A Holistic View of Effective Document Retrieval

Rifaa Qadri **Joseph Hong** **Andrew Zheng** **Xinchen Yang** **Vrundal Shah**

1 Introduction

Information retrieval is the process of searching a corpus and returning relevant documents and information for a given query. In the realm of Natural Language Processing (NLP), tasks such as question answering, text summarizing, fact checking, machine translation among others, rely heavily on practical and efficient retrieval mechanisms to retrieve relevant documents pertaining to a user's query Thakur et al. 2021. Traditionally, lexical approaches like TF-IDF and BM25 have dominated textual information retrieval Robertson et al. 1995. Recent efforts to use neural networks to improve or replace these lexical approaches aim to learn a high-dimensional representations of text to capture semantic relationships between words and documents. While much of the current research concentrates on fixed-length document representations, our investigation delves into understanding the impact of factors such as chunking, overlap, and dimensionality reduction, on retrieval effectiveness, particularly for queries of varying lengths.

Our findings reveal that computational-intensive models do not consistently translate into improved retrieval performance. Further, our work show that it is possible to shrink our index size without harming retrieval accuracy. Through thoughtful preprocessing of documents, accounting for the intrinsic nature of the data and the writing style of the author, we demonstrate that retrieval can be enhanced in challenging retrieval scenarios.

2 Related Work

Several studies have evaluated various information retrieval systems, including lexical, sparse, dense, late-interaction, and re-ranking architectures. BEIR's evaluation of 10 state-of-the-art models highlights the continued robustness of BM25 while emphasizing the effectiveness of re-ranking and late-interaction approaches in achieving strong zero-shot performance Thakur et al. 2021. Additionally, the Massive Text Embedding Benchmark (MTEB) assessed the performance of 33 embedding models across 8 tasks and 58 datasets, demonstrating that no single method dominates across all domains and tasks Muennighoff et al. 2023.

Lexical retrieval approaches like BM25 utilize bag-of-words retrieval functions based on token-matching between query and document vectors weighted by TF-IDF scores. State-of-art methods use mutli-stage techniques such as retrieve-and-re-rank architectures that first employ a lexical retriever to maximize recall, followed by a neural ranker to refine the candidate set. Hitachi recently demonstrated a multi-stage re-ranking model, where training miniLM on MS-MARCO was shown to perform best using as the high performance LM during the third stage Sasazawa et al. 2023. Further, dual-encoder architectures leverage separate encoders for queries and documents, mapping them to dense representations in a shared vector space. Retrieval occurs via k-nearest-neighbor search, identifying documents with representations most similar to the query.

These results provide valuable context for our work on chunking, overlap, and dimensionality reduction for diverse query lengths. We aim to leverage these insights to improve retrieval performance and domain robustness, contributing to the ongoing advancement of neural IR research.

3 Data Acquisition and Processing

We begin with a collection of highly correlated Wikipedia documents covering selected topics. Examples include Chinese Dynasties such as the Han, Ming, Qin dynasties, and sorting algorithms like bubble, selection and heap sort. This accounts for our biggest corpus, at 1.6 megabytes. For our second dataset covering Bio-Medical IR, we first utilize the National Library of Medicine to extract "human-preferred terms" and retrieve a subset of documents from the TREC_COVID dataset containing the terms (2K). Examples of documents retrieved include those with titles like "flu", "influenza", "asthma", "dyspnea", "respiratory hypersensitivity", "hypertension", "cardiovascular", "alzheimer" and "parkinson". The same extraction method is used for the NFCorpus corpus (203K sized PubMed Articles). We assess document similarity using cosine similarity, BM25, and Jaccard similarity measures. Some of the results can be seen in the appendix.

The text is tokenized and processed with varying chunk sizes ranging from 100 to 500 words and overlap sizes ranging from 0.05 to 0.5. Dimensionality reduction techniques are further applied to reduce the size of the learned document representations. We test retrieval against queries of varying length (25-50 vs 50-100).

