Information Retrieval

Phase 1: Classical Information Retrieval on the IR2025 Collection

Evaluating BM25 Retrieval with Elasticsearch

May 2025

Nikos Mitsakis - 3210122

Maria Schoinaki - 3210191

In **Phase 1** of our project, we built and evaluated a **classical information‑retrieval pipeline** over the **IR2025 corpus** using **Elasticsearch's BM25 similarity**. We first preprocessed the **JSONL collection** with a **custom analyzer** (*standard tokenization, lowercase, English stop‑word removal, Krovetz stemming*), then **indexed** all documents using **streaming_bulk**. We implemented a **retrieval script** that issues a simple **match query** for each user query and collects the **top-k results** (*k = 20, 30, 50*). Finally, we evaluated **retrieval effectiveness** via **Mean Average Precision** at cutoff **k (MAP@k)** and **Precision@k (P@5, 10, 15, 20)** using both the Python **pytrec_eval API** and the **NIST trec_eval binary**.

---

1. Introduction

Information Retrieval (IR) systems **match** user queries to **relevant** documents within extensive collections. A persistent challenge is the **vocabulary gap**. Users and documents may use **different words** for the **same concept** (*e.g., "hiking" vs. "trekking"*), causing purely **term‑based models** to **miss relevant results**. **Phase 1** isolates this baseline behavior by **deploying a classical BM25 pipeline** on **IR2025 (trec-covid)** without any query expansion, thus quantifying the performance ceiling of **pure term‑matching models** and **identifying precise targets** for **improvement** in **subsequent phases**.

---

2. Dataset & Preprocess

2.1 IR2025 (trec-covid) Collection

- **Corpus**: **JSONL file** containing **~50 000** documents, each with fields **"_id"** and **"text"**.
- **Relevance Judgments** (*qrels*): **TSV file** with columns (*query-id, corpus-id, score*), where **score∈{0,1}** and the **average number** of **relevant docs per query is ~10**.

2.2 Text‑Analysis Pipeline

To **maximize term overlap** while **minimizing noise**, we configured a **custom analyzer** that applies:

1. **Standard Tokenization**
2. **Lowercasing**

3. **English Stop-Word Removal (*built-in _english_ list*)**
4. **Krovetz Stemming**

We selected the **Krovetz stemmer** for our custom analyzer because, as Rivas et al. (*2014*) demonstrate in their study of query-expansion techniques for biomedical information retrieval, Krovetz strikes a **better balance** than more aggressive stemmers (*e.g., Porter*) between conflating true morphological variants and preserving the core semantic integrity of terms. In their experiments, Krovetz stemming yielded **higher precision** and recall by avoiding over and under stemming, which is critical for **maintaining query-document** term overlap without introducing noise. By integrating Krovetz stemming into our **IR2025** baseline, we ensure that inflected word forms (*such as "retrieval" vs "retrieve"*) are **normalized** to a common root while **minimizing spurious conflations**, thereby improving **BM25's** ability to rank semantically relevant documents. This choice aligns with our **Phase 1** objective of establishing a **robust classical retrieval pipeline** before applying more advanced synonym-expansion methods.
This pipeline **reduces** the impact of **case differences**, function words, and **morphological variants**, thereby improving **BM25's ability** to **score semantically relevant matches**.

---

3. Index Construction

We created a fresh **Elasticsearch index** with the following characteristics:

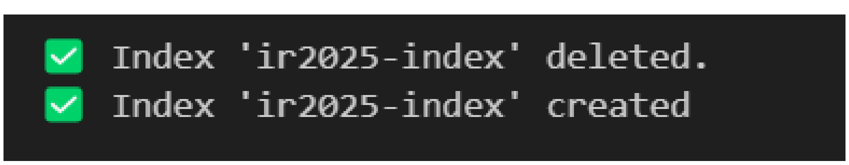
1. Similarity: **BM25**
2. Fields:

   doc_id as **keyword** for stable identifiers.

   text as **text**, processed by our **custom analyzer**.

Any existing index with the same name was **deleted** to ensure reproducibility.

Figure 1 shows the successful index-creation console output.



---

4. Document Ingestion

Documents were ingested using the **streaming_bulk** helper in chunks of 500, achieving **~5000 docs/min**. A progress bar provided real‑time feedback. Upon completion, manual spot‑checks (*e.g., querying for "pandemic"*) confirmed expected retrieval behavior.

