# A Comparative Analysis of BM25, Semantic, and Hybrid Retrieval Models in Scientific Document Search

Background & Motivation

This project addresses a research-focused information retrieval (IR) problem within the domain of scientific literature search. To be specific, it explores how users such as researchers and academic professionals can more effectively retrieve relevant research documents using different retrieval techniques. Traditional keyword-based retrieval systems (like BM25), [A.1] Myscale.com. (2024), often fail to capture the semantic context of queries, which can result in suboptimal document rankings. This project aims to evaluate how well semantic search (vector embeddings via pre-trained language models) and hybrid search (combing both the BM25 paradigm & a vector approach utilising the BERT Model) perform compared to traditional lexical matching, in terms of precision, recall and MAP (Mean Average Precision). As mentioned before, the beneficiaries include the following: Academic researchers who need to retrieve relevant papers for literature reviews or knowledge discovery & Researchers based in Information Retrieval modules, can analyse the effectiveness of modern semantic techniques against lexical baselines. To evaluate the three core paradigms (I.e. BM25, Semantic & Hybrid), relevance assessments is conducted using evaluation metrics mentioned above, which are used for standard IR system benchmarking. To use relevance assessments for this project, the choice of using a pre-developed relevance assessment was chosen, which meant that the test collection needed to include these values. Therefore, the project uses the Cranfield Collection, (B.1) oussbenk (2022)., which is a well-established TREC style IR dataset for experimental retrieval research. It includes 1400 aerospace research documents in XML format, a set of 365 fixed information needs (queries) in XML format & relevance judgments mapping queries to relevant documents.

IR Application: Development Process and Technical Documentation

To successfully deploy Apache Solr for the project, a custom batch file named temp.bat, [E.10] temp.bat batch file, was created. This batch script automates the environment setup and launches Solr with standalone ZooKeeper, specifying ports and configuration options. The main challenge I faced was with Solr’s most recent cloud-mode release that no longer bundle a standalone zookeeper. Initial attempts to use the latest version (Solr 9.6.+) caused dependency issues and service boot failures when attempting to connect to Solr’s Admin UI. To solve this issue, I downloaded Solr 9.5.0, (D.2) Important Dependencies, because it still included a standalone zookeeper instance that can be launched in standalone mode. Furthermore, I used JDK-17 as previous attempts to run Solr on newer JDKs resulted in runtime errors.

To index the Cranfield TREC collection into Apache Solr, I built a schema that accommodates both traditional BM25 paradigms and modern semantic search using dense vector embeddings. The schema was defined to include the following fields: id (a unique document identifier), title, author, text (full document content), abstract (first 50 tokens of the full text) & vector (a 384-dimensional dense embedding used for semantic search). [E.1] The documents were transformed into a JSON format before being submitted to the Solr indexer. This format allows for compatibility with Solr’s endpoints. The transformation and indexing process is handled by the “collection\_updates.py” python file. The data generated in the vectors field was embedded by the, [C.1] Hugging Face (n.d.), Sentence-BERT Model. A major challenge in indexing was configuring Solr to support dense vector search. This involved the following: Ensuring that the vector field was defined as “knn\_vector” in the Solr schema with 384 dimensions. Incorrectly sized vectors or schema mismatches initially led to Solr errors such as: “incorrect vector dimension. The vector value has size X while it is expected a vector with size 384”. This was resolved by embedding the full text field using Sentence-BERT and ensuring all 384 dimensions were uploaded.

The full application implements and compares 3 primary search paradigms using Apache Solr & Python. The first paradigm implemented was BM25, [E.5] BM25 Lexical Paradigm, which constructs the query using all document fields (i.e. title, abstract, text, etc..) to enhance recall for evaluation metrics. This was then tested on a draft GUI and returned slightly meaningful results. However, an issue was raised after testing where queries beginning with the word “what..” gave incorrect or the same 10 results and could be a result of the query being too vague. The second paradigm to be implemented was the semantic paradigm, [E.6] Semantic Paradigm, where I encoded the queries into a 384-dimensional embedding and executes the K-NN search with Solr’s dense vector capabilities. The libraries used to support the BERT model is sentence-transformers, Appendix D. Testing these yielded interesting issues where vector queries produced 400 Bad Request errors and to solve this issue, I tested difference endpoints that Solr provided, [E.9] SOLR Query/Select Methods, which resulted in only using “/select” instead of “/query”. Another issue was raised by testing the semantic search where there was a vector length mismatch, where it expected 384 and led to no results being returned. This was due to the document vectors field being improperly indexed. The solution to this issue was encoding documents with the BERT Model during collection creation, this ensured that queries use the same model consistently and initially, debug statements to verify the Solr schema had the correct vector type and dimension. And finally, the Hybrid Paradigm was implemented, [E.7] Hybrid Paradigm using both BM25 & Embedded Vectors, and this was done by reranking the top 100 BM25 results using vector similarity. The main issue with the hybrid paradigm is that BM25 has a lot of influence as it’s the initial ranking, and due to time restrictions, the solution could have been running a semantic search first, then run the BM25 search which could have yielded clearer results. This means that both the BM25 and Hybrid searches will have similar results when querying.