4 Retrieval Methods

4.1 Traditional Methods

BM25 (Best Match 25) tries to solve is similar to that of TFIDF (Term Frequency, Inverse Document Frequency), that is representing our text in a vector space. It improves upon TFIDF by casting relevance as a probability problem. A relevance score, according to probabilistic information retrieval, ought to reflect the probability a user will consider the result relevant.

$$\text{BM25}(D, Q) = \langle f_\theta(q), f_\theta(d) \rangle$$

where $f_\theta(q)$ and $f_\theta(d)$ represent the embeddings of the query Q and document D respectively.

4.2 Pre-Trained Models

By leveraging the power of pre-trained language models, we encode queries and documents in a shared semantic space This allows for retrieval based on semantic similarity, even if the query terms haven't appeared in the training data. For this study, we focus on the following models: Sentence-Transformers/all-MiniLM-L6-v2 utilizes MiniLM, a Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers Wang et al. 2020. Muennighoff/SGPT-125M-weightedmean-nli-bitfit leverages SGPT, a sparse gated permutation transformer model, for efficient and generalizable text retrieval. Facebook/Contriever employs a two-tower bi-encoder trained with contrastive InfoNCE loss and BM25 hard negatives. Despite their architectural differences, all models rely on the dot product between query and document representations in the metric space for calculating relevance scores:

$$s(q, d) =< f_\theta(q), f_\theta(d) >$$

where q is the query, d is the document, $f_\theta(q)$ and $f_\theta(d)$ are the encoded representations of the query and document, respectively, and θ represents the model parameters.

5 Results

5.1 Wikipedia

We start our experiments with extracting a subset from the Wikipedia dataset as described above. Figure 1 shows the similarity of a subset of documents across the corpus. The average document length is around 38996. Average number of tokens per document after tokenization is around 8476.

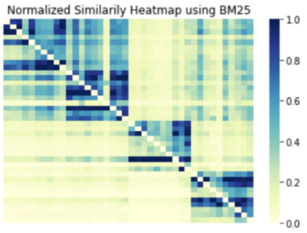

Figure 1: Similarity of Wikipedia articles using BM25

We can notice that with this dataset, retrieval is best when the data has been processed into smaller chunks. Figure 2 shows the effect of increasing chunk size on the retrieval accuracy. As we can see, a chunk size of approximately 200-400 performs the best, while trying to optimize storage. Furthermore, miniLM performs the best across all settings, with an average accuracy of 75%.

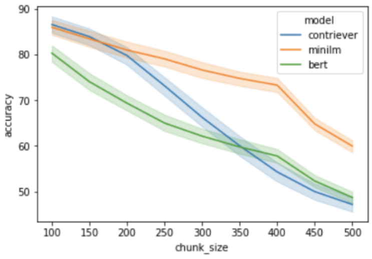

Figure 2: Effect of chunk size across models

Moreover, we notice that while preprocessing our data in smaller chunks is beneficial, representing this data in a higher dimension helps in retrieval accuracy, but not tremendously. As seen in figure 3, we notice an insignifcant difference between the results achieved with dimension 200 and 300.

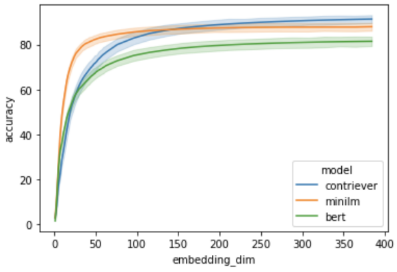

Figure 3: Dimension Effect across models

The chunking has a greater impact on the retrieval than dimension representation. Yet, as seen in figure 4 with chunk sizes ranging between 200 to 400, retrieval performs best with the higher dimension such as the 364 dimensional embedding.

