HomeWork-6 Report

Karthik Chintamani Dileep

1. Data Preparation
    a. Data Selection
        i. Description of the QREL-based document selection process
        File qrels.adhoc.51-100.AP89.txt contains the relevance judgments for 25 queries from the AP89 dataset. The relevant document IDs for each query are extracted and stored in the qrel_docs dictionary.
        ii. Explanation of non-relevant document inclusion
        For each query, in addition to the relevant documents, 1000 non-relevant documents are included in the dataset. This is done by iterating through the scores from different ranking models (Elasticsearch, Okapi BM25, Okapi TF, TF-IDF, Unigram LML, Unigram LMJM) and selecting the top non-relevant documents until 1000 documents are collected.
    b. Data Splitting
        i. Methodology for splitting data into training and testing queries
        5-fold Cross validation to split the queries.

2. Feature Extraction
    a. Document-Query IR Features
        i. Detailed explanation of the feature extraction process
        Features are the various scores mentioned below. Calculated features for all queries vs all docs and stored in text files. If any score is missing from the top 1000 score files of hw1, I extracted it from the text file.
        ii. List of IR models and features used (e.g., BM25, Language Models)
        The features used are:
            - es_score
            - okapi_bm25_score
            - okapi_tf_score
            - tf_idf_score
            - unigram_lml_score
            - unigram_lmjm_score

        These features are scores obtained from various Information Retrieval (IR) models, such as Elasticsearch's built-in scoring, Okapi BM25, TF-IDF, and language models (unigram LML and LMJM).

        iii. Approach to handling documents outside the top 1000 rankings
        Mentioned above.

3. Machine Learning Model
    a. Training the Model
        i. Description of the learning algorithm(s) used
        Trained a Logistic Regression model from scikit-learn with L1 regularization (penalty='l1') and C=0.01 (regularization strength). The model is trained using the liblinear solver with a maximum of 1000 iterations.

    b. Model Testing and Evaluation
        i. Methodology for testing the model on the 5 testing queries
        Split the queries into training and testing sets using train_test_split from scikit-learn. The test set contains 5 queries (20% of the total queries). Used 5-fold cross validation for effective average metrics.
        ii. Approach for evaluating the model on the 20 training queries
        Ran the model on the training queries. The average precision of the model on the training data is 0.2944.
        iii. Description of treceval application and results interpretation
        the precision at 10 documents is 0.4800, which means that, on average, 4.8 out of the top 10 retrieved documents are relevant.

4. Results and Analysis
    a. Testing Performance
        i. Presentation and analysis of results from testing queries
        For the 5-fold cross-validation results, the average precision for each fold is as follows:
        Average precision 0: 0.5188
        Average precision 1: 0.1746
        Average precision 2: 0.2244
        Average precision 3: 0.2337
        Average precision 4: 0.3058
        The overall average precision across the 5 folds is (0.5188 + 0.1746 + 0.2244 + 0.2337 + 0.3058) / 5 = 0.29146.

        ii. Detailed treceval results and interpretation
    b. Training Performance
        i. Discussion of the model's performance on training data.
        The average precision of the model on the training data is 0.2944.
        ii. Comparative analysis of training vs. testing performance
        The model's performance on the training data (average precision: 0.2944) is slightly better than its performance on the random sample of the test data (average precision: 0.2735). However, the cross-validation results show a wide range of average precision values across the folds, with the overall average precision (0.29146) being comparable to the training performance. These results suggest that the model's performance may vary depending on the specific queries included in the training and testing sets.