Figure 2 displays the ingestion progress bar at **100 %**.

```
100%|██████████| 171332/171332 [00:53<00:00, 3191.38docs/s]
✅ Indexed 171332/171332 documents into 'ir2025-index'
```

---

5. Query Execution

For each test query **q**, we issued a simple full‑text match query on the text field, retrieving the **top-k** document IDs for **k** ∈ **{20, 30, 50}**. This straightforward approach isolates the impact of the analyzer and **BM25** ranking without additional heuristics.

---

6. Evaluation

6. 1 Metrics

- **Mean Average Precision (MAP)** over the full retrieved list.
- **Average Precision@k (avgPre@k)** for k = 5, 10, 15, 20.

6.2 Tools

- **pytrec_eval**: Python library for IR metrics.
- **trec_eval**: NIST reference binary for cross‑validation.

We computed per-query metrics for each run **(k = 20, 30, 50)** and then averaged over **100** queries to obtain **global MAP** and **avgPre@k** scores.

---

7. Results

	Phase 1 MAP	Phase 1 avgPre@5	Phase 1 avgPre@10	Phase 1 avgPre@15	Phase 1 avgPre@20
k					
20	0.020569	0.64	0.582	0.564	0.548
30	0.027753	0.64	0.582	0.564	0.549
50	0.039911	0.64	0.582	0.564	0.549

---

8. Analysis

1. **Mean Average Precision (MAP) Trends**
   - **MAP** nearly doubles (*from ~0.021 to ~0.040*) as the cutoff expands from 20 to 50, indicating that additional relevant documents lurk beyond the **top 20** but are ranked **below** the top positions.
2. **Early Precision Stability**
   - The invariance of **avgPre@5**, **avgPre@10**, and **avgPre@15** across all runs (*0.640, 0.582, 0.564, respectively*) reveals that the very top of each ranking is dominated by the same set of relevant documents regardless of **k**. This suggests **high confidence** in **BM25's** top results for our queries.
3. **Trade-off at Deeper Cutoff**
   - **Precision@20** shows a marginal gain (*+0.001*) when moving from **k = 20 to k = 30/50**, reflecting that **BM25's** ranking quality degrades gradually as **lower-ranked** documents enter the result set.
4. **Implications of Low MAP**
   - Although **early precision** is substantial, **MAP** values remain **low** (*<0.04*), signifying that while the first few retrieved documents are often relevant, the overall ranking precision across positions **1–k** is modest. In other words, many relevant documents are **scattered** beyond rank **1–k** or interleaved with non-relevant items.
5. **Underlying Causes**

- **Vocabulary gap**: Pure term-matching **fails** when query terms **do not** precisely **match** document **vocabulary** (*e.g., synonyms*).
- **Uniform field treatment**: A **single match** over the whole text **ignores** structural cues (*titles, abstracts*) that could **improve ranking**.

These observations confirm that **BM25** provides a **reliable head-start**, delivering **relevant** documents at the **top**. It also underscores its limitations in achieving high recall and **ranked precision** across a **broader result set**.

---

9. Conclusion

Phase 1 establishes a clear classical **IR baseline** on the **IR2025** collection using **Elasticsearch's BM25** with **text normalization**. Key findings:

- **Top-20** yields the best **early precision (*avgPre@5 = 0.640*)** but a **low overall MAP (*0.0206*)**.
- **Expanding k** to 50 **improves MAP** to **0.0399**, yet precision beyond the top few ranks drops only marginally.

This baseline **quantifies** the **strengths** and **boundaries** of pure **term-matching retrieval**. It excels at **selecting** a handful of **highly relevant documents** but **struggles** to **maintain precision across deeper ranks**. Having identified the precise performance ceiling of **BM25**, **Phases 2** and **3** will focus on **query expansion** with **synonym lookup** and **distributional semantics** to bridge the **observed vocabulary gap** and **elevate overall retrieval effectiveness**.

References

1. Elasticsearch Reference (v8.17.2) [*https://www.elastic.co/guide/en/elasticsearch/reference/8.17*](https://www.elastic.co/guide/en/elasticsearch/reference/8.17)
2. Thakur, N., Reimers, N., Rücklé, A., Srivastava, A. and Gurevych, I., 2021. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. arXiv preprint arXiv:2104.08663.

3. Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.
4. Rivas, Andreia & Iglesias, Eva & Borrajo, María. (2014). Study of Query Expansion Techniques and Their Application in the Biomedical Information Retrieval. The Scientific World Journal. 2014. 1-10. 10.1155/2014/132158.