WSM Project 2: Building IR systems based on the Pyserini Project

Thesis/Dissertation by

Hsu Shao Wen

Contents

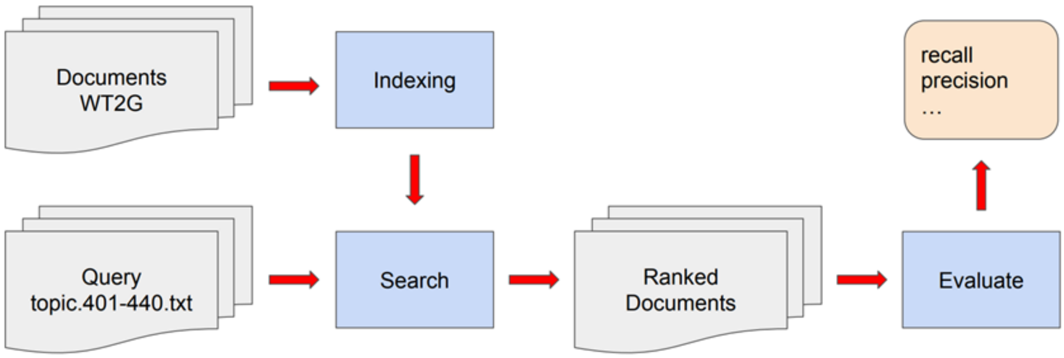
Chapter 1

Introduction

The research calculates relevance scores for each document for given user queries in information retrieval. To achieve this objective, we employ various retrieval methodologies, including the OKAPI BM25 language model and other language models incorporating distinct smoothing techniques, such as Maximum Likelihood Estimates with Laplace smoothing and Jelinek-Mercer smoothing. Implementing these models is designed to provide accurate assessments of document relevance concerning specific queries, thereby optimizing performance in information retrieval systems.

Chapter 2

Data and Methods

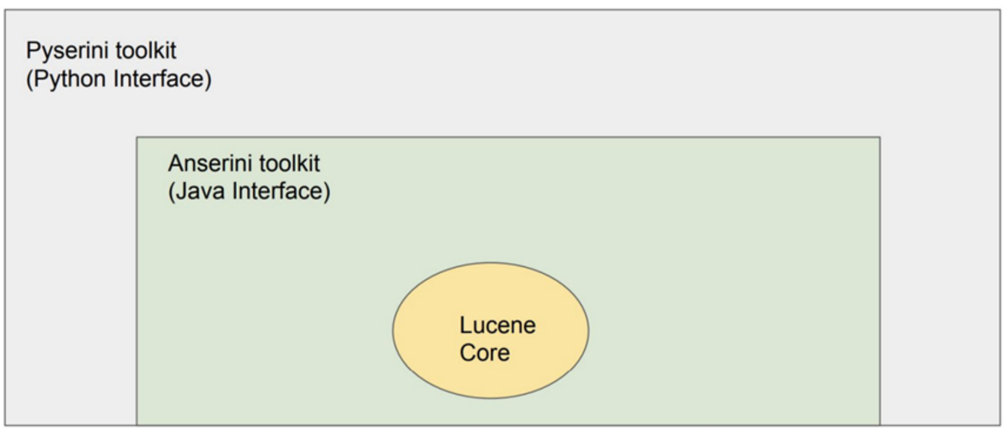


1. **Corpus**

   The utilized corpus for this study is the WT2g dataset, comprising 2GB of web documents. This collection consists of 40 TREC queries formatted according to the standard TREC format, which includes topic titles, descriptions, and narratives. NIST assessors manually annotated the documents in the corpus based on their relevance to these queries.

2. **Pyserini**

   To support indexing, searching, scoring and evaluation functionalities, we utilized Pyserini's toolkit, which leverages Anserini/Lucene JAVA classes. Anserini is a bridge that invokes Apache Lucene, a highly efficient and feature-rich full-text search engine library.



