CODEX RAG — RAG-BASED MODEL DEVELOPMENT FOR EFFICIENT PROCESSING OF ICRC AND WHO'S DYNAMIC ARCHIVES AND GUIDELINES

Camille Challier
MSc semester project report
09/2024
EPFL
camille.challier@epfl.ch

Mary-Anne Hartley & Martin Jaggi & Eduard Ďurech
Supervision
Intelligent Global Health Research Group
Machine Learning and Optimization Laboratory
EPFL
mary-anne.hartley@epfl.ch
martin.jaggi@epfl.ch
eduard.durech@epfl.ch

ABSTRACT

BACKGROUND. The International Committee of the Red Cross (ICRC) and the World Health Organization (WHO) manage vast archives and guidelines. Processing these documents is challenging due to their size, complexity, and constant update. Existing Large Language Models (LLMs) require extensive computational resources which are often inaccessible to these organizations.

AIM. This study aims to develop a scalable Retrieval-Augmented Generation (RAG) framework to automate the processing, retrieval, and summarization of large document collections for the ICRC and WHO, minimizing the need for extensive computational resources.

METHODS/FINDINGS. A new chunking strategy was developed and applied to two datasets: one provided by the ICRC, consisting of 200 multilingual documents of varying quality, and another consisting of 1,000 high-quality English Wikipedia articles from the WikiText dataset. Various retrieval and embedding techniques were tested across both datasets. For the ICRC dataset, the RAG model achieved a final Hit Rate of 0.742, a Mean Reciprocal Rank of 0.688, a Faithfulness score of 0.91, and a Relevancy score of 0.88. On the WikiText dataset, the model also delivered strong results, with a Hit Rate of 0.986, a Mean Reciprocal Rank of 0.943, a Faithfulness score of 0.89, and a Relevancy score of 0.98. The model's performance on multilingual data was significantly improved by leveraging a vector database and applying fine-tuning techniques.

CONCLUSION. The proposed RAG model offers an initial exploration of an efficient and scalable solution for managing large multilingual document archives. It addresses document processing challenges while minimizing the need for extensive computational resources. [1]

[1]The code, primarily utilizing the LlamaIndex library, has been made publicly available at https://github.com/LiGHT-YalEPFL/Codex/tree/RAG.

1 Background

The ICRC manages the world's largest humanitarian archives, with records spanning over 160 years. These extensive archives, while invaluable, present significant challenges in terms of accessibility and efficient data retrieval, necessitating automation for reading and parsing such vast amounts of information. Similarly, the WHO, the largest health organization in the world, produces 20 to 30 major guidelines annually. These guidelines, which set global standards in healthcare, require extensive review of academic papers and resources. The growing volume and complexity of these documents create an increasing need for AI-assisted tools to streamline the review and guideline development process.

While high-accuracy language models exist, which have the capability to process and summarize large datasets, they often require substantial computational resources [1]. ICRC and WHO do not have access to such high-end computational infrastructure. Additionally, much of the sensitive data they handle must remain local and cannot be uploaded to remote servers for fine-tuning state-of-the-art (SOTA) models, further complicating the adoption of such technologies. Moreover, most SOTA models are closed source, limiting the flexibility to adapt them for specific needs. The continuous update of documents, the need for accurate citation of sources, and the risk of generating hallucinated data also pose significant challenges.

To address these challenges, a scalable Retrieval-Augmented Generation (RAG) model is proposed [2]. A RAG system works by retrieving documents that semantically match a query and then utilizing a LLM to extract the correct answer from the retrieved documents. The goals of RAG systems are to: minimize the issue of hallucinated responses from LLMs, provide sources or references for generated answers, and eliminate the need for manual document annotation with metadata. By leveraging a small LLM with RAG, this approach aims to preprocess documents efficiently and embed them into an offline database. This solution would enable the ICRC and WHO to automate the processing of their vast and continuously updated archives and guidelines, ensuring accurate and efficient data management without heavy computational demands.

2 Aim and Objectives

This project aims to develop a scalable RAG model to efficiently manage and accurately process the extensive, continuously updated archives and guidelines of the ICRC and WHO. To achieve this, the following objectives are pursued:

1. **Objective 1.** To review the literature on RAG models and their applications (Section 3)
2. **Objective 2.** Build a vector database by processing archives and guideline data (Section 5.1).
3. **Objective 3.** Accurately retrieve the most relevant documents from the database (Section 5.2).
4. **Objective 4.** Effectively re-rank the retrieved documents to maximize their relevance (Section 5.3).
5. **Objective 5.** Integrate a summarization module with a compact LLM for improved document processing (Section 5.4).

