Homework-6 Report

Rohith Chandra Kandambeth

1. **Data Preparation**
    a. **Data Selection**
        i. **Description of the QREL-based document selection process**

        The QREL file was accessed to retrieve both the relevant and non-relevant. The fourth value is every line in the file represents the relevance of the document, if the value is one the document is relevant for that query and zero would mean otherwise.

        ii. **Explanation of non-relevant document inclusion**

        There are some non-relevant documents in the QRELs file, but the total number of rows is not enough to train a good model. We collect more data which was not present in the QRELs file and mark the values having no relevance.

    b. **Data Splitting**
        i. **Methodology for splitting data into training and testing queries**

        We first split the data in train and test by randomly select 5 query ids for the testing set. The data for these 5 queries are stored in the test set and the data of remaining 20 queries are stored in the training set.

2. **Feature Extraction**
    a. **Document-Query IR Features**
        i. **Detailed explanation of the feature extraction process**

        All the retrieval models were run again, and the result was saved for all the query documents pair. The scores were retrieved and added to the dataframe.

        ii. **List of IR models and features used (e.g., BM25, Language Models)**
        Okapitf, TFIDF, okapiBM25, Unigram with Laplace Smoothing, Unigram JM

        iii. **Approach to handling documents outside the top 1000 rankings**
        Scores for all the query and documents were saved to a file for each retrieval model. Since there are not many documents in the qrels file the remaining were extracted using the results from the retrieval models.

b. **Additional Features**
    i. **Description of any additional features used (e.g., document length, PageRank)**

Additional features include document length and query length.

3. **Machine Learning Model**
    a. **Training the Model**
        i. **Description of the learning algorithm(s) used**

I used two ML algorithms in this assignment – Logistic Regression and SVM. Logistic Regression is a statistical model that uses a logistic function to model a binary dependent variable, whereas SVM (Support Vector Machine) is a robust classification technique that finds the hyperplane that best separates different classes by maximizing the margin between the closest data points of classes.

    b. **Model Testing and Evaluation**
        i. **Methodology for testing the model on the 5 testing queries**

The testing data with the features are passed to predict method after training. Specifically, predict_proba method for target value 1 which gives the probability of the getting the target value 1. Using the probability value the documents were ranked for each query.

        ii. **Approach for evaluating the model on the 20 training queries**

The same approach as with the testing data was followed in the training data. All the documents were ranked for each query based on the probability value of attaining target value 1.

        iii. **Description of treceval application and results interpretation**

After the documents were ranked for each query and saved to the file, the trec eval was run on the result file. Overall, the performance metrics indicate a general trend of decreasing precision as more documents are retrieved, with some queries performing exceptionally well in early retrieval stages (high precision at low document counts) but significantly worse as the number of retrieved documents increases.

4. Results and Analysis

a. Testing Performance

i. Presentation and analysis of results from testing queries

	precision	recall	f1-score	support
0	0.92	0.94	0.93	5000
1	0.25	0.18	0.21	497
accuracy			0.88	5497
macro avg	0.58	0.56	0.57	5497
weighted avg	0.86	0.88	0.87	5497

The model performs well in correctly identifying the majority class (class 0) with high precision and recall, but it struggles significantly with the minority class (class 1), indicating a potential class imbalance issue. Overall accuracy is high, but the macro average scores, which give equal weight to both classes, suggest that the model's performance is not as effective for the less represented class.

ii. Detailed treceval results and interpretation

Queryid (Num): 56
Total number of documents over all queries
Retrieved: 1000
Relevant: 167
Rel_ret: 166
Interpolated Recall - Precision Averages:
at 0.00 1.0000
at 0.10 0.9565
at 0.20 0.9481
at 0.30 0.9481
at 0.40 0.9481
at 0.50 0.9397
at 0.60 0.9397
at 0.70 0.9379
at 0.80 0.9379
at 0.90 0.9042
at 1.00 0.0000
Average precision (non-interpolated) for all rel docs(averaged over queries)
0.9172
Precision:
At 5 docs: 1.0000
At 10 docs: 0.9000
At 15 docs: 0.9333
At 20 docs: 0.9500
At 30 docs: 0.9000

At 100 docs: 0.9300
At 200 docs: 0.8100
At 500 docs: 0.3320
At 1000 docs: 0.1660
R-Precision (precision after R (= num_rel for a query) docs retrieved):
Exact: 0.9042

Interpolated Recall-Precision Averages
Interpolation: These values at different recall points (from 0.00 to 1.00) show how precision varies as the recall increases. Higher precision at higher recall levels generally indicates better performance.
Drops at 1.00: The precision often drops to 0 at a recall of 1.00, which typically means that while trying to retrieve all relevant documents, many irrelevant documents are also retrieved, diluting the precision.
Average Precision
Non-interpolated: This metric averages the precision obtained after each relevant document is retrieved. It's a good measure of overall effectiveness across different levels of recall, independent of the number of retrieved documents.
Precision at Fixed Document Cuts
Early Retrieval: Precision at top-ranked documents (like at 5 or 10 docs) can indicate how well the system retrieves the most relevant documents first.
Decline Over More Documents: As more documents are retrieved (e.g., at 100, 200, 500, 1000 docs), precision generally declines, which is expected as more potentially irrelevant documents get included in the larger result sets.
R-Precision
Exact: This is the precision after retrieving a number of documents equal to the number of relevant documents for the query. It's a strong indicator of how well the retrieval system's ranking aligns with the distribution of relevant documents.

b. Training Performance

i. Discussion of the model's performance on training data

	precision	recall	f1-score	support
0	0.96	0.99	0.97	20000
1	0.68	0.30	0.42	1335
accuracy			0.95	21335
macro avg	0.82	0.65	0.70	21335
weighted avg	0.94	0.95	0.94	21335

The model demonstrates excellent performance on the majority class with very high precision and recall, while its performance on the minority class is weaker, especially in terms of recall. The overall accuracy is very high, and while the f1-score for the minority class is relatively low, this suggests that the model may have difficulty with class imbalance.

ii. Comparative analysis of training vs. testing performance

The model exhibits better performance on the training data across all metrics compared to the testing data, suggesting potential overfitting to the training set. The minority class (class 1) has notably better precision in the training data but suffers from low recall in both datasets, indicating the model's consistent difficulty with correctly identifying positive cases under class imbalance.

The above results are from Logistic Regression Model.

5. Extra Credit (If Attempted)

a. **Description of the extra credit task(s) undertaken. Show the results (if any**)

EC3: Advance Learning Algorithms
Used SVM to train and test data. The results are as follows:
Training:

	precision	recall	f1-score	support
0	0.96	1.00	0.98	20000
1	0.81	0.31	0.45	1198
accuracy			0.96	21198
macro avg	0.88	0.65	0.71	21198
weighted avg	0.95	0.96	0.95	21198

Testing:

	precision	recall	f1-score	support
0	0.92	0.98	0.95	5000
1	0.67	0.34	0.45	634
accuracy			0.91	5634
macro avg	0.80	0.66	0.70	5634
weighted avg	0.89	0.91	0.89	5634