**Information Retrieval system using Cranfield Dataset.**

**Assignment - 1**

**Course code - CA6005**

**Name – Krishan D Radadia**

**Student Number: 23102528**

**Abstract.**

This report investigates how well different information retrieval systems find relevant documents. It compares three specific methods: Vector Space Model (VSM), Best Match 25 (BM25), and Okapi BM25. These methods are assessed using a large collection of Cranfield documents (1400) and corresponding questions to see which one delivers the most accurate results. Instead of simply showing which documents match the search query, the ideal method ranks the results based on their relevance. To measure this effectiveness, the report uses established metrics like Mean Average Precision (MAP), Precision at 5 (P@5), and Normalized Discounted Cumulative Gain (NDCG) for e.g. Imagine you're searching a library catalog for books. This report analyzes different search techniques to determine which one brings the most useful books to the top of your search results.

**Introduction of Models.**

*1 - The Vector Space Model (VSM) in Information Retrieval*

The Vector Space Model (VSM) is a fundamental approach to representing documents and user queries within an information retrieval system. It operates under the principle that documents with similar content will occupy closer positions in a high-dimensional space. Each dimension corresponds to a unique term within the document collection.

***Core Components:***

* ***Vector Representation*:** Documents and queries are transformed into vectors. These vectors can be visualized as arrows in space, with their direction determined by the terms they contain.
* ***Term Weighting*:** The importance of each term is assessed using two factors:
  + ***Term Frequency* (tf):** This reflects the number of times a term appears within a specific document. Higher frequencies indicate greater significance.
  + ***Document Frequency* (df):** This indicates how many documents contain the term. Terms present in many documents hold less weight in distinguishing between them.
  + ***Inverse Document Frequency* (idf):** This value adjusts the weight based on document frequency. Less frequent (more specific) terms receive a higher weighting, emphasizing their importance for retrieval.
* ***Similarity Calculation*:** The similarity between document and query vectors is computed. Documents positioned closer in space share a higher degree of topical similarity.
* ***Document Ranking*:** Based on the calculated similarity scores, documents are ranked in order of relevance to the user's query. Documents with the highest similarity scores are presented first. (Polettini, 2004)

***2 - Best Match 25 (BM25).***

The Best Match 25 (BM25) model is a powerful technique for ranking documents based on their relevance to a user's query. It achieves this by incorporating several key factors alongside computational efficiency.

***Relevance Factors:***

* ***Term Importance*:** Like the Vector Space Model, BM25 considers both the frequency of terms within a document (term frequency) and their prevalence across the document collection (inverse document frequency).
* ***Document Length*:** BM25 acknowledges that document length can influence the significance of terms. It compensates for this by adjusting the weight assigned to terms within longer documents.
* ***Query Term Matching*:** Documents containing a higher proportion of the query's terms are deemed more relevant by BM25. This logic aligns with the intuition that documents containing all the search terms are likely more pertinent to the user's needs.

***Adaptability and Efficiency:***

* ***Tunable Parameters*:** BM25 possesses parameters that can be optimized based on the characteristics of the document collection and the specific requirements of the search system. This flexibility allows the model to adapt to diverse query and document types.
* ***Document Scoring*:** BM25 assigns a relevance score to each document after evaluating the aforementioned factors. Documents with higher scores are positioned higher in the search results.
* ***Balanced Approach*:** By simultaneously considering term frequency, document length, and query term presence, BM25 offers a more balanced approach to ranking compared to simpler models. This translates to a significant improvement in the accuracy and relevance of search results.
* ***Scalability*:** While more intricate than some models, BM25 maintains a manageable level of computational complexity, making it suitable for large-scale information retrieval systems. The algorithm's time complexity scales linearly or near-linearly with the number of documents and terms, ensuring efficiency in practical applications.

In essence, BM25 acts as an intelligent filter, meticulously analyzing documents to identify the ones that best match the user's query. It achieves this by considering a multitude of relevance indicators while maintaining computational efficiency for real-world information retrieval tasks. (Zaragoza, 2009)

***3 – Okapi BM25 Language Model.***

Okapi BM25 stands as a cornerstone technique for search engines, adept at pinpointing the most relevant documents for your queries. It acts as a sophisticated assistant, meticulously evaluating documents to ensure an accurate and efficient search experience.

***Understanding Relevance*:** At the heart of Okapi BM25 lies its ability to rapidly assess a vast collection of documents and estimate their relevance to your search terms.

***Prioritizing the Best Matches*:** As the "BM" in its name suggests (Best Matching), Okapi BM25 prioritizes identifying documents that closely align with your search intent.

***Analyzing Term Importance*:** Like other retrieval models, Okapi BM25 considers the frequency of terms within documents. However, it goes a step further by placing greater emphasis on terms that appear frequently within a specific document but are uncommon across the entire document collection. Such terms hold greater significance for determining a document's relevance to your unique query.

***Adaptability through Parameters*:** Okapi BM25 incorporates a small set of parameters, like k1 and b, that can be adjusted to tailor the model's behavior to specific search scenarios. These parameters allow the model to adapt to diverse query types and document collections.

