CS 6200 HW1

Parth Shah

February 4, 2023

Introduction:

The objective of this report is to describe the process of developing an information retrieval system using Elasticsearch, Kibana, and Docker. The system was designed to retrieve relevant documents for a set of queries by preprocessing the documents and queries, building retrieval models, and implementing pseudo-relevance feedback.

Step 1: Installation and Data Collection:

Docker, Kibana, and Elasticsearch were installed, and the necessary data was downloaded.

Step 2: Indexing the Documents:

A parser.py script was used to create an index with Elasticsearch for all the documents. The documents were preprocessed using nltk to remove stop words and stem words. The preprocessed documents were added to the index using the bulk add with relevant information such as doc-id and text.

Step 3: Preprocessing Queries:

The queries were preprocessed using preprocess_query.py by removing stop words and stemming the words left.

Step 4: Retrieving Term Vectors:

prepare_stats.py was used to retrieve term vectors for all the doc_ids, and the results were saved to a pickle file.

Step 5: Building Retrieval Models:

Retrieval models were built using .py files and the following general implementation:
-Iterating over each document in the corpus
-For each document, iterating over each query
-For each query, iterating over each word
-Calculating the score for each query-document combination
-Adding the scores to a dictionary and summing up the scores
-Sorting the scores, taking the first 1000 per query, and writing them to a .txt output file.

Step 6: Running Queries and Comparing Results:

Various retrieval models were run using the queries, and the results were compared using trec_eval.

Step 7: Adjusting Queries:

The queries were adjusted if needed until the precision threshold was reached.

Step 8: Implementing Pseudo-Relevance Feedback:

A custom pseudo-relevance feedback function was created in pseudo_relevance.py by:
-Getting the top K (k=1) documents for each query
-Calculating the TF-IDF score for each term in those documents
-Appending the top X (x=2) scores from those terms to the query
-Rerunning the model with the extended query.

Step 9: Implementing Pseudo-Relevance Feedback using ES aggs:

Pseudo-relevance feedback was also implemented using the "significant terms" API in pseudo_relevance_aggs.py by:
-Finding the top K (k=2) terms for each query word using the "significant terms" API
-Scoring the terms using IDF scores
-Adding the top X (x=1) terms from the IDF scoring to the query
-Rerunning the models.

Conclusion:

The results showed that the Okapi BM25 and TF-IDF models met the precision threshold for the vector space or probabilistic models (.28), while Jelinek-Mercer met the precision threshold for the language models (.25). Performances of Okapi TF and Laplace models improved using pseudo-relevance feedback, while the second pseudo-relevance model performed poorly. This may be due to the former using TF-IDF scores and the latter using the "significant terms" API and IDF scores.

In conclusion, the information retrieval system was successfully developed using Elasticsearch, Kibana, and Docker, and the results have been attached to this report.

Baseline Retrieval Models			
Model	**Uninterpolated Mean Average Precision**	**Precision at 10 documents**	**Precision at 30 documents**
ES Inbuilt	0.2978	0.4480	0.3653
Okapi TF	0.2460	0.4360	0.3307
TF-IDF (above par score of 0.28)	0.2879	0.4560	0.3613
Okapi BM25 (above par score of 0.28)	0.2948	0.4440	0.3640
Laplace	0.2271	0.4280	0.3280
Jelinek-Mercer (above par score of 0.25)	0.2505	0.3520	0.3200

Custom Pseudo-relevance Feedback			
Model	**Uninterpolated Mean Average Precision**	**Precision at 10 documents**	**Precision at 30 documents**
ES Inbuilt	0.2843	0.4360	0.3427
Okapi TF (increased from basedline)	0.2548	0.4080	0.3440
TF-IDF	0.2780	0.4200	0.3413
Okapi BM25	0.2852	0.4240	0.3413
Laplace (increased from basedline)	0.2498	0.4360	0.3387
Jelinek-Mercer	0.2495	0.3600	0.3227

ES "significant terms" Pseudo-relevance Feedback			
Model	**Uninterpolated Mean Average Precision**	**Precision at 10 documents**	**Precision at 30 documents**
ES Inbuilt	0.2209	0.3560	0.2760
Okapi TF	0.1831	0.3320	0.2560
TF-IDF	0.2215	0.3160	0.2827
Okapi BM25	0.2243	0.3600	0.2787
Laplace	0.1682	0.3320	0.2480
Jelinek-Mercer	0.1972	0.2840	0.2333