WSM Project2

109208064

December 6, 2022

Abstract

This paper represent several different retrieval methods, i.e. algorithms that given a user's request (query) and a corpus of documents assign a score to each document according to its relevance to the query. Some of these retrieval methods are the implementation of the basic retrieval models (e.g. TF-IDF, BM25, Language Models with different Smoothing). In this case, I use the toolkits of Lemur Project, which includes search engines, browser toolbars, text analysis tools, and data resources that support research and development of information retrieval and text mining.

1 Introduction

This paper represents the experiments results based on modified algorithms of Indri Project (four retrieval models), running on a set of 50 TREC queries against WT2G, which is a 2GB collection, that contains Web documents. Documents from the corpus have been judged with respect to their relevance to queries by NIST assessors. It then return a ranked list of the top 1,000 documents for each query, which have non-zero scores. and then evaluation of the ranked lists, which includes MAP, precision at various recall cut-offs, R-Precision. For the queries, they are in standard TREC format having topic description (used only in last experiment), or title(used in all other experiments). For the corpus (WT2G), I construct two indexes, with stemming (using porter stemmer), and without stemming. Both indexes contain stopwords.[Tsa]

2 TF-IDF Model

2.1 Raw Tfidf

TF-IDF (term frequency-inverse document frequency) is a statistical measure that evaluates how relevant a word is to a document in a collection of documents.[Sei]
The model assume that the more times a document contains a term, the more likely it is to be about that term. That's to say, we'll use term frequency (TF), the number of occurrences of a term in a document, as a proxy for relevance.
To prevent filler words from dominating, the model judge the importance of the terms in a query by calculating the rarity of the term (IDF). If a term doesn't occur in most documents in the corpus, then this occurrence is significant. But if a term occurs in most of the documents in our corpus, then the presence of that term in any particular document will lose its value as an indicator of relevance. [Ste]
Therefore, the model get the funcion : $TF - IDF = TF * IDF$

2.2 Improved TF-IDF Model model

The TF-IDF Model model fails to consider document length. Longer documents are given an unfair advantage over shorter ones because they have more space to include more occurrences of a term, even though they might not be more relevant to the term. So I improved the TF-IDF model by rewarding matches in short documents, while penalizing matches in long documents. We then get the following function. [Ste] $weight(t,d) = \frac{TF(t,d)*IDF(t)}{k*(documentlength/averagedocumentlength)}$ where k is parameter

2.3 Compare Raw TF-IDF and Improved TF-IDF Models

According below experiment data, Improved TF-IDF Model performs better than Raw TF-IDF Model.

- **Models With Stemming**
  In Figure1, the upper curve is Improved TF-IDF and the lower curve is Raw TF-IDF. Improved TF-IDF gets higher precision than Raw TF-IDF.
  In Figure2, the upper curve is Improved TF-IDF and the lower curve is Raw TF-IDF. Improved TF-IDF gets higher precision than Raw TF-IDF before retrieveing 1000 documents.
  According to the table, Improved TF-IDF have all higher total Precision, total recall, precision at the 10th document, and R-precision. Therefore, we can say that Improved TF-IDF Model performs better than the Raw TF-IDF Model.

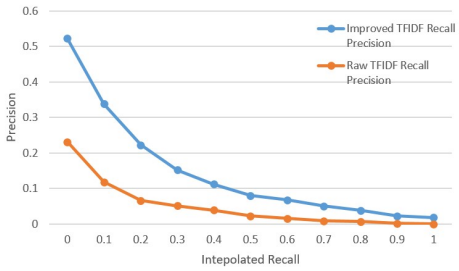


Figure 1: Recall-Precision

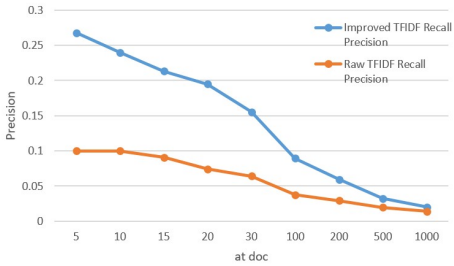


Figure 2: Uninterpolated precision

	Raw TF-IDF	Improved TF-IDF
Total Precision	0.3154	0.44318
Total Recall	0.0142	0.24
P@10	0.1	0.074
MAP	0.0386	0.1212
R-Precision	0.0628	0.1566

- **Models without Stemming**
  In Figure3, the upper curve is Improved TF-IDF and the lower curve is Raw TF-IDF. Improved TF-IDF gets higher precision than Raw TF-IDF when interpolated recall is smaller than 0.9.
  In Figure4, the upper curve is Improved TF-IDF and the lower curve is Raw TF-IDF. Improved TF-IDF gets higher precision than Raw TF-IDF before retrieveing 1000 documents.
  According to the table, Improved TF-IDF have all higher total Precision, total recall, precision at the 10th document, and R-precision. Therefore, we can say that Improved TF-IDF Model performs better than the Raw TF-IDF Model.