3. **Indexing**

   For the WT2g corpus, we constructed two indexes: one utilizing Porter stemming and the other without Porter stemming. The application of Porter stemming aims to enhance retrieval accuracy by reducing words to their root forms. These indexes facilitate efficient retrieval operations.

Index Description	Statistics (Pyserini TrecwebCollection)
WT2G with stemming	terms=184,971,623 unique_terms=1,674,417 docs=246,772
WT2G without stemming	terms=184,971,623 unique_terms=1,834,566 docs=246,772

4. **Searching**

   Subsequently, we executed 40 TREC queries against the WT2g corpus, returning a ranked list of relevant documents for each query (top 1000). We employed diverse language models, including the OKAPI BM25 model and two language models incorporating different smoothing techniques: Maximum Likelihood Estimates with Laplace smoothing and Jelinek-Mercer smoothing.

5. **Ranking and Performance Evaluation of Language Models**

   The scoring of retrieved documents involved the application of three plus one distinct language models and evaluate performance of each model via treceval.pl. The language model formulas are as follows:

   - OKAPI BM25

tf / tf + k1((1 - b) + b * doclen / avgdoclen)

set k1 = 2 and b = 0.75

   - maximum-likelihood with Laplace smoothing

$$\rho_i = \frac{m_i+1}{n+\frac{t}{k}} + \frac{\frac{t-k}{k}P(w|C)}{n+\frac{t}{k}}$$

where m = term frequency, n=number of terms in document (doc length) , k=number of unique terms in corpus, t=total terms in corpus, and P(w/C) is the estimated probability from corpus, background probability.

   - Jelinek-Mercer smoothing

$$\rho_i = \lambda P(w|D) + (1-\lambda)P(w|C)$$

where P(w/D) is the estimated probability from document and P(w/C) is the estimated probability from corpus, background probability.

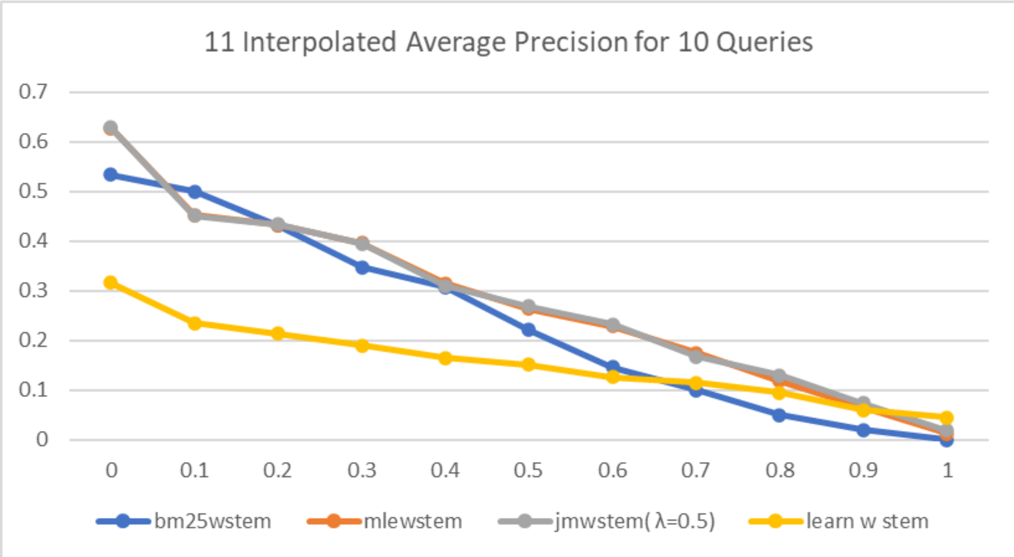
- Learn to Rank

Ensemble above three scores as features to train XGBoost model to improve the ranking.