3 Related Works

3.1 RAG Methods on Medical and Non-Governmental Organization (NGO) Documents

Recent research on vector databases (VectorRAG) highlights their role in enhancing Natural Language Processing (NLP) by retrieving relevant textual information to support generation tasks [2, 3]. VectorRAG is particularly effective when context from related documents is crucial for generating coherent responses. However, retrieving relevant texts from multiple documents or longer contexts remains challenging. For NGO documents, which often include domain-specific language and varied data formats, traditional RAG systems with paragraph-level chunking may miss critical context [4]. This limitation can lead to inconsistent and incomplete quality of the LLM retrieved-context from a vast and heterogeneous corpus. In the medical domain, RAG technology is still evolving. The MEDRAG system evaluated different retrievers and corpora for medical question-answering [5]. The Self-BioRAG project integrated Self-RAG technology to assess document relevance and quality in answer generation [6, 7]. To enhance performance, it is essential to incorporate domain-specific knowledge during the fine-tuning phase, which ensures that the model accurately recognizes and utilizes relevant domain-specific information [8].

Knowledge graphs (KGs) [9] offer an alternative by representing documents as triplets of entities and relationships. KGs are valuable for structured data management and analysis, widely used in search engines, recommendation systems, and biomedical research [10, 11]. Despite their advantages, constructing and maintaining KGs and integrating diverse data sources presents challenges. For example, entity disambiguation can be difficult, KGs can be slow at retrieval, and handling data changes is complex. Additionally, loss of context and poor data quality can impact their accuracy and reliability [12].

This paper reviews and tests various approaches for retrieval-augmented text generation tasks to address these challenges.

3.2 RAG for Multilingual Corpi

RAG has recently shown promise in enhancing LLMs with domain-specific knowledge. However, most studies have focused primarily on English. Chirkova et al. extended this work to a multilingual context (mRAG), incorporating user queries and data stores in 13 languages [13]. Their study reveals the need for stronger multilingual LLMs and decoding strategies. They found that even the most advanced multilingual LLMs struggle with mixed-language contexts and often require ad-hoc prompting to maintain consistency in user languages. The study suggests that incorporating mixed-language examples in instruction tuning or developing specific decoding strategies could mitigate these issues. Additionally, the research points out that current multilingual retrievers and re-rankers,

primarily trained on Wikipedia-based data, may have limited applicability to other domains, indicating a need for multi-domain multilingual retrieval approaches.

3.3 RAG Limitations

RAG systems face limitations, both from traditional information retrieval systems' constraints and their dependence on LLMs. Barnett et al. highlight that expanding context and using metadata improves retrieval accuracy [14]. However, excessive noise or conflicting information can hinder the model's ability to find the correct answer. This paper tackles these challenges by reviewing and testing chunking methods for retrieval-augmented text generation tasks.

3.4 Contributions

This paper makes several key contributions to the field of RAG and its application to diverse domains:

- **Systematic Evaluation of Chunking Methods**: Different chunking methods and their impact on critical downstream processes, such as embedding and similarity matching, are systematically evaluated. This approach addresses the need for a comprehensive evaluation framework that compares chunking techniques, as highlighted in Barnett et al. [14]. By identifying the strengths and limitations of various chunking strategies, valuable guidance for improving retrieval accuracy and context relevance is provided.
- **Comparison of Retrieval Techniques for NGO and Medical Documents**: Various retrieval techniques are systematically compared to evaluate their effectiveness in handling NGO and medical documents, which often feature domain-specific language and diverse data formats. This comparative analysis provides insights into which methods are most effective for retrieving and generating accurate responses from these specialized corpora.
- **Adaptation of Retrieval Methods for Multilingual Data**: Existing retrieval methods are adapted to better handle multilingual documents. Building on recent advancements in multilingual RAG, this work focuses on tailoring established techniques to address the challenges posed by mixed-language contexts and domain-specific data. The adaptation is aimed at enhancing the effectiveness of retrieval and generation across multiple languages.