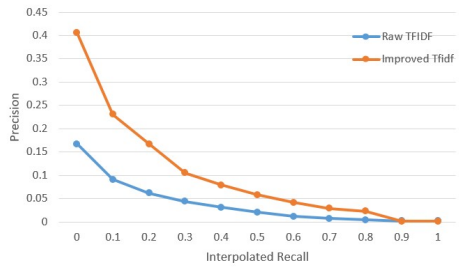


Figure 3: Recall-Precision

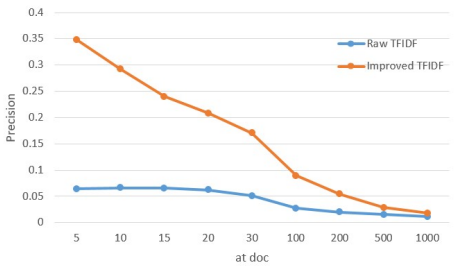


Figure 4: Recall-Precision

	Raw TF-IDF	Improved TF-IDF
Total Precision	0.24879	0.3967
Total Recall	0.01134	0.0181
P@10	0.066	0.0202
MAP	0.0316	0.0818
R-Precision	0.0496	0.1287

3 Okapi Model

3.1 Disadvantages Raw TF

1. **Overestimated Contribution of TF**
   In TF-IDF Model, once a document is saturated with occurrences of a term, more occurrences should not a have a significant impact on the score.
   For example, if a document contains 200 occurrences of "elephant," is it really twice as relevant as a document that contains 100 occurrences? The model argue that if "elephant" occurs a large enough number of times, say 100, the document is almost certainly relevant, and any further mentions don't really increase the likelihood of relevance.

2. **Longer documents have unfair advantage**
   Longer documents have more space to include more occurrences of a term, even though they might not be more relevant to the term. [Ste]

3.2 Okapi TF

1. **Control the Contribution of TF**
   Okapi control the contribution of TF to our score by counting Tf as $tf/(tf+k)$. The contribution increase fast when tf is small and then increase more slowly, approaching a limit, as tf gets very big.

2. **Penalize Long Documents**
   Okapi treat the importance of document length as a second parameter by adding the (1 – b + b*(dl/adl)) multiplier. This multiplier adjust k up if the document is longer than average, and adjust it down if the document is shorter than average.

3. **Result**
   TF = tf/(tf + k*(1 - b + b*(dl/adl))) [Ste]

3.3 Compare Okapi TF with Raw TF

Below experiments' parameters are set as k1 = 2 and b = 0.75

- **Models with Stemming**
  In Figure5, the upper curve is Okapi TF and the lower curve is Raw TF. Okapi gets higher precision than Raw TF.
  In Figure6, the upper curve is Okapi TF and the lower curve is Raw TF. Okapi gets higher precision than Raw TF before retrieveing 1000 documents.
  According to the table, Okapi TF also results in higher total precision, total recall, precision at the 10th doc, MAP, and R-Precision.

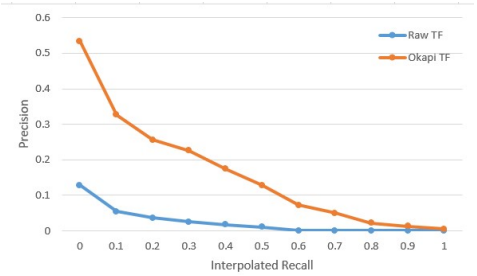


Figure 5: Recall-Precision

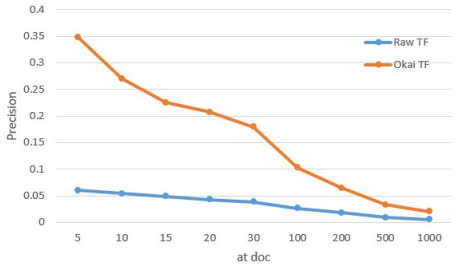


Figure 6: Uninterpolated precision

	Raw TF	Okapi TF
Total Precision	0.1237	0.4498
Total Recall	0.0056	0.0209
P@10	0.054	0.27
MAP	0.0167	0.1452
R-Precision	0.0369	0.1809

- **Models without Stemming**
  In Figure7, the upper curve is Okapi TF and the lower curve is Raw TF. Okapi gets higher precision than Raw TF.
  In Figure8, the upper curve is Okapi TF and the lower curve is Raw TF. Okapi gets higher precision than Raw TF before retrieveing 1000 documents.
  According to the table, Okapi TF also results in higher total precision, total recall, precision at the 10th doc, MAP, and R-Precision.

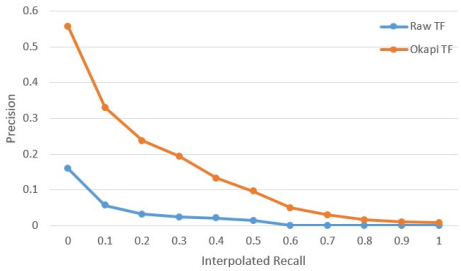


Figure 7: Recall-Precision

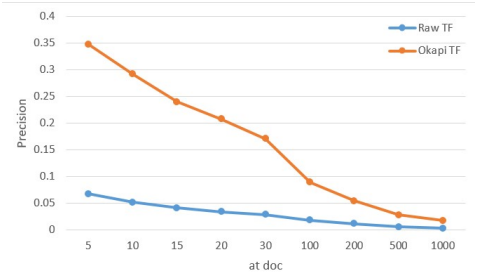


Figure 8: Uninterpolated precision

	Raw TF	Okapi TF
Total Precision	0.0715	0.3888
Total Recall	0.0033	0.01772
P@10	0.052	0.0292
MAP	0.019	0.1266
R-Precision	0.0367	0.1553