To utilise the relevance judgements and produce evaluation metrics for paradigm analysis, [E.8 Evaluation Metrics, the function “evaluate\_results” uses both the “cran.qry.xml” file which contains structured fixed queries with unique IDs and the “cranqrel.trec.txt” file which contains relevance judgements mapping the query IDs to relevant document IDs, formatted in TREC style. Previously, tested with only the top 10 results to calculate the precision, recall and mean of precisions for each relevant document. The main issues that I came across in earlier testing was that all outputs for precision, recall & MAP were often 0 because of incorrect query ID parsing from XML which led to no matching documents. To solve this issue, ensured the “query\_id” is always extracted as a string and document IDs matched exactly.

Evaluation of IR Application

The evaluation of this IR system was carried out using a batch-style evaluation, leveraging the Cranfield TREC test collection. As mentioned in previous sections, this collection provided a framework which included documents, fixed queries and relevance judgements which removed the need to create new topics or perform manual relevance assessments. To evaluate retrieval performance across three search paradigms, I applied Precision at 50, Recall, and Mean Average Precision (MAP) which are standard IR evaluation metrics. However, due to the size of the collection, I only selected queries with the most relevant documents and marked them with a star icon, which means evaluation metrics remained low across all paradigms for vague or short queries, consistent with the limitations of lexical matching and sparse embeddings.

To help the beneficiaries (as mentioned in the background & motivation section), a graphs tab was developed contain a Score Distribution Bar chart indicating relevance score patterns across documents and an Evaluation Metrics bar graph plotting precision, recall and MAP for BM25, semantic and hybrid paradigms. Users can visually compare search effectiveness, supporting insight-driven debugging and optimization.

The main limitations are as follows:

* Zero Score Issue - Some queries still return 0.0 across all metrics due to insufficient recall or absence of relevant documents in top 50 results. This is also the result of a small document collection of only 1400 and possibly incorrect relevance judgements.
* Hybrid Paradigm Bias - The hybrid paradigm heavily depends on BM25’s initial ranking which results in the semantic search to have little influence over document relevance ranking.
* Fixed Queries - Evaluation is limited to the Cranfield collection’s fixed queries, which may not represent broader modern IR needs.

Future Work:

* Reverse the hybrid search to prioritize semantic matching first.
* Implement query expansion or reformulation using NLP techniques for vague queries.
* Conduct real-world usability testing with academic users to evaluate perceived relevance.
* Extend system support to other TREC datasets (e.g., COVID search, medical retrieval) for broader evaluation.

REFERENCES

**Appendix A: Research Papers**

(**A.1)** Myscale.com. (2024). Understanding BM25 Limitation: A Detailed Analysis. [online] Available at: <https://myscale.com/blog/bm25-limitation-comparative-analysis/>.

**(A.2)** Fucci, D., Romano, S., Baldassarre, M., Caivano, D., Scanniello, G., Thuran, B. and Juristo, N. (2022). A Longitudinal Cohort Study on the Retainment of Test-Driven Development. arXiv, [online] 1(1). Available at: https://arxiv.org/pdf/1807.02971.pdf.

**(A.3)** Robertson, S. (2010). The Probabilistic Relevance Framework: BM25 and Beyond. Foundations and Trends® in Information Retrieval, 3(4), pp.333–389. doi:https://doi.org/10.1561/1500000019.

**Appendix B: Cranfield TREC Collection**

‌**(B.1)** oussbenk (2022). GitHub - oussbenk/cranfield-trec-dataset: Cranfield collection in TREC XML format. [online] GitHub. Available at: https://github.com/oussbenk/cranfield-trec-dataset/tree/main

**‌Appendix C: Sentence Transformer BERT Model**

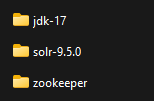
**(C.1)** Hugging Face (n.d.). sentence-transformers/all-MiniLM-L6-v2 · Hugging Face. [online] huggingface.co. Available at: https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2.