4 Methods

To address the challenge of developing a model capable of answering multilingual NGO questions, a dedicated RAG pipeline was implemented. Initially, data was extracted from PDF documents using Optical Character Recognition (OCR). The extracted data was then segmented into manageable chunks to facilitate efficient processing. Embeddings were generated for each chunk to enable similarity matching during the retrieval phase. After retrieval, a re-ranker was applied to prioritize the most relevant documents, ensuring higher-quality results. Finally, the ranked information was passed through a generation model to produce accurate and context-aware responses. Each of these steps, from data extraction to response generation, is illustrated comprehensively in Figure 1.

4.1 Data

	Wikitext Dataset	ICRC Dataset
Quantity	1293 articles	193 documents
Quality	Extracted from the set of verified Good and Featured articles on Wikipedia.	Extracted using Marker and Surya for OCR.
Languages	English	French, English and German
Advantages	High-quality data	Subset of the real data
Disadvantages	Monolingual and not Domain-Specific	Poor Quality

Table 1: Overview of Datasets. This table summarizes the WikiText dataset of high-quality Wikipedia articles and the ICRC dataset of multilingual documents, used for optimizing the RAG system.

WikiText dataset This initial dataset comprises 1,000 high-quality English Wikipedia articles [15].

ICRC dataset The second dataset is provided by the ICRC and consists of 200 multilingual documents, some of which are lengthy and were extracted using Marker and Surya [2]. Marker is a pipeline of deep learning models. It works by extracting text (using OCR when necessary with heuristics, Surya, and Tesseract), detecting page layout and reading order, cleaning and formatting each block (using heuristics and Texify), and finally combining blocks and post-processing the complete text (using heuristics and pdf postprocessor). The ICRC dataset is a subset of the actual data employed by the Red Cross. However, due to the age of certain documents, the quality of the extracted data may be inconsistent, with some sections being unreadable.

In contrast, the WikiText dataset consistently provides high-quality data. The combination of these datasets allows for the development of a RAG system that will be optimized for the final ICRC dataset.

4.2 RAG Model architecture

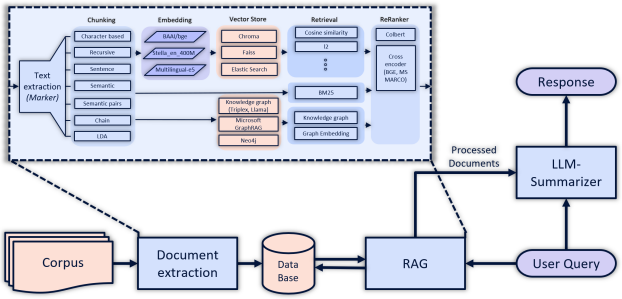

Figure 1: Schema illustrating the overall RAG system pipeline. The diagram visualizes the steps and models tested and used from the initial data to the final model.

[2] https://github.com/VikParuchuri/marker

4.2.1 Chunking

After extracting the data, it is segmented into smaller, more manageable pieces through a process known as chunking. This technique involves dividing large texts into chunks, facilitating efficient indexing and retrieval while preventing information overload in LLMs. Chunking plays a crucial role in optimizing data processing by ensuring that text is organized into appropriately sized portions for downstream tasks. Various chunking methods are available, including **character-based** chunking, which involves dividing the text into fixed-length segments, and **sentence** splitting, which breaks the text at sentence boundaries while considering maximum length constraints. Additionally, **recursive character** splitting is employed to define chunks by primarily leveraging paragraph breaks and other delimiters [16]. This method is guided by a prioritized list of characters, where the splitting occurs progressively according to this hierarchy until the chunks reach the desired size. The default character list includes, in order, line skip, newline, and space. This ensures that paragraphs are preserved first, followed by sentences, and then words. This approach aims to maintain the semantic relationships within the text by keeping these meaningful units intact as long as possible.