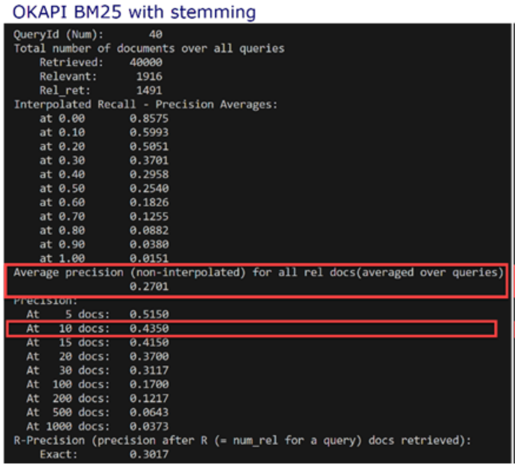
Chapter 3

Results

1. **Advantages or Disadvantages of Stemming**

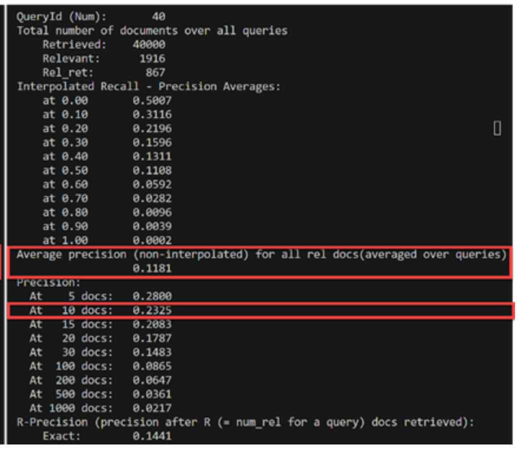


OKAPI BM25 with stemming



```
QueryId (Num):      40
Total number of documents over all queries
    Retrieved:    40000
    Relevant:      1916
    Rel_ret:       1491
Interpolated Recall - Precision Averages:
    at 0.00       0.8575
    at 0.10       0.5993
    at 0.20       0.5051
    at 0.30       0.3701
    at 0.40       0.2958
    at 0.50       0.2540
    at 0.60       0.1826
    at 0.70       0.1255
    at 0.80       0.0882
    at 0.90       0.0380
    at 1.00       0.0151
Average precision (non-interpolated) for all rel docs(averaged over queries)
                  0.2701
Precision:
  At    5 docs:   0.5150
  At   10 docs:   0.4350
  At   15 docs:   0.4150
  At   20 docs:   0.3700
  At   30 docs:   0.3117
  At  100 docs:   0.1700
  At  200 docs:   0.1217
  At  500 docs:   0.0643
  At 1000 docs:   0.0373
R-Precision (precision after R (= num_rel for a query) docs retrieved):
    Exact:        0.3017
```

OKAPI BM25 without stemming



```
QueryId (Num):      40
Total number of documents over all queries
    Retrieved:    40000
    Relevant:      1916
    Rel_ret:        867
Interpolated Recall - Precision Averages:
    at 0.00       0.5007
    at 0.10       0.3116
    at 0.20       0.2196
    at 0.30       0.1596
    at 0.40       0.1311
    at 0.50       0.1108
    at 0.60       0.0592
    at 0.70       0.0282
    at 0.80       0.0096
    at 0.90       0.0039
    at 1.00       0.0002
Average precision (non-interpolated) for all rel docs(averaged over queries)
                  0.1181
Precision:
  At    5 docs:   0.2800
  At   10 docs:   0.2325
  At   15 docs:   0.2083
  At   20 docs:   0.1787
  At   30 docs:   0.1483
  At  100 docs:   0.0865
  At  200 docs:   0.0647
  At  500 docs:   0.0361
  At 1000 docs:   0.0217
R-Precision (precision after R (= num_rel for a query) docs retrieved):
    Exact:        0.1441
```