4 Language Modeling with Laplace Smoothing

4.1 LM with Laplace Smoothing

Language modeling (LM) is the use of various statistical and probabilistic techniques to determine the probability of a given sequence of words occurring in a sentence. [Lut]
Laplace Smoothing is a technique to smooth categorical data. Laplace Smoothing is introduced to solve the problem of zero probability. [Jay] Here we calculate the score by maximum likelihood estimates with Laplace smoothing only, query likelihood. And because we are using multinomial model, for every

document, only the probabilities associated with terms in the query must be estimated because the others are missing from the query-likelihood formula.[Tsa]
instead of : $P_i = \frac{m_i}{n}$
LM using Laplace Smoothing is estimated as : $P_i = \frac{m_i+1}{n+k}$
where m = term frequency, n=number of terms in document (doc length) , k=number of unique terms in corpus.

4.2 Laplace Smoothing Disadvantages

1. The probability of frequent n-grams is underestimated.
2. The probability of rare or unseen n-grams is overestimated
3. All the unseen n-grams are smoothed in the same way.
4. Too much probability mass is shifted towards unseen n-grams.[HM13]

4.3 Improvement of LM with Laplace Smoothing

One improvement to solve the disadvantage is to use smaller added count following the equation:
$score = \frac{m_i+\mu*collectionFrequency+1}{n+k+\mu}$

- **Models With Stemming**
  In Figure9, the upper curve is Original LM with Laplace Smoothing and the lower curve is Improved LM using Laplace Smoothing. Original model gets higher precision.
  In Figure10,the upper curve is Original LM with Laplace Smoothing and the lower curve is Improved LM using Laplace Smoothing. Original model gets higher precision.
  According to the table, the Original model have higher total precision, total recall, and precision at the 10th document. However, it is the improved model that get the higher MAP and R-Precision than the Original mode. But we can not say that Improved model really work when using stemming.

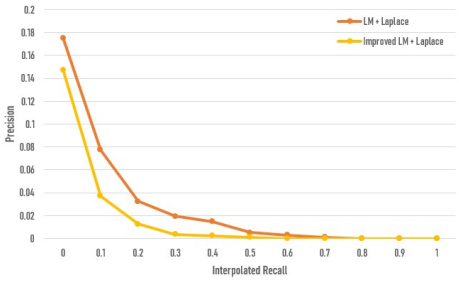


Figure 9: Recall-Precision

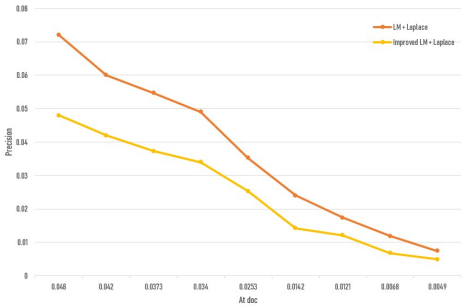


Figure 10: Lambda - Uninterpolated precision (with Stemming)

	LM + Laplace	Improved LM + Laplace
Total Precision	0.16191	0.1083
Total Recall	0.0074	0.0049
P@10	0.06	0.042
MAP	0.0203	0.096
R-Precision	0.0443	0.0248

- **Models Without Stemming**
  In Figure11,Improved LM using Laplace Smoothing gets slightly higher precision than Original

LM with Laplace Smoothing.
In Figure12, Improved LM using Laplace Smoothing also gets slightly higher precision than Original LM with Laplace Smoothing before retrieving 1000 documents.
According to the table, Improved model get the higher MAP, R-Precision, and precision at the 10th document. Original model only wins at total precision. Therefore, we can say that Improved model work in situation without stemming.

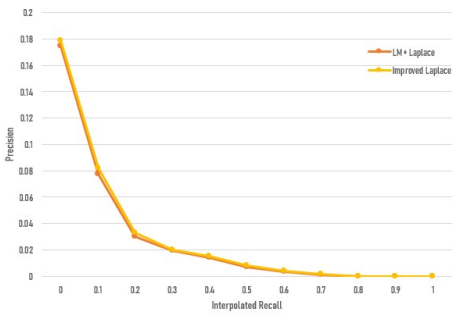


Figure 11: Recall - Precision

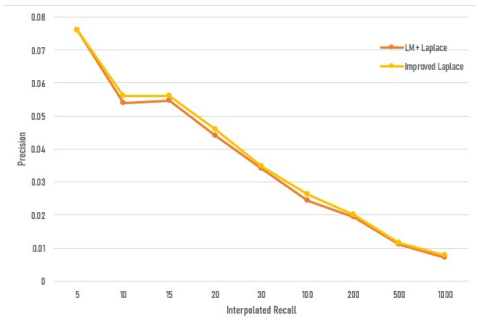


Figure 12: Uninterpolated Precision

	LM + Laplace	Improved LM + Laplace
Total Precision	0.1540	0.1684
Total Recall	0.007	0.0077
P@10	0.54	0.056
MAP	0.0202	0.0213
R-Precision	0.0386	0.0415