Unlike traditional chunking methods that rely on physical boundaries such as characters or sentences, **semantic chunking** organizes text based on its meaning and context [17]. This method allows for a more coherent grouping of related information, avoiding arbitrary divisions. A key parameter in this approach is the buffer size, which determines the initial window for chunk formation. A buffer size of 2 was found to perform best and was therefore used in subsequent experiments. A new technique introduced in this study is **Semantic Pairs** chunking, which enhances semantic chunking by restructuring sentence groupings based on pairs of preceding and following sentences. Furthermore, a '**Chaining**' approach was tested, grouping sentences sequentially by comparing each sentence with the next based on the cosine similarity of their embeddings. If the similarity exceeds a predefined threshold, the sentences are merged and compared to the following one. If the similarity falls below the threshold, the sentences are treated as distinct, preserving semantic boundaries. The *w601sxs/b1ade–embed* [18] model was employed for all the semantic chunking techniques. Additionally, other models were tested, including *intfloat/multilingual–e5–base* [19], *Lajavaness/bilingual–embedding–large* [20, 21], and *dunzhang/Stella_en_400M_v5* [22]. Performance details for these models can be found in Figure 18 of the Appendix. To further improve chunk coherence, topic modeling using **Latent Dirichlet Allocation** (LDA) was also tested [23]. Each tokenized sentence was converted into a bag of words, and the LDA model from the Gensim library was used to obtain the topic probabilities for each sentence, facilitating more accurate semantic grouping. More details on the mechanism of these techniques can be found in Figures 13 and 14 of Appendix.

4.2.2 Retriever

In the first phase of the retrieval process, various embedding models were tested. The principle behind this approach is based on a **Vector Store Index**, where embeddings are stored and the top-k most similar chunks are fetched for each query. Various embedding models were tested, including *BAAI/bge–small–en* [24], *dunzhang/stella_en_400M_v5* [22], and *intfloat/multilingual–e5–large* [19]. **Faiss** was employed for retrieval using k-means clustering [25, 26]. This method segments the data into clusters, narrowing the search space by focusing on the most relevant clusters for each query. After clustering, quantization and processing methods are applied, followed by similarity scoring to retrieve the best documents. **ChromaDB** was also tested, leveraging an approximate nearest neighbor (ANN) search algorithm for efficient retrieval.

Following the embedding-based retrievers, the **Best Matching 25 (BM25)** algorithm was evaluated. BM25 ranks documents based on the occurrence of query terms and their rarity within the corpus. The BM25 score for a document is computed using the following formula:

$$BM25 = \sum_{t \in q} \log\left(\frac{N}{df(t)}\right) \cdot \frac{(k_1+1)\cdot tf(t,d)}{k_1\left[(1-b)+b\cdot\frac{dl(d)}{dl_{avg}}\right]+tf(t,d)}$$

with k_1, b – parameters ; $dl(d)$ – length of document d ; dl_{avg} – average document length ; tf - term Frequency ; df - document Frequency. Additionally, **Elastic Search** was evaluated for comparison.

The **Knowledge Graph** method was tested using an LLM to extract sets of triplets from text [27, 28]. A triplet consists of two objects and their relationship. Two models were used to extract triplets: Llama 3 8B, and Sciphi Triplex, a finetuned version of Phi3-3.8B, which provides a less computationally expensive alternative [29]. Additionally, Microsoft GraphRag [30] and Neo4jGraphStore [31] were evaluated for graph-based retrieval and knowledge management.

4.2.3 Re-Ranker

The re-ranker serves as a second-pass document filter, aiming to reorder the documents retrieved by the initial retriever to improve relevance and ranking accuracy. In this study, the retriever selects the top 10 documents, and the re-ranker refines this to the 3 most relevant. Four re-ranker models were evaluated. ColBERT v2 [32] [33], utilizes a BERT-based architecture with late interaction to enhance retrieval by considering deeper contextual relationships within the text. The remaining models, BGE Base, BGE Large [24], and MS MARCO MiniLM [34], are cross-encoders. BGE models are designed for multilingual language understanding, while MS MARCO MiniLM is limited to English only [35].

4.2.4 Summarizer head

Llama3-8B: Building on the foundation laid by Llama and Llama2, this latest model integrates more advanced algorithms and utilizes a larger dataset, enabling it to perform at a higher level across a wide range of language tasks. [36]

GPT-2: a 1.5 billion-parameter transformer model, pretrained with self-supervised learning on a large English corpus. [37]

Mistral-7B: This decoder-only model excels in text generation and comprehension and is optimized for low-resource environments. [38]