maximum likelihood estimates with Laplace smoothing
with stemming

```
QueryId (Num):       40
Total number of documents over all queries
    Retrieved:    40000
    Relevant:      1916
    Rel_ret:       1490
Interpolated Recall - Precision Averages:
    at 0.00       0.7587
    at 0.10       0.5040
    at 0.20       0.4249
    at 0.30       0.3373
    at 0.40       0.2631
    at 0.50       0.2368
    at 0.60       0.1694
    at 0.70       0.1195
    at 0.80       0.0893
    at 0.90       0.0575
    at 1.00       0.0209
Average precision (non-interpolated) for all rel docs(averaged over queries)
                  0.2391
Precision:
  At    5 docs:   0.4600
  At   10 docs:   0.3900
  At   15 docs:   0.3667
  At   20 docs:   0.3350
  At   30 docs:   0.2917
  At  100 docs:   0.1720
  At  200 docs:   0.1207
  At  500 docs:   0.0649
  At 1000 docs:   0.0373
R-Precision (precision after R (= num_rel for a query) docs retrieved):
    Exact:        0.2776
```

maximum likelihood estimates with Laplace smoothing
without stemming

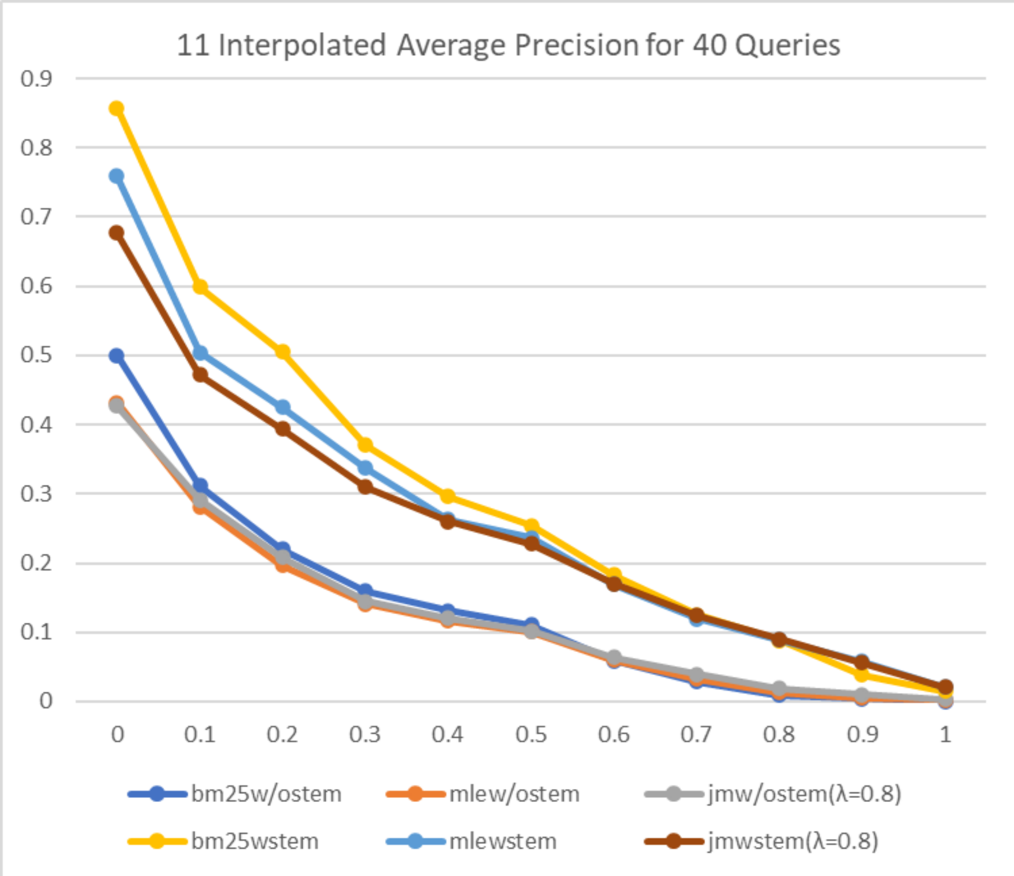
```
QueryId (Num):       40
Total number of documents over all queries
    Retrieved:    40000
    Relevant:      1916
    Rel_ret:        841
Interpolated Recall - Precision Averages:
    at 0.00       0.4322
    at 0.10       0.2813
    at 0.20       0.1964
    at 0.30       0.1414
    at 0.40       0.1164
    at 0.50       0.1011
    at 0.60       0.0600
    at 0.70       0.0322
    at 0.80       0.0133
    at 0.90       0.0049
    at 1.00       0.0026
Average precision (non-interpolated) for all rel docs(averaged over queries)
                  0.1065
Precision:
  At    5 docs:   0.2350
  At   10 docs:   0.2150
  At   15 docs:   0.1950
  At   20 docs:   0.1725
  At   30 docs:   0.1533
  At  100 docs:   0.0837
  At  200 docs:   0.0595
  At  500 docs:   0.0336
  At 1000 docs:   0.0210
R-Precision (precision after R (= num_rel for a query) docs retrieved):
    Exact:        0.1349
```