5 Language Modeling with Jelinek-Mercer Smoothing

5.1 LM with Jelinek-Mercer Smoothing

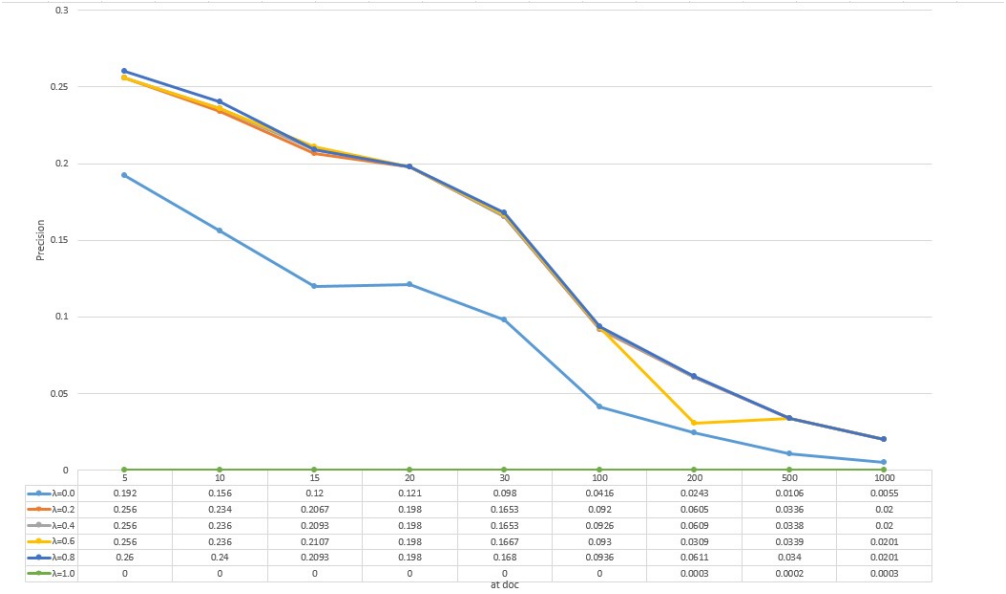
Laplace Smoothing is a simple linear interpolation method for combining the information from lower-order n-gram in estimating higher-order probabilities. The equation of linear interpolation is given below: $P_i = \lambda P + (1 - \lambda)Q$
Where λ is the confidence weight for the longer ngram. In general, λ is learned from a held-out corpus. It is useful to interpolate higher-order ngram models with lower-order n-gram models, because when there is insufficient data to estimate. P is the estimated probability from document (max likelihood $= m_i/n$) and Q is the estimated probability from corpus (background probability = cf / terms in the corpus).
Here we use 0.8 of the weight attached to the background probability, query likelihood.

5.2 Effect of Lambda

The experiment data below is tested at $\lambda = 0, 0.2, 0.4, 0.6, 0.8, 1.0$

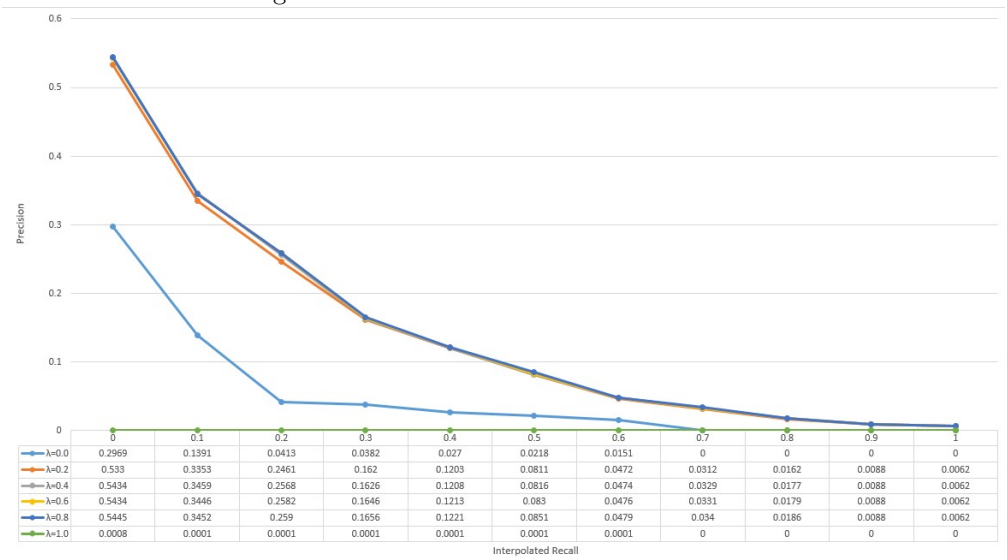
- **Models with Stemming**
  In Figure13 and Figure14, the closer λ increase approach to 0.8, the higher interpolated recall precision and uninterpolated precision it get. And the highest record of MAP is 0.1186, and the highest R-Precision is 0.1679, where $\lambda = 0.8$. However, the precision drop dramatically when $\lambda > 0.8$ The lowest and flattest curve represent $\lambda = 1.0$, the MAP only gets 0.0001, and the R-Precision = 0, which is extremely low. It is because when $\lambda = 1.0$, the scoring function become $P_i = 1 * P + 0 * Q = P$, which do not consider the estimated probability from corpus, and only consider the probability from document without smoothing. And since there is probability that

$m_i = 0$, the whole score become 0 directly and wipe up the calculation done therefore the recall and precision is low.