4.3 Evaluation method

4.3.1 Precise queries

The RAG model was tested using a series of questions generated by Llama 3 8B, based on 500 randomly selected data chunks. The model's effectiveness in retrieving the specific document sections used to generate the queries is evaluated using two key metrics: Hit Rate (HR) and Mean Reciprocal Rank (MRR). LLM prompt can be found in Figure 2 of the Appendix. The **HR** is calculated as HR $= \frac{|U^L_{\text{hit}}|}{|U_{\text{all}}|}$ where $|U^L_{\text{hit}}|$ represents the number of questions for which the correct answer is included in the top L retrieved chunks, and $|U_{\text{all}}|$ is the total number of questions. The **MRR** is given by MRR $= \frac{1}{|U_{\text{all}}|} \sum_{u=1}^{|U_{\text{hit}}|} \frac{1}{\text{rank}_u}$ where rank_u is the rank of the correctly retrieved chunk among the top L retrieved chunks for each question. MRR not only assesses correctness but also assigns higher scores to correctly retrieved documents that are ranked more favorably. Furthermore, each step of the pipeline is evaluated individually by systematically changing one parameter at a time. This approach allows for a detailed analysis of how each component contributes to the overall performance of the RAG model.

4.3.2 Broad queries

A second evaluation pipeline assesses the summarization models' relevancy and faithfulness for broad questions with multiple potential answers. The process starts with Latent Dirichlet Allocation (LDA) using the Gensim library to identify key topics, which are then employed to generate broad, open-ended questions with the Llama 3 8B model. Relevant documents are retrieved to provide context, summarized by a generative model, and evaluated for **relevance** and **faithfulness** [39]. Relevance checks if the response aligns with the query's intent and stays on topic. Faithfulness ensures that the response accurately reflects the source material, maintaining consistency and preventing errors. This evaluation pipeline offers a structured approach to testing both the retrieval mechanisms and the summarization for broad and precise queries.

4.4 Baselines

To assess RAG performance, several baseline models are used for comparison. For chunking, a lower baseline involves fixed-length character chunking, while an upper baseline utilizes an LLM to split the text by theme. Retrieval baselines include traditional methods like BM25, as well as other RAG models with varying configurations. Comparing each RAG component model's results with these baselines serves as a reference to gauge their effectiveness and limitations.

5 Results

5.1 Chunking Strategies

The test HR of the different character-based chunking strategies on the Wikitext dataset can be seen in Figure 2. Recursive splitting is effective for Wikipedia articles, as each paragraph generally addresses a distinct sub-topic. This approach attains an HR of 0.928 and an MRR of 0.853 with a maximum chunk size of 1024 and an overlap of 20 tokens. However, in other datasets, subjects are not always separated by paragraphs. In such cases, alternative techniques, such as semantic chunking, which do not rely on paragraph breaks, are necessary.

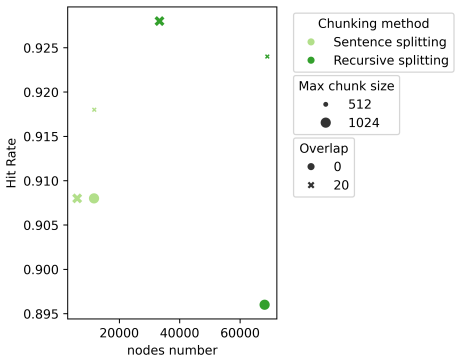

Figure 2: Hit Rate obtained using different chunking methods. The embedding model "bge-small-en" was used to retrieve the top 3 documents. It was tested on the WikiText datasets from Hugging Face "Salesforce/wikitext". The test queries consist of 500 precise questions.

When applied to the Wikitext dataset, semantic chunking revealed limitations, such as isolated sentences and the occasional generation of overly large chunks. Additionally, it struggled to reliably separate two distinct articles. To address these issues, the technique was refined by modifying the process for grouping sentences and adding a max and min length, improving the separation of distinct articles, as explained in 4.2.1. Figure 3 shows that the Chaining approach is the best-performing embedding-based technique for WikiText data, achieving an HR of 0.924 and an MRR of 0.890, while the Semantic Pairs approach performs best for the ICRC dataset as shown in Figure 4. These two splitting techniques are used for the rest of the experiment for each dataset.

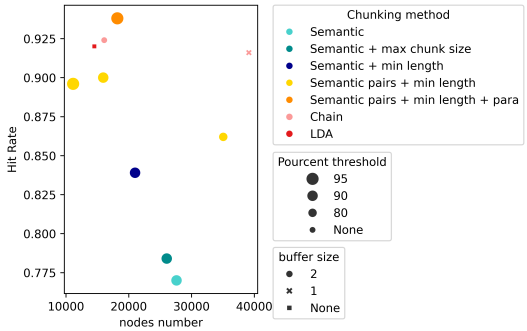

Figure 3: Hit Rate obtained using different semantic and topics-based chunking methods.The embedding model "bge-small-en" was used to retrieve the top 3 documents and tested on the WikiText datasets. The top point in orange is achieved by first separating the text by paragraphs, followed by applying semantic splitting to handle chunks that are too large. This point represents the target score, serving as a benchmark for comparison. The objective is to identify a method capable of segmenting the text at paragraph boundaries without relying on them.