Jelinek-Mercer smoothing(lambda=0.5)
with stemming

```
QueryId (Num):       40
Total number of documents over all queries
    Retrieved:    40000
    Relevant:      1916
    Rel_ret:       1522
Interpolated Recall - Precision Averages:
    at 0.00       0.7789
    at 0.10       0.5078
    at 0.20       0.4412
    at 0.30       0.3385
    at 0.40       0.2900
    at 0.50       0.2573
    at 0.60       0.1819
    at 0.70       0.1309
    at 0.80       0.1102
    at 0.90       0.0610
    at 1.00       0.0217
Average precision (non-interpolated) for all rel docs(averaged over queries)
                  0.2497
Precision:
  At    5 docs:   0.4400
  At   10 docs:   0.4025
  At   15 docs:   0.3617
  At   20 docs:   0.3225
  At   30 docs:   0.2900
  At  100 docs:   0.1745
  At  200 docs:   0.1209
  At  500 docs:   0.0658
  At 1000 docs:   0.0381
R-Precision (precision after R (= num_rel for a query) docs retrieved):
    Exact:        0.2984
```

Jelinek-Mercer smoothing(lambda=0.5)
without stemming

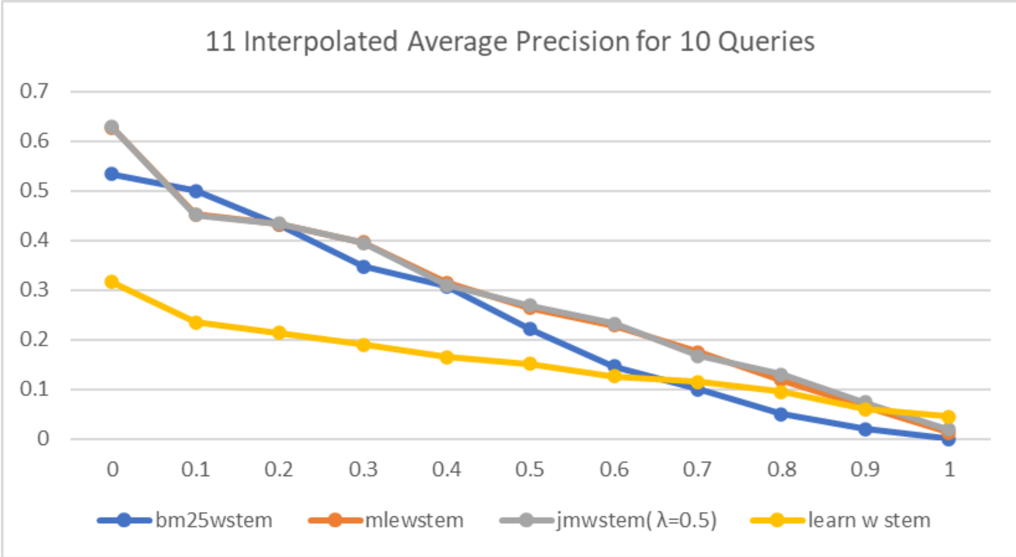
```
QueryId (Num):       40
Total number of documents over all queries
    Retrieved:    40000
    Relevant:      1916
    Rel_ret:        861
Interpolated Recall - Precision Averages:
    at 0.00       0.4275
    at 0.10       0.2909
    at 0.20       0.2087
    at 0.30       0.1447
    at 0.40       0.1204
    at 0.50       0.1021
    at 0.60       0.0637
    at 0.70       0.0394
    at 0.80       0.0193
    at 0.90       0.0102
    at 1.00       0.0033
Average precision (non-interpolated) for all rel docs(averaged over queries)
                  0.1115
Precision:
  At    5 docs:   0.2400
  At   10 docs:   0.2150
  At   15 docs:   0.2083
  At   20 docs:   0.1875
  At   30 docs:   0.1550
  At  100 docs:   0.0857
  At  200 docs:   0.0571
  At  500 docs:   0.0347
  At 1000 docs:   0.0215
R-Precision (precision after R (= num_rel for a query) docs retrieved):
    Exact:        0.1445
```