	5	10	15	20	30	100	200	500	1000
λ=0.0	0.192	0.156	0.12	0.121	0.098	0.0416	0.0243	0.0106	0.0055
λ=0.2	0.256	0.234	0.2067	0.198	0.1653	0.092	0.0605	0.0336	0.02
λ=0.4	0.256	0.236	0.2093	0.198	0.1653	0.0926	0.0609	0.0338	0.02
λ=0.6	0.256	0.236	0.2107	0.198	0.1667	0.093	0.0309	0.0339	0.0201
λ=0.8	0.26	0.24	0.2093	0.198	0.168	0.0936	0.0611	0.034	0.0201
λ=1.0	0	0	0	0	0	0	0.0003	0.0002	0.0003

Figure 13: Recall - Precision



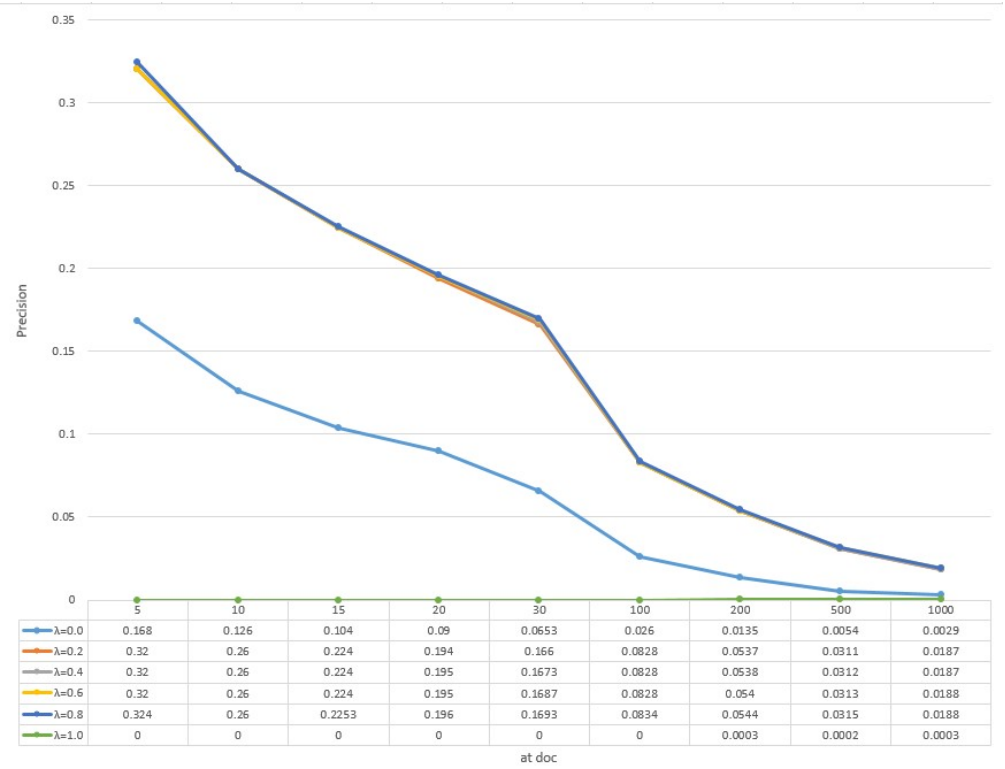
	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
λ=0.0	0.2969	0.1391	0.0413	0.0382	0.027	0.0218	0.0151	0	0	0	0
λ=0.2	0.533	0.3353	0.2461	0.162	0.1203	0.0811	0.0472	0.0312	0.0162	0.0088	0.0062
λ=0.4	0.5434	0.3459	0.2568	0.1626	0.1208	0.0816	0.0474	0.0329	0.0177	0.0088	0.0062
λ=0.6	0.5434	0.3446	0.2582	0.1646	0.1213	0.083	0.0476	0.0331	0.0179	0.0088	0.0062
λ=0.8	0.5445	0.3452	0.259	0.1656	0.1221	0.0851	0.0479	0.034	0.0186	0.0088	0.0062
λ=1.0	0.0008	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0	0	0	0

Figure 14: Uninterpolated precision

λ	0.0	0.2	0.4	0.6	0.8	1.0
Total Precision	0.1211	0.1383	0.4388	0.4401	0.4414	0.0070
Total Recall	0.0052	0.0199	0.0200	0.0201	0.0201	0.0003
MAP	0.0571	0.1143	0.1174	0.1179	0.1186	0.0001
R-Precision	0.0916	0.1658	0.1666	0.167	0.1679	0

