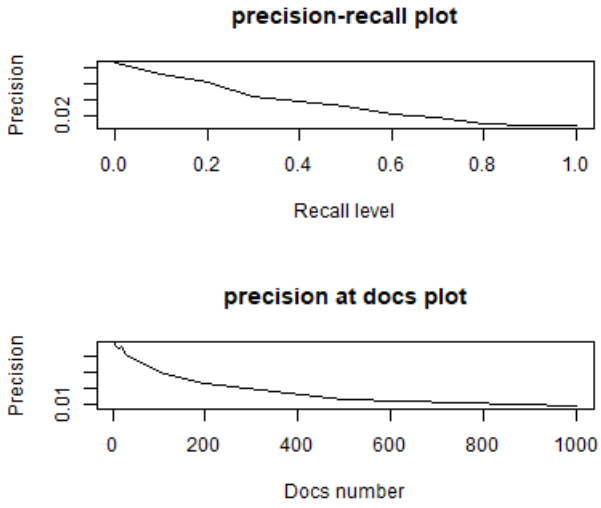
3.2.2 Language model with Laplace smoothing

Figure 7: Language model with Laplace smoothing with stemming



3.2.3 Language model with Jelinek-Mercer smoothing

Figure 8: Language model with Jelinek-Mercer smoothing with stemming



3.2.4 Compare

Three methods with stemming results are similar to methods without stemming, so let's compare them again.

Figure 9: Three methods Precision-Recall plot with stemming

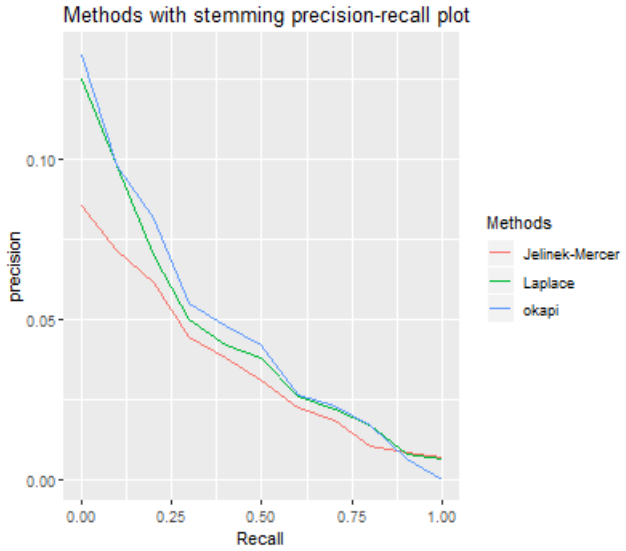
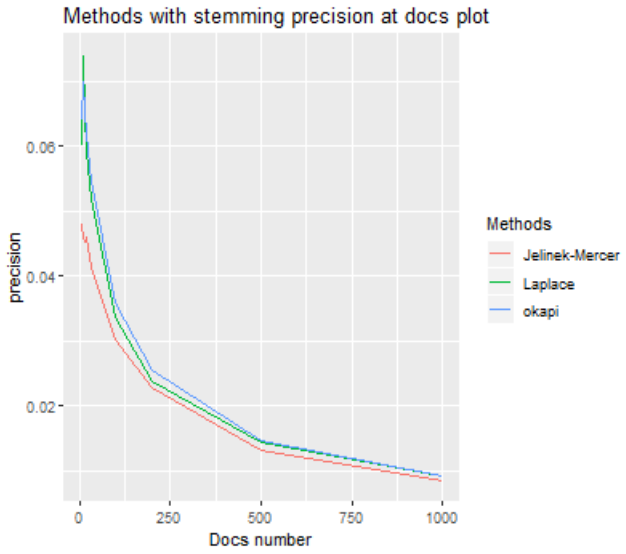


Figure 10: Three methods Precision at docs plot with stemming



By figure 9 and figure 10, we can get the same result like methods without stemmming. The Jelinek-Mercer smoothing method is still the worse way to smooth the term frequency.

4 Idea

In these trials , We can see that the results are bad enough. However, that means there's still room for improvement. I think that we can clean text by stemming, deleting stopwords, numbers and punctuation to get more accurate documents. Moreover, we can not only use queries' title from WT2G but also add nouns from queries' description and narrative which are used to describe more precisely about queries. Also, we can input qeury by query to find that in which query the method will work fine or in which query the method will work poor.

5 Conclusion

In this project, I learned the toolkit Indri and some methods for information retrieval. In my opinion, I think that all of these methods are similar, and only the smoothing ways are a little different. I want to know the ultimate solution to solve language.**How did language start? Where languages come from? Why was language created?**A language's appearance was a singular point, but why and how? I want to find these answers.