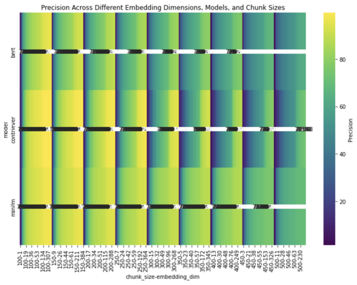

Figure 4: Chunk size&dimension effect across models

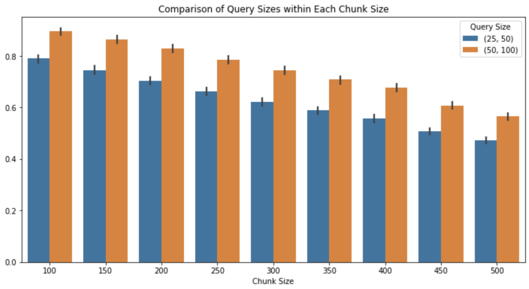

Figure 5: Effect of query length

Figure 5 demonstrates the impact of the query length of retrieval across the different chunking settings. A surprising finding is how longer queries seem to be better for retrieval in this dataset. A reason for this can be due to the fact that these articles are longer and more comprehensive, and therefore longer queries might be more effecting in capturing the information present in the lengthy documents.

5.2 TREC-COVID

We filter documents with similar topics including 'flu', 'influenza', 'asthma', 'respiratory hypersensitivity', 'hypertension'... The similarity of a subset of the documents in this corpus is seen in figure 6. Documents in this dataset have on average length of 1192. After tokenization, 254 tokens are extracted on average.

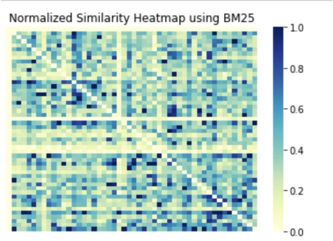

Figure 6: Similarity of TREC_COVID articles using BM25

We observe a significant correlation between chunk size and performance, as seen in figure 7, with optimal precision achieved when the chunk size falls within the range of 400-500.

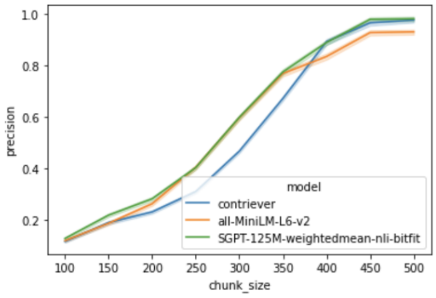

Figure 7: Effect of chunk size across models

While we notice close similarity in performance, the SGPT model slightly outperforms the others in this dataset, with an average accuracy of 67%. Moreover, a higher dimensional representation seems to help. This effect is yet again less prominent here, as we notice a greater impact from the chunking choice. For instance, we see that we can achieve high retrieval accuracy with a much lower dimensionality representation for chunking choices of 400 and above, as seen in figure 8.

Finally, we note that queries of shorter length seem to perform generally better within this dataset,

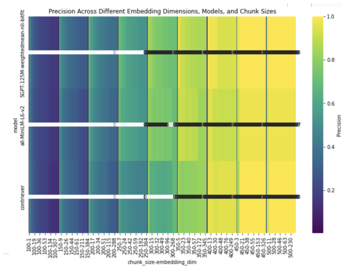

Figure 8: Chunk size&dimension effect across models

with an observed tighter gap as the number of chunk size increased, as seen in figure 9.

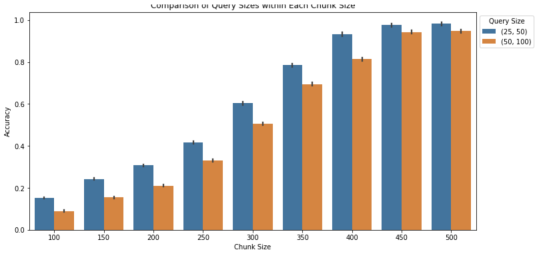

Figure 9: Effect of query length

5.3 NFCorpus

The average text length is 1659 in our extracted subset of this dataset. After tokenization, there is an average of 364 tokens per document. The similarity of a subset of the documents in this corpus is seen in figure 10.