***Accounting for Document Length*:** Recognizing that longer documents might dilute the importance of certain terms, Okapi BM25 factors in document length during its evaluation process. This ensures a more balanced assessment of relevance.

***Ranking through Scoring*:** After meticulously considering all these elements, Okapi BM25 assigns a relevance score to each document. Documents with scores that best reflect their alignment with your query are positioned higher in the search results.

***Simplicity at its Core*:** Despite its technical-sounding name, Okapi BM25 operates on a straight forward principle: acting as an intelligent tool for search engines to efficiently identify the documents that hold the most value for your specific search needs. (Yaël Champclaux) (Lorenzo Bonetti, 2021)

**Architecture of the models**.

* ***Vector Space Model (VSM):***
  + The Module\_Term\_Document\_Matrix module calculates the TF-IDF weighted term-document matrix, which serves as the foundation for document ranking.
  + Cosine similarity is employed to rank documents based on their relevance to a given query.
* ***BM25*:**
  + The BM25\_ScoreCalculation function implements the core BM25 scoring algorithm. This function calculates relevance scores for each document within the collection with respect to a particular user query.
* ***Okapi BM25*:**
  + The implementation for Okapi BM25 is like BM25, with potential variations in the scoring calculation logic.

***Query Processing:***

* The Load Queries function is responsible for loading queries from a designated file.
* These loaded queries undergo preprocessing steps akin to those applied to document data.

***Ranking and Output Generation:***

* For each query, the system ranks the documents using the retrieval models (VSM, BM25, Okapi BM25).
* The Output\_File function generates an output file containing the ranked lists of documents alongside their corresponding relevance scores for each model.

***Evaluation:***

The evaluation component of the information retrieval system encountered an obstacle. While the system attempted to leverage trec\_eval (Unable to get if from google drive) to assess the performance of each model (VSM, BM25, Okapi BM25), this process resulted in an error. Consequently, the system was unable to evaluate the models based on the intended parameters: Mean Average Precision (MAP), Precision at 5 (P@5), and Normalized Discounted Cumulative Gain (NDCG).

***Additional Components:***

* The code incorporates functions for calculating vital metrics like Inverse Document Frequency (IDF) using Calculate\_InverseDocumentFrequency, Term Frequency (TF) using Calculate\_TermFrequency, and the construction of the term-document matrix within Module\_Term\_Document\_Matrix.

In essence, this information retrieval system adheres to a well-defined pipeline, effectively processing and retrieving documents based on user queries through the utilization of multiple retrieval models. The evaluation stage plays a crucial role in gauging the efficacy of each model in retrieving relevant documents.

**Ranking**

This section delves into the design choices and implementation details of the information retrieval system's search and ranking functionalities. The core objective lies in retrieving documents relevant to a user's query and subsequently ranking them based on their degree of relevance.

***Retrieval Strategies:***

* ***Vector Space Model (VSM):*** The system leverages the VSM paradigm for document retrieval. Upon receiving a query, a term-document matrix is constructed. Each document within this matrix is represented by a vector containing TF-IDF weights for all the unique terms. Cosine similarity is then employed to compute the similarity between the query vector and each document vector in the matrix. Documents exhibiting higher cosine similarity scores are deemed more relevant to the query and are consequently retrieved.
* ***BM25 and Okapi BM25:*** For these models, document relevance scores are calculated based on the query terms and their occurrences within each document. Both BM25 and Okapi BM25 incorporate term frequency (TF), inverse document frequency (IDF), and document length normalization factors into their scoring mechanisms.

***Ranking Mechanism:***

Following the retrieval of documents using the chosen model (VSM, BM25, or Okapi BM25), a ranking process is initiated to order them based on their relevance scores. This ranking determines the order in which documents are presented to the user. To achieve this, documents are sorted in descending order based on their relevance scores. This ensures that the most relevant documents are positioned at the top of the retrieved list. The ranking process itself involves iterating over the retrieved documents and employing built-in sorting functions or algorithms to arrange them according to their relevance scores.

***Data Structure Motivations:***

* ***Dictionaries*:** The system utilizes dictionaries to represent various data structures, including term-document matrices, document scores, and others. Dictionaries offer efficient retrieval of document information by leveraging keys such as document identifiers or terms.
* ***Lists*:** Lists are employed to store document rankings and other sequential data. This choice facilitates effortless sorting and manipulation of ranked documents.
* ***DefaultDict*:** To gracefully handle situations where keys might be missing within dictionaries, the system utilizes DefaultDict. This not only simplifies the code but also ensures smooth retrieval and ranking operations.

***Rationale Behind Specific Choices:***

* ***TF-IDF Weighting*:** The decision to utilize TF-IDF weighting for VSM stems from its ability to reflect the significance of terms within documents relative to the entire document collection. This weighting scheme effectively captures the semantic relevance of documents to the user's query.
* ***BM25 Parameters*:** The system allows for the configuration of parameters like k1 and b within the BM25 scoring function. These parameters influence the impact of term frequency and document length normalization on the relevance scores assigned to documents. This tunability empowers the system to adjust the ranking algorithm based on the characteristics of the specific dataset and the desired retrieval objectives.