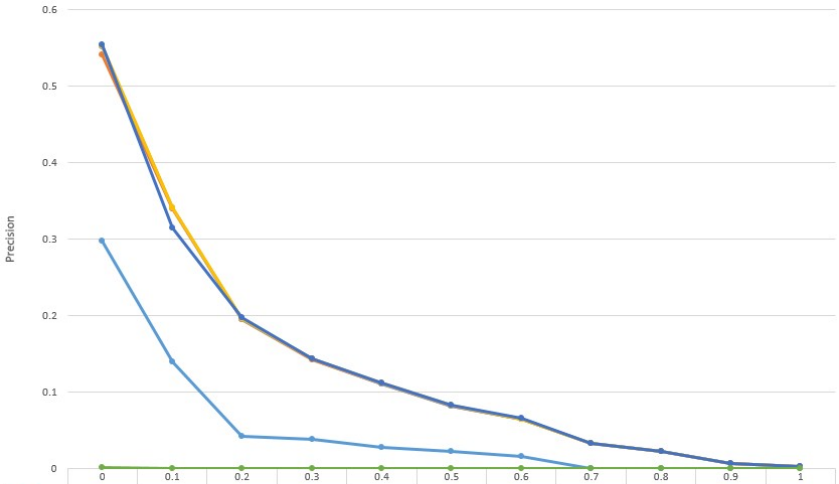
- **Models without Stemming**
  In Figure15 and Figure16, the closer λ increase approach to 0.8, , the higher interpolated recall precision and uninterpolated precision it get. And the highest record of MAP is 0.1148, and the highest R-Precision is 0.1516, where $\lambda = 0.8$. However, the precision drop dramatically when

$\lambda > 0.8$ The lowest and flattest curve represent $\lambda = 1.0$, the MAP only gets 0.0001, and the R-Precision = 0, which is extremely low.



	5	10	15	20	30	100	200	500	1000
λ=0.0	0.168	0.126	0.104	0.09	0.0653	0.026	0.0135	0.0054	0.0029
λ=0.2	0.32	0.26	0.224	0.194	0.166	0.0828	0.0537	0.0311	0.0187
λ=0.4	0.32	0.26	0.224	0.195	0.1673	0.0828	0.0538	0.0312	0.0187
λ=0.6	0.32	0.26	0.224	0.195	0.1687	0.0828	0.054	0.0313	0.0188
λ=0.8	0.324	0.26	0.2253	0.196	0.1693	0.0834	0.0544	0.0315	0.0188
λ=1.0	0	0	0	0	0	0	0.0003	0.0002	0.0003

Figure 15: Recall - Precision



	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
λ=0.0	0.2969	0.1391	0.0413	0.0382	0.027	0.0218	0.0151	0	0	0	0
λ=0.2	0.541	0.34	0.1948	0.1422	0.1105	0.0816	0.0638	0.0324	0.0217	0.0058	0.0017
λ=0.4	0.5517	0.3409	0.195	0.1425	0.1108	0.0818	0.0639	0.0325	0.0218	0.0058	0.0017
λ=0.6	0.5522	0.3411	0.1963	0.1426	0.1112	0.082	0.0646	0.0326	0.0218	0.0058	0.0017
λ=0.8	0.5539	0.3144	0.1966	0.1435	0.1118	0.0822	0.065	0.0327	0.022	0.0058	0.0017
λ=1.0	0.0008	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0	0	0	0

Figure 16: Uninterpolated precision

λ	0.0	0.2	0.4	0.6	0.8	1.0
Total Precision	0.065	0.409	0.4102	0.4125	0.4133	0.0070
Total Recall	0.0030	0.0186	0.0187	0.0188	0.0189	0.0003
MAP	00.0381	0.1122	0.1127	0.1131	0.1138	0.0001
R-Precision	0.0644	0.1518	0.1516	0.1516	0.1516	0

6 Compare Laplace and JM Smoothing Techniques

- **Models With Stemming**
  In Figure17, the upper curve is LM with JM Smoothing and the lower curve is LM with Laplace Smoothing. JM Smoothing gets higher precision than Laplace Smoothing.
  In Figure18, the upper curve is LM with JM Smoothing and the lower curve is LM with Laplace Smoothing. JM Smoothing gets a lot higher precision than Laplace Smoothing before retrieveing 1000 documents.

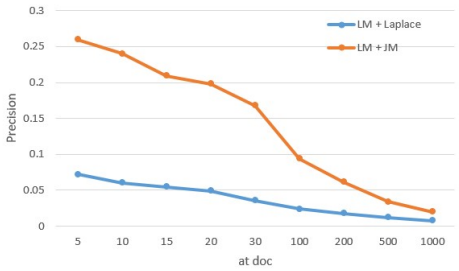


Figure 17: Recall - Precision

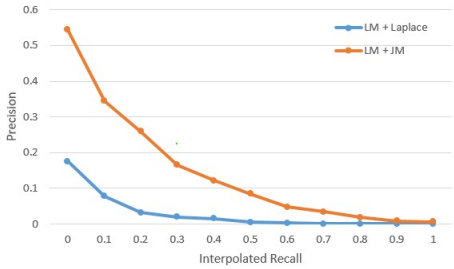


Figure 18: Uninterpolated precision

	LM + Laplace	LM + JM
Total Precision	0.01619	0.4414
Total Recall	0.0074	0.0201
P@10	0.06	0.24
MAP	0.0203	0.1186
R-Precision	0.0443	0.1679

- **Models without Stemming**
  In Figure19, the upper curve is LM with JM Smoothing and the lower curve is LM with Laplace Smoothing. JM Smoothing gets higher precision than Laplace Smoothing.
  In Figure20, the upper curve is LM with JM Smoothing and the lower curve is LM with Laplace Smoothing. JM Smoothing gets a lot higher precision than Laplace Smoothing before retrieving 1000 documents.