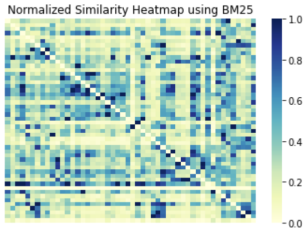

Figure 10: Similarity of NFCorpus articles using BM25

We notice that a chunk size of 300 is sufficient to achieve our desired accuracy. As seen in figure 11, miniLM performs really well across all chunking settings. Moreover, there is less benefit in the higher dimensional representation of texts in this dataset. In fact, we notice that the computationally extensive techniques do not help us in retrieval. As seen in figure 12, MiniLM is able to outperform others across all chunking, overlap and dimensionality reduction, with the average accuracy of 97.6%. Such results makes this model stand out from all others in both speed and performance.

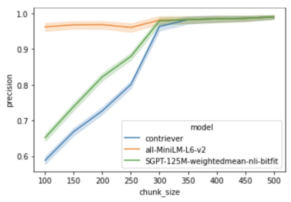

Figure 11: Effect of chunk size across models

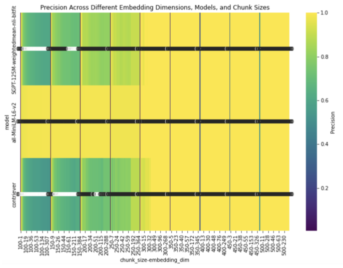

Figure 12: Chunk size&dimension effect across models

With less of an impact on retrieval, we notice that the smaller length queries generally perform better when the chunking size is less than 300, but becomes irrelevant beyond that, as seen in figure 13.

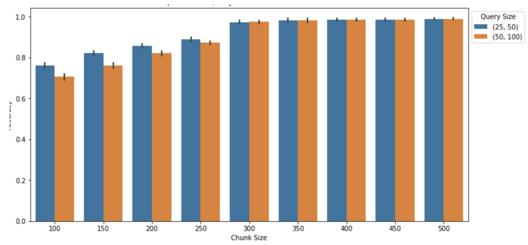

Figure 13: Effect of query length

6 Conclusion

In this study, we conducted a comprehensive evaluation of text segmentation techniques, specifically focusing on the impact of chunking, overlap, and dimensionality reduction on information retrieval within the context of the Wikipedia, TREC_COVID, and NFcorpus datasets, using BM25, SGPT, MiniLM, and Contriever. Our findings highlight the nuanced relationship between these factors and retrieval effectiveness, emphasizing that computational intensity alone does not guarantee optimal performance. Through careful consideration of chunking strategies and overlap mitigation, as well as dimensionality reduction techniques, we have demonstrated the importance of tailored preprocessing methods in improving retrieval outcomes. When performing work in information retrieval, taking these factors, including article lengths, chunking and embedding into account depending on its domain during processing stage can help the system tremendously.

7 Future Work

Several promising directions emerge from our current research. Investigating the performance of unsupervised tokenizers, such as SentencePiece employing byte-pair-encoding (BPE) and unigram language models, could offer insights into more flexible and adaptive tokenization strategies. Exploring how these techniques handle diverse domains and contribute to improved retrieval outcomes remains an intriguing area.

The creation of a diverse dataset by combining data from different domains using models like Latent Dirichlet Allocation (LDA) presents an exciting avenue. Exploring the integration of topic probability into document representations offers a pathway to richer and more contextually nuanced document embeddings. This can enhance the semantic understanding of documents and contribute to improved retrieval precision. Investigating the potential for semantic interpretation of queries based on underlying topics could lead to more sophisticated query expansion techniques. This exploration aligns with the goal of refining information retrieval systems to better understand and respond to user intent. The identified future directions offer exciting opportunities to refine and extend our understanding of text representation and retrieval, ultimately contributing to the ongoing evolution of information retrieval systems.