**Evaluation**

This section explores how the information retrieval system is designed to work seamlessly with the Cranfield collection, a renowned benchmark dataset frequently used in information retrieval research.

***Data Handling:***

* ***Document Structure*:** The Cranfield collection is comprised of a set of documents, each uniquely identifiable by a document ID and containing its corresponding text content. The system retrieves these documents from a data file (e.g., "cran.all.1400" or other file names with formatted documents). Following retrieval, the documents undergo a preprocessing stage to generate an inverted index, and the processed documents are stored in a separate folder ("Preprocessed Documents").
* ***Query Creation*:** Queries relevant to the Cranfield collection are typically provided in a distinct file (e.g., "cran.qry"). The system leverages the Load\_Queries function to load these queries and subjects them to a preprocessing phase akin to that applied to documents. The preprocessed queries are then stored within the same folder as the preprocessed documents.
* ***Inverted Index and Preprocessing*:** Before constructing the inverted index matrix, both documents and queries are meticulously preprocessed. This preprocessing involves essential steps such as tokenization, stop-word removal, and stemming. These steps ensure that the text data is normalized and optimized for efficient retrieval. The preprocessed data is then stored in a dedicated folder named "Preprocessed Documents."

***Evaluation Considerations:***

The system's evaluation section strives to assess the effectiveness of the implemented information retrieval models (VSM, BM25, Okapi BM25) using established metrics like Mean Average Precision (MAP), Precision at 5 (P@5), and Normalized Discounted Cumulative Gain (NDCG). However, as mentioned previously, an error was encountered during this evaluation phase, hindering the acquisition of these results.

***Importance of Evaluation Metrics:***

Had the evaluation section functioned flawlessly, the obtained metrics (MAP, P@5, NDCG) would have provided valuable insights into the system's performance in retrieving relevant documents for the Cranfield collection queries. These metrics serve as crucial indicators of the system's effectiveness and can guide potential improvements and optimizations in future iterations.

**Conclusion, limitations, and Future Directions**

This project successfully achieved its core objective: developing a fundamental Information Retrieval (IR) system using Python. While an obstacle prevented the evaluation of the model's accuracy through metrics like MAP and P@5, the exploration of three prominent retrieval models (VSM, BM25, Okapi BM25) yielded valuable insights. This exploration provided a deeper understanding of each model's strengths and weaknesses in document ranking and retrieval tasks.

Despite achieving partial objectives, the project presents exciting avenues for further improvement, encompassing both the overall design and code implementation. Here are some potential areas for future exploration:

* ***Enhancing Query Understanding*:** Implement techniques like query expansion to improve retrieval effectiveness. This could involve incorporating additional relevant terms into the original query using user feedback or external resources like WordNet, a lexical database.
* ***Advanced Text Processing*:** Experiment with advanced pre-processing techniques such as part-of-speech tagging, named entity recognition, or spelling correction. These techniques can refine the quality of indexed data and potentially improve retrieval accuracy.
* ***Modeling Language Relationships*:** Explore language modeling techniques like Latent Semantic Analysis (LSA) or Latent Dirichlet Allocation (LDA) to understand the semantic connections between terms and documents. This could lead to more effective retrieval performance.
* ***User Feedback Integration*:** Incorporate a relevance feedback mechanism. This would allow users to provide feedback on retrieved documents, enabling the system to adapt its retrieval strategy based on user preferences, ultimately leading to the retrieval of more relevant documents for subsequent queries.
* ***Result Diversity*:** Implement strategies to ensure retrieved documents cover a broader range of relevant topics, rather than focusing solely on the most relevant ones. Techniques like Maximal Marginal Relevance (MMR) can be explored for this purpose.
* ***Scalability Enhancements*:** Utilize parallel processing techniques to distribute indexing and retrieval tasks across multiple cores or machines. This can significantly improve system scalability and performance when dealing with large document collections.
* ***Data Structure Optimization*:** Optimize the data structures and algorithms used for indexing and retrieval. This could involve considering factors like memory usage, indexing speed, and retrieval efficiency. Experimenting with compressed inverted index representations can potentially reduce memory overhead.
* ***Streamlined Evaluation*:** Integrate established evaluation frameworks (e.g., pytrec\_eval) directly into the system. This would streamline the evaluation process and ensure accurate measurement of retrieval performance metrics.

By exploring these potential improvements, the system can evolve into a more robust and effective information retrieval system, capable of delivering superior user experience.

# **References**

Lorenzo Bonetti, P. P. (2021). DESIGN AND IMPLEMENTATION OF A REALWORLD SEARCH ENGINE BASED ON OKAPI BM25 AND SENTENCEBERT. *IEEE*.

Polettini, N. (2004). The Vector Space Model in Information Retrieval - Term Weighting Problem. *IEEE*.

Yaël Champclaux, T. D. (n.d.). ENHANCING HIGH PRECISION BY COMBINING OKAPI BM25 WITH STRUCTURAL SIMILARITY IN AN INFORMATION RETRIEVAL SYSTEM. *IEEE*.

Zaragoza, S. R. (2009). The Probabilistic Relevance Framework: BM25 and Beyond. pp. 333-389.