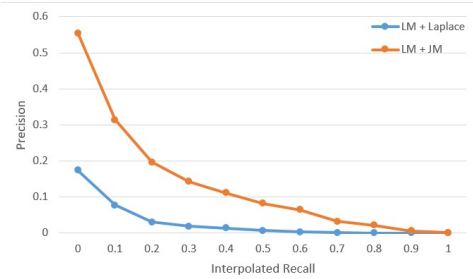


Figure 19: Recall - Precision

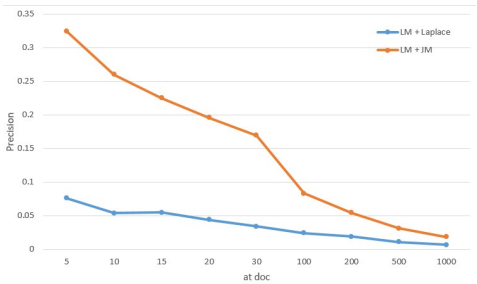


Figure 20: Uninterpolated precision

	LM + Laplace	LM + JM
Total Precision	0.154	0.413
Total Recall	0.007	0.0188
P@10	0.054	0.26
MAP	0.0202	0.1138
R-Precision	0.0386	0.1516

7 Compare Okapi, LM with Laplace Smoothing, and LM with JM Smoothing

Models with Stemming

	Okapi	LM + Laplace	LM + JM	Improved LM + Laplace
Total Precision	0.4498	0.01619	0.4414	0.1084
Total Recall	0.021	0.0074	0.0201	0.0049
P@10	0.27	0.06	0.24	0.042
MAP	0.1452	0.0203	0.1186	0.0096
R-Precision	0.1809	0.0443	0.1679	0.0248

Models without Stemming

	Okapi	LM + Laplace	LM + JM	Improved LM + Laplace
Total Precision	0.3887	0.154	0.413	0.1685
Total Recall	0.0177	0.007	0.0188	0.0077
P@10	0.292	0.054	0.26	0.056
MAP	0.1266	0.0202	0.1138	0.0213
R-Precision	0.1553	0.0386	0.1516	0.0415

7.1 Recall and Precision

The definition of Precision is the total number of documents retrieved that are relevant in total number of documents that are retrieved. And the definition of Recall is the total number of documents retrieved that are relevant in total number of relevant documents in the database.

- **Models with Stemming**
  The rankings of precision when interpolated recall of models is < 0.2 : LM with JM smoothing, VSM with Okapi TF, LM with Laplace smoothing, Improved LM with Laplace smoothing.
  The rankings of precision when interpolated recall of models is $>= 0.45$ and $<= 0.6$: VSM with Okapi TF, LM with JM smoothing, Improved LM with Laplace smoothing, LM with Laplace smoothing.
  The rankings of precision when interpolated recall of models is > 0.6 : VSM with Okapi TF, LM with JM smoothing, Improved LM with Laplace smoothing = LM with Laplace smoothing.

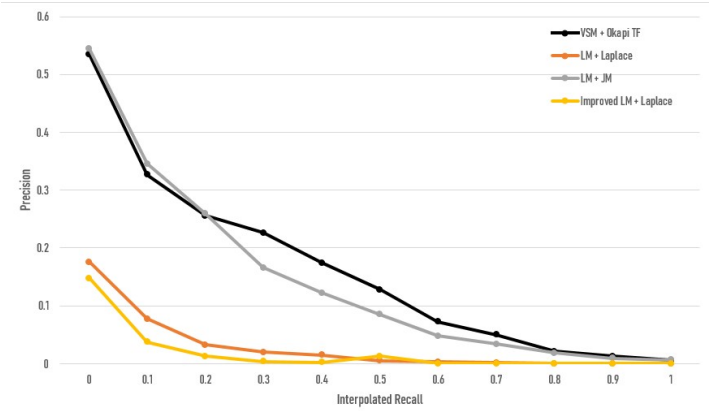


Figure 21: Interpolated Recall - Precision Averages (with Stemming)

- **Models without Stemming**
  The rankings of precision when interpolated recall of models is < 0.55 : VSM with Okapi TF, LM with JM smoothing, Improved LM with Laplace smoothing, LM with Laplace smoothing.
  The rankings of precision when interpolated recall of models is > 0.55 : LM with JM smoothing, VSM with Okapi TF, Improved LM with Laplace smoothing = LM with Laplace smoothing.
  The rankings of precision when interpolated recall of models is > 0.6 : LM with JM smoothing, VSM with Okapi TF, Improved LM with Laplace smoothing = LM with Laplace smoothing.

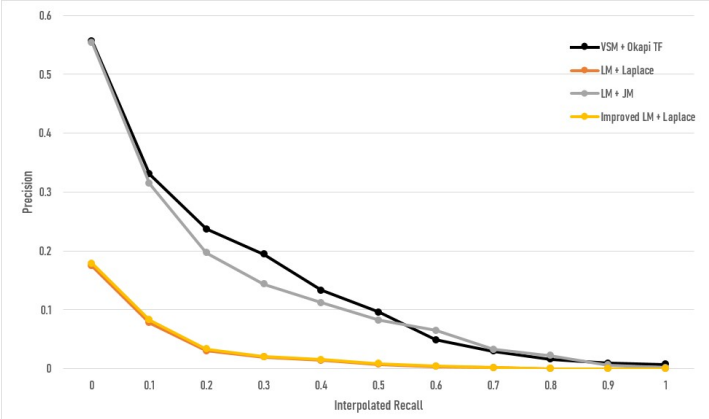


Figure 22: :Interpolated Recall - Precision Averages (without Stemming)