References

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2023. Mteb: Massive text embedding benchmark.

Stephen Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford. 1995. Okapi at trec-3. In *TREC*, pages 109–126, Gaithersburg, MD. NIST.

Yuichi Sasazawa, Kenichi Yokote, Osamu Imaichi, and Yasuhiro Sogawa. 2023. Text retrieval with multi-stage re-ranking models.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers.

A Appendix

A.1 Jaccard Similarities

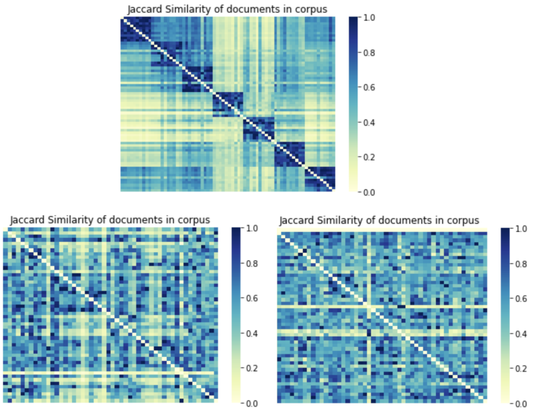

Figure 14: Jaccard Similarity matrices of corpus

A.2 Data Visualization Graphs

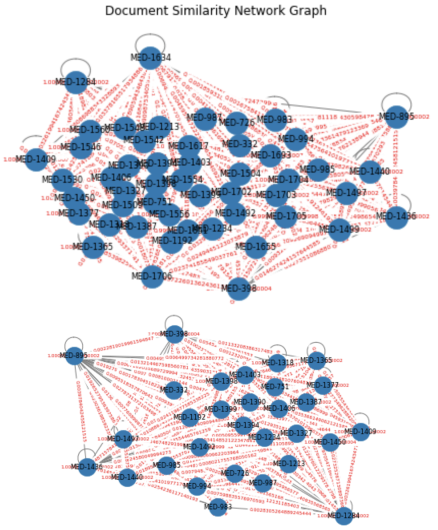

Figure 15: Network Data Representation of TREC_COVID and NFCorpus respectively

A.3 Next steps with LDA

An example to demonstrate the potential of training the unsupervised LDA model to be helpful in different stages in IR (as described above). The following retrieves the top 10 topics from our Wikipedia corpus.

The following demonstrates the results for the top 2 words:

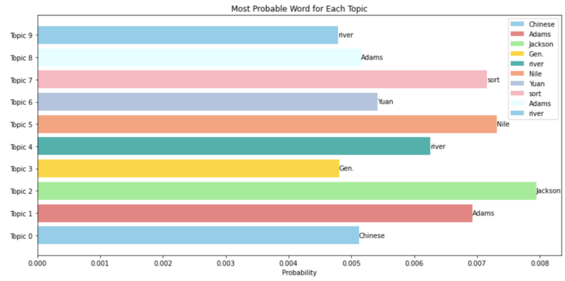

Figure 16: Top 1 word for each topic

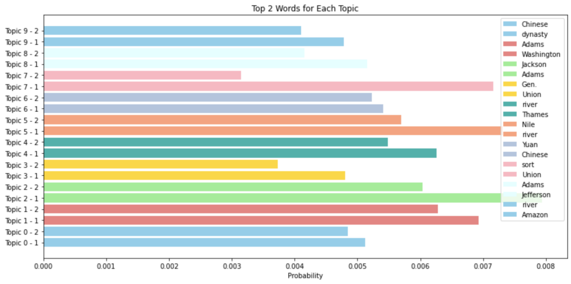

Figure 17: Top 2 words for each topic