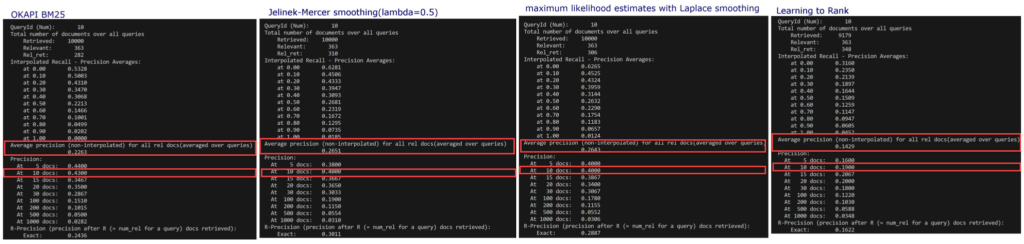


Upon applying stemming to the text, the performance metrics, including Maximum Likelihood Estimates (MLE), BM25, and Jelinek-Mercer, exhibited a notable improvement, approximately doubling their original values. Stemming, which involves reducing words to their root forms, evidently enhanced the effectiveness of the language models.

2. **Different Smoothing Techniques and Learning to Rank**

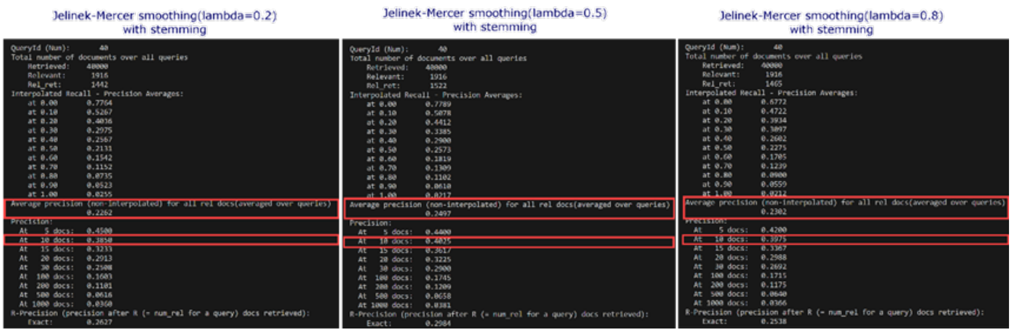
   Given the substantial improvement in performance observed after stemming, we focused on the stemmed index for further analysis. We compared the performance of the Jelinek-Mercer model (with =0.5) with the other two language models (MLE and BM25) across 10 queries.

