Report on BM25 Information Retrieval System on openAlex dataset

SM3201434

September 4, 2025

1 Basic Idea

This project builds an Information Retrieval (IR) system using the BM25 ranking function. We take a collection of research papers, clean the text, turn it into tokens, and compute scores to find which papers best match a given query. The system adds query expansion using Pseudo-Relevance Feedback (PRF) and evaluates how well the system performs using standard IR metrics. To note is that there were 2 datasets one of 50 academic papers 'openalex_papers.csv' and another of 5000 'openalex_papers2.csv'. The ones described in this report are of the smaller dataset. The Queries on the other hand are inside the json folder ground_values.json

2 Libraries Used

- **pandas** – reading CSV files and handling tabular data.
- **re** – regular expressions for cleaning text.
- **numpy** – mathematical arrays and fast operations.
- **collections.Counter** – counting words easily.
- **nltk** – natural language tools like stopwords and lemmatization.
- **scipy.sparse** – efficient storage of document-term matrix.
- **time** – measuring search speed.
- **json** and **os** – reading ground truth files and file paths.
- **statistics** – used for measuring latency.

3 Preprocessing Function

The function `preprocess` prepares text for retrieval.

- **Lowercasing:** Treats "Apple" and "apple" as the same.
- **Regex cleaning:** Keeps only letters and numbers, removes symbols.
- **Tokenization:** Splits text into words (tokens).
- **Stopword removal:** Removes common words like "the" that add little meaning.
- **Lemmatization:** Reduces words to base form, e.g., "running" $\rightarrow$ "run", improving matches between queries and documents.
- **Bigrams:** Creates word pairs like "information_retrieval" to capture phrases.

4 BM25 Class

BM25 scores how relevant a document is to a query.

4.1 IDF Formula

$$idf(t) = \log \frac{N - df(t) + 0.5}{df(t) + 0.5} + 1$$

Explanation: - N is the total number of documents.
- $df(t)$ is how many documents contain term t.
- This gives higher scores to rare terms (more informative), lower scores to common words.
- Adding 1 ensures the score is positive.

Rare words like "unsupervised" in a corpus carry more distinguishing power.

4.2 BM25 Score Formula

$$score(d, q) = \sum_{t \in q} idf(t) \cdot \frac{f(t, d)(k_1 + 1)}{f(t, d) + k_1 \left(1 - b + b \cdot \frac{|d|}{avgdl}\right)}$$

Explanation: - $f(t, d)$ = frequency of term t in document d.
- $|d|$ = number of terms in document d.

- $avgdl$ = average document length in the collection.
- k_1 controls how term frequency affects score; b controls document length normalization.

Terms that appear more often in a document increase its score, but very long documents are normalized to prevent unfair advantage.

5 Pseudo-Relevance Feedback (PRF)

PRF expands a query with terms from top-ranked documents.

Steps:

1. Search with the original query.
2. Take top k documents.
3. Count terms that appear frequently in them.
4. Exclude original query terms.
5. Add the top frequent terms to the query.

PRF helps capture semantic context so documents that don't contain exact query words but are conceptually relevant can be retrieved.

6 Evaluation Metrics

- **Precision_k:** Proportion of top k results that are relevant.

$$P_k = \frac{|Retrieved_k \cap Relevant|}{k}$$

- **Recall_k:** Fraction of relevant documents retrieved in top k.

$$R_k = \frac{|Retrieved_k \cap Relevant|}{|Relevant|}$$

- **Average Precision (AP):** Average of precision values at ranks of relevant documents.
- **F1 Score:** Harmonic mean of precision and recall, balances both.

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

- **DCG / nDCG:** Measures ranking quality.

$$DCG = \sum_{i=1}^{k} \frac{2^{rel_i} - 1}{\log_2(i+1)}$$

- rel_i = relevance of document at rank i.
- nDCG normalizes DCG by the ideal ranking DCG.
Higher-ranked relevant documents contribute more to the score.

- **MRR:** Mean Reciprocal Rank, focuses on the position of the first relevant document.

$$MRR = \frac{1}{rank\ of\ first\ relevant\ doc}$$

7 Query Latency Measurement

The system measures the time to answer queries to ensure it is faster than reading the collection from disk. For R runs, each query q_i is timed as t_i.

Average latency:

$$\bar{t} = \frac{1}{R} \sum_{i=1}^{R} t_i$$

Standard deviation of latency:

$$\sigma_t = \sqrt{\frac{1}{R} \sum_{i=1}^{R} (t_i - \bar{t})^2}$$

Interpretation:
- $\bar{t}$ shows typical query speed.
- σ_t shows variability; a high σ_t means some queries are much slower or faster than average.

8 Evaluation

Average query latency over 100 runs is about:

$$\bar{x} = 0.006265 \text{ sec}, \quad \sigma = 0.008554 \text{ sec}$$

The relatively large standard deviation, $\sigma = 0.008554$, compared to the mean, $\bar{x} = 0.006265$ sec, indicates high variability: some queries may take much

longer than the average, while others are faster, reflecting inconsistent query latency across different inputs.

Average metrics over 59 queries:

Precision@10: 0.1155, Recall@10: 0.5948, AveragePrecision: 0.2707

F1@10: 0.1763, nDCG@10: 0.3827, MRR@10: 0.2879

Interpretation: - System retrieves many relevant documents (high recall) but top results include irrelevant docs (low precision).
- Average Precision and nDCG show relevant docs often appear early, but ranking is not perfect.
- F1 and MRR indicate overall retrieval quality and position of first relevant document are moderate.

Given the Query

> distance between nlp and ml

An example expansion of the query is:

> distance between nlp and ml learning data unsupervised dialect based

The learned terms added via pseudo-relevance feedback are:

> learning, data, unsupervised, dialect, based

Interpretation

1. **Effect of Query Expansion:**
   The original query was expanded using terms frequently appearing in top-ranked documents. This captures semantic context, allowing retrieval of documents that are relevant even if they do not contain the exact original words.

2. **Top Results Relevance:**
   The top five retrieved documents are:

   (a) Arabic Dialect Identification with an Unsupervised Learning (Based on a Lexicon). Application Case

   (b) Exploring Unsupervised Learning Techniques for the Internet of Things

(c) Generalized K-Harmonic Means – Dynamic Weighting of Data in Unsupervised Learning

(d) Supervised and Unsupervised Learning for Data Science

(e) An Overview on Unsupervised Learning from Data Mining Perspective

All top documents match the expanded query terms, especially `unsupervised`, `learning`, and `data`, showing that pseudo-relevance feedback effectively improves relevance.

3. **Score Distribution:**
   The highest score is 27.5407, while the fifth score is 13.6614. This indicates that some documents are much more strongly related to the expanded query than others.

4. **System Insight:**
   Pseudo-relevance feedback surfaces documents that the original query might miss due to vocabulary mismatch. Learned terms guide retrieval toward conceptually related papers, not only literal matches.

9 Conclusion

BM25-based retrieval system, enhanced with pseudo-relevance feedback, thus effectively improves access to relevant documents. Query expansion captures semantic context, allowing retrieval of papers that match the intended meaning even when exact query terms are absent. Evaluation shows high recall but moderate precision, indicating that while most relevant documents are retrieved, top results may include some irrelevant items. Metrics such as Average Precision, nDCG, F1, and MRR confirm that relevant documents generally appear early in the ranking, and the first relevant document is often positioned near the top. Overall, the system highlights the practical benefit of combining BM25 scoring with pseudo-relevance feedback for small-scale academic datasets.