7.2 Un-interpolated Mean Average Precision

MAP (Mean Average Precision) is the average of AP over all queries.
Figure23 represents the MAP of the 8 models. The Bar Chart from the right to the left were models each in stemming condition and without stemming condition.
The Uninterpolated precision 10 of the 8 models are as Figure24 :VSM with Okapi Tf, Language Modeling with JM smoothing, Language Modeling with Laplace Smoothing, Improved Language Modeling with Laplace Smoothing.
Of which Improved LM with Laplace Smoothing have slightly higher MAP than Orginal LM with Laplace Smoothing under without stemming condition.

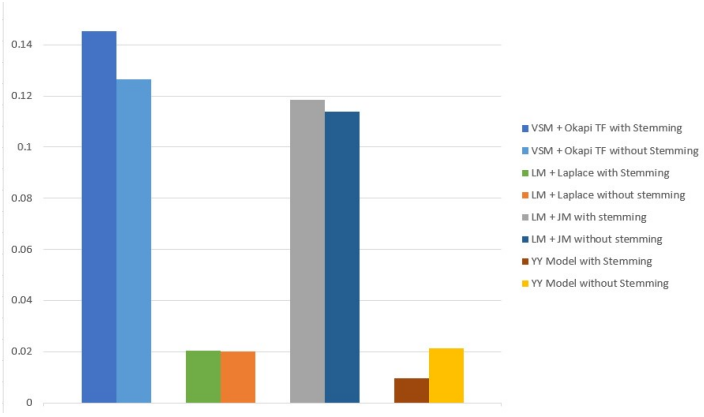


Figure 23: MAP Chart

7.3 R-Precision at doc 10

R-Precision is that for a given query topic Q, R-precision is the precision at R, where R is the number of relevant documents for Q. In other words, if there are r relevant documents among the top-R retrieved documents, then R-precision is $\frac{r}{R}$.
The Precision at document 10 of the 8 models are as Figure24 :VSM with Okapi Tf, Language Modeling with JM smoothing, Language Modeling with Laplace Smoothing, Improved Language Modeling with Laplace Smoothing.

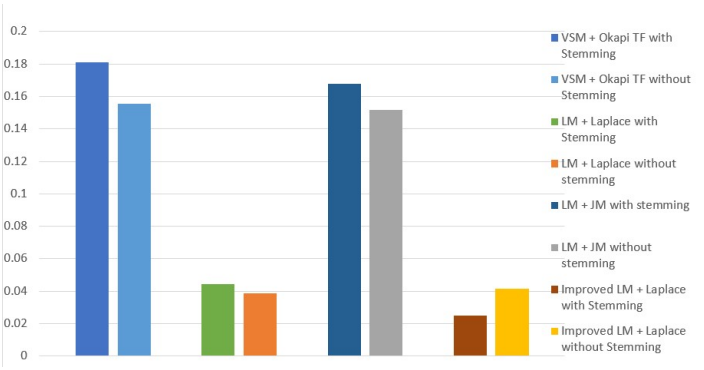


Figure 24: R-Precision Chart

8 Effects of Stemming

Stemming algorithms (stemmers) are used to convert the words to their root form (stem); this process is used in the pre-processing stage of the Information Retrieval Systems. [HIBT17] In this paper, I use the Porter stemming algorithm, it is a process for removing the commoner morphological and inflexional endings from words in English. Its main use is as part of a term normalisation process that is usually done when setting up Information Retrieval systems.[Por]
According to Figures23 and Figure24, most right bar chart are higher or equal to the left side as initially expected. Which means that the models with stemming mostly gets higher recall and precision than non-stemming ones.

References

[HIBT17] Safaa I Hajeer, Rasha M Ismail, Nagwa L Badr, and Mohamed Fahmy Tolba. A new stemming algorithm for efficient information retrieval systems and web search engines. In *Multimedia Forensics and Security*, pages 117–135. Springer, 2017.

[HM13] Amir Hazem and Emmanuel Morin. A comparison of smoothing techniques for bilingual lexicon extraction from comparable corpora. In *Proceedings of the Sixth Workshop on Building and Using Comparable Corpora*, pages 24–33, 2013.

[Jay] Vaibhav Jayaswal. Laplace smoothing in naïve bayes algorithmn. https://towardsdatascience.com/laplace-smoothing-in-na%C3%AFve-bayes-algorithm-9c237a8bdece.

[Lut] Ben Lutkevich. What is language modeling? https://www.techtarget.com/searchenterpriseai/definition/language-modeling.

[Por] The porter stemming algorithm. https://tartarus.org/martin/PorterStemmer/.

[Sei] Rudi Seitz. Understanding tf-idf and bm-25. https://kmwllc.com/index.php/2020/03/20/understanding-tf-idf-and-bm-25/.

[Ste] Bruno Stecanella. Understanding tf-id: A simple introduction. https://monkeylearn.com/blog/what-is-tf-idf/.

[Tsa] Ming-Feng Tsai. Project 2 introduction. https://wm5.nccu.edu.tw/base/10001/course/10026264/content/proj02/index.html.