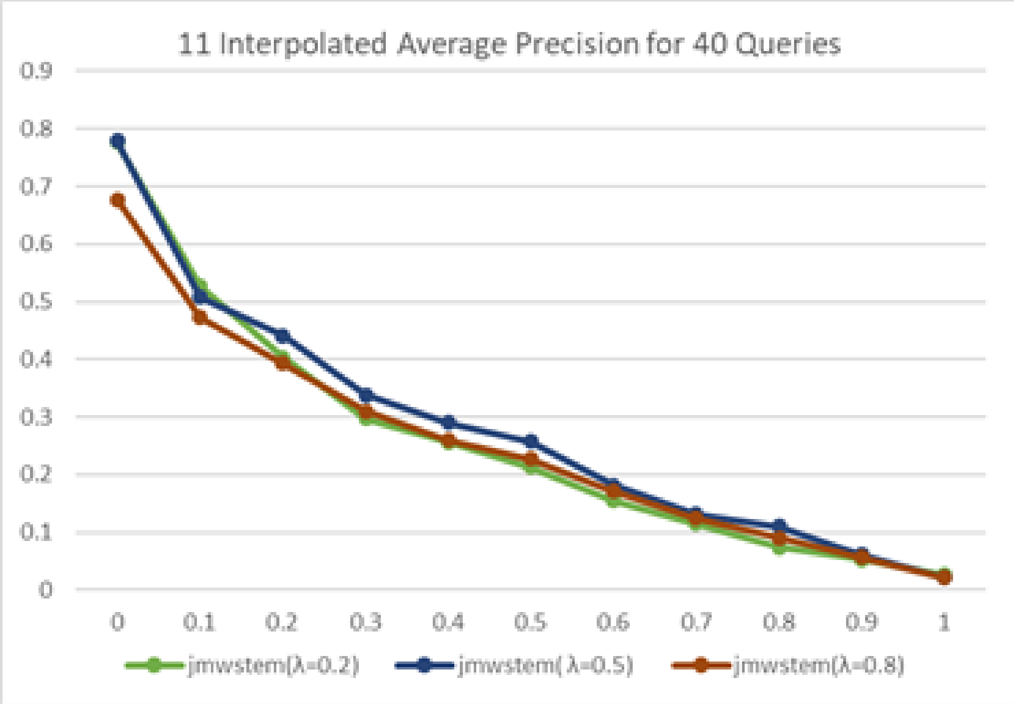




Performance Comparison: Jelinek-Mercer vs. MLE vs. BM25: Jelinek-Mercer and MLE demonstrated comparable performance, both outperforming BM25. While machine learning approaches, specifically XGBoost, exhibited the poorest performance in terms of Mean Average Precision (MAP) and Precision at 10, they excelled in retrieving the highest proportion of relevant documents. Notably, users typically focus on the top few results during retrieval, making the machine learning approach less advantageous in this experimental context.

3. **Varying Lambda for Jelinek-Mercer Smoothing**





In the experimentation with Jelinek-Mercer smoothing, the performance varied significantly with different parameter settings. The observed performance ranking for different values of lamda: 0.5 setting is better than 0.2 or 0.8. This implies that a moderate smoothing parameter (lamda0.5) resulted in the highest performance, while extreme values (lamda0.2 and lamda0.8) were less effective in improving retrieval accuracy.

Chapter 4

Concluding Remarks

In conclusion, stemming significantly enhanced the effectiveness of the language models. The comparison across different smoothing techniques and the incorporation of a machine learning approach highlighted nuanced trade-offs in performance metrics. While traditional language models like Jelinek-Mercer and MLE exhibited superior performance in precision-related metrics, the machine learning approach, despite lower MAP and Precision at 10, demonstrated a unique strength in retrieving a higher volume of relevant documents. Ultimately, the choice of methodology depends on the specific priorities and preferences in information retrieval scenarios.