**Appendix D: Libraries and Dependencies Utilised**

**(D.1) Imported Libraries**

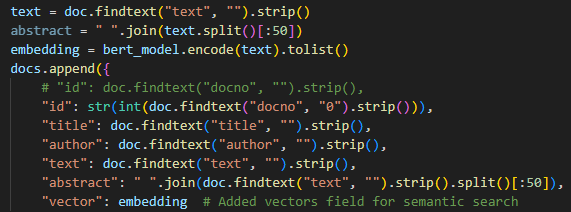
|  |  |  |
| --- | --- | --- |
| **Library Name** | **License/Legal Documentation** | **Used in** |
| PyQt5 | GPL V3 | IR\_Main.py |
| Sentence-transformers | Apache License 2.0 | Collection\_updates.py & IR\_Main.py |
| Transformers | Apache License 2.0 | Collection\_updates.py & IR\_Main.py |
| Requests | Apache License 2.0 | Collection\_updates.py & IR\_Main.py |
| matplotlib | PSF-compatible license | IR\_Main.py |
| numpy | BSD 3-Clause License | collection\_updates.py & IR\_Main.py |

**(D.2) Important Dependencies**

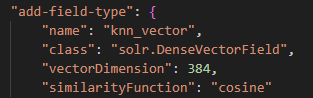


**Appendix E: Snippets of Source Code**

**(E.1) Schema Design and Data Format**



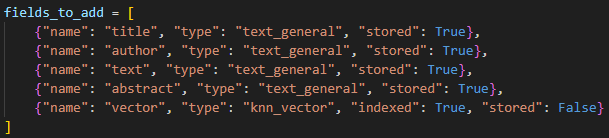
**(E.2) Vector Field and Field Type**



**(E.3) Indexing Tools**



**(E.4) Schema Creation in Solr**

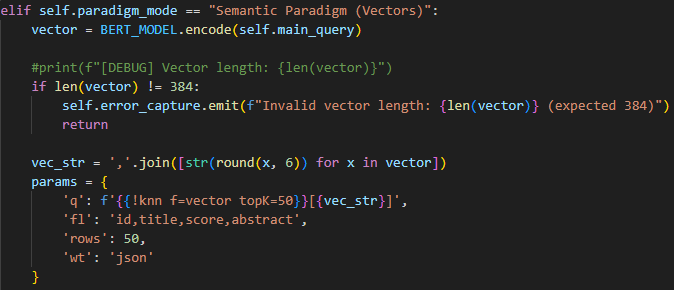
****

**(E.5) BM25 Lexical Paradigm**

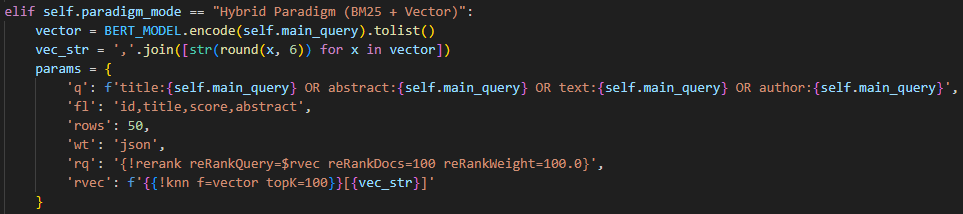
A computer screen shot of a black background

AI-generated content may be incorrect.

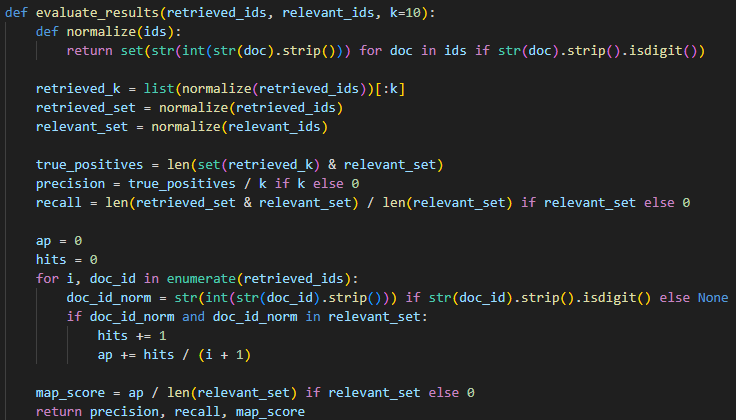
**(E.6) Semantic Paradigm (Vectors Only Utilising Bert Model)**

****

**(E.7) Hybrid Paradigm (Using both BM25 & Embedded Vectors)**

****

**(E.8) Evaluation Metrics**

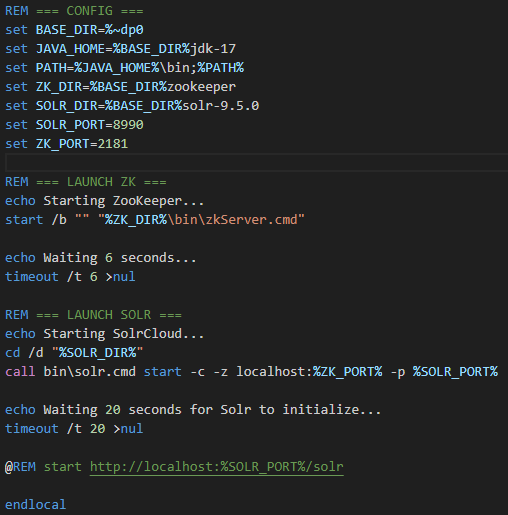
****

****

**(E.9) SOLR Query/Select Methods**

****

**(E.10) temp.bat batch file**

****