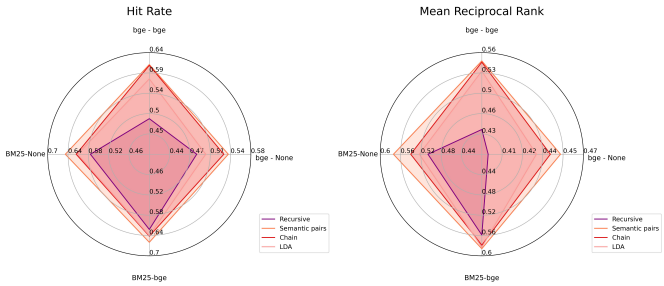

Figure 4: Hit Rate and MRR obtained using different chunking methods on the ICRC dataset. The test queries consist of 500 precise questions. The notation model refers to the embedding model name used to retrieve the top 3 documents, followed by the re-ranker name (bge refers to bge-small-en).

Another chunking technique explored was proposition splitting proposed by Chen et al. [40]. This method decomposes text into propositions by repeating the subject, treating each as an individual chunk. Initially, this approach was implemented using the Llama 3 8B model. However, it proved computationally intensive due to the large number of resulting chunks and the need to pre-split the text, as the model could not process the entire text simultaneously. To address this issue, the Flan-T5 model, specifically trained by Chen et al. [40] for text splitting, was tested. Unfortunately, this led to a decline in performance, as individual sentences lacked sufficient context when processed in isolation, and the results were limited to English data. In response to these challenges, an alternative approach was employed: first, sentences were rephrased to repeat the subject, and then semantic chunking was applied. This aimed to improve chunking by maintaining contextual coherence between sentences. Although this method yielded a slight improvement, the gains were not significant enough to justify the additional computational costs. The result of this technique is shown in Figure 16 in the Appendix. Additionally, rephrasing sentences was tested after chunking to enhance data quality without altering the chunks themselves. However, this post-chunking rephrasing led to a minor drop in performance.

5.2 Retrieval

Among the embedding models tested, *Stella_en_400M_v5* performed best on both the Wikitext and ICRC datasets, as shown in Figure 5. Stella performs competitively on the English, multilingual, and French HuggingFace MTEB leaderboards [41]. This model was trained using the multi-lingual models *gte-large-en-v1.5* and *gte-Qwen2-1.5B-instruct* of *Alibaba-NLP*.

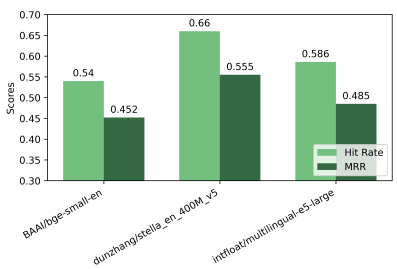

Figure 5: Performance of the RAG model on the ICRC dataset using various embedding models. Similarity search is conducted with the Faiss VectorStore. The test set comprises 500 queries generated from documents by the Llama 3 8B model.

The similarity search method, utilizing the Faiss VectorStore approach, delivered the best performance, as shown in Figure 6. It achieved an HR of 0.66 and an MRR of 0.555. This method outperformed other retrieval techniques by identifying and ranking the most relevant chunks for each query.

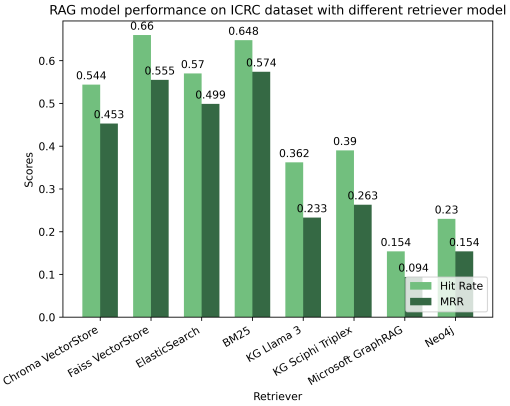

Figure 6: Performance of the RAG system on the ICRC dataset with various retriever models. All VectorStore methods utilized the embedding model `Stella_en_400M_v5` and were tested on their ability to retrieve the top 3 documents for 500 queries.

Comparing the top models across languages (Figures 7 and 8) shows significantly lower performance in French than in English, with prior translation improving the results. The models were not tested in German or Spanish due to the limited amount of available text, which made the data insufficient for a meaningful analysis.

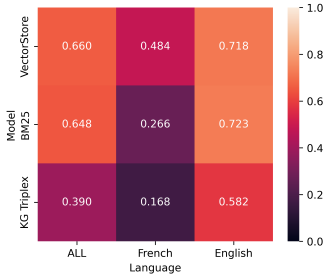

Figure 7: Hit Rate performance of Faiss VectorStore with *Stella_en_400M_v5*, KG Triplex, and BM25 models on the ICRC dataset across different languages.

Several methods were used to enhance multilingual performance such as fine-tuning the embedding model with query-document pairs and topic-specific fine-tuning with theme-document pairs (See Figure 3 of the Appendix for the prompting template). A third method involved fine-tuning a linear adapter or neural network (NN) on top of the embeddings, allowing the transformation of embedding representations into a new latent space optimized for retrieval based on specific data and queries. Additionally, a fine-tuned XGBoost model was tested to retrieve themes from the embeddings, followed by further fine-tuning of the embeddings based on these themes. As shown in Figure 9, both the fine-tuning and adapter approaches yielded improved results. Finetuning improved French performance by 14% and overall performance by 6%. Each technique was trained on 1000 queries based on 70% of the data chunks and tested on 500 queries based on 30% of the chunks.

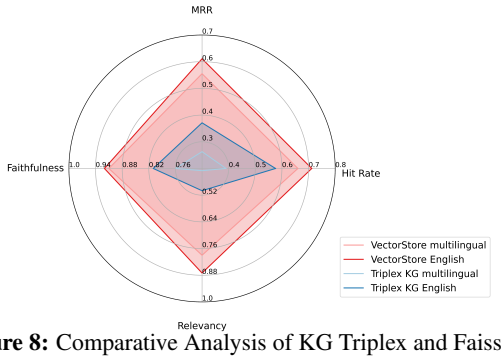

Figure 8: Comparative Analysis of KG Triplex and Faiss VectorStore using the *Stella_en_400M_v5* model, tested with and without English translation on the ICRC dataset. The translation utilized the T5 model, fine-tuned on the OPUS Books dataset [42]. Performance is measured by HR and MRR for 500 precise queries and Faithfulness and Relevance for 500 broad queries. Retrieval was tested for the top 3 documents.

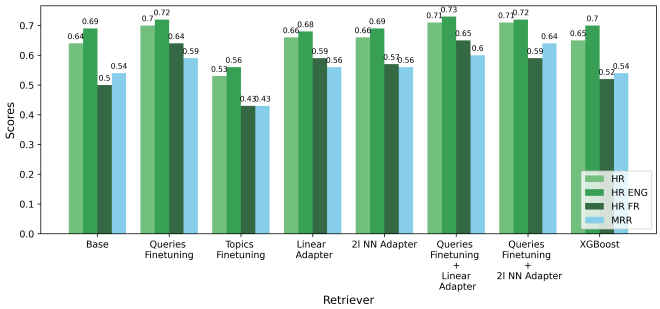

Figure 9: RAG model performance on the multilingual ICRC dataset with different improvements to the retriever model. The evaluation used a Semantic Pairs chunking strategy and Faiss VectorStore retrieval, employing the `Stella_en_400M_v5` model. The test set consists of 500 precise queries. Retrieval was tested for the top 3 documents.

The hyperparameters used for these experiments are detailed in Figure 20 of the Appendix. Due to computational limitations, a combination of 2 and 4 epochs, batch sizes of 4 and 10, and learning rates of 0.01 and 2e-5 were tested.

5.3 Re-Ranking

BGE-large from the Beijing Academy of Artificial Intelligence [43] was established as the most effective re-ranker, demonstrating superior performance in refining search results, as illustrated in Figure 10. It increased the model's HR by 8% (from 0.66 to 0.74) and the MRR by 9% (from 0.56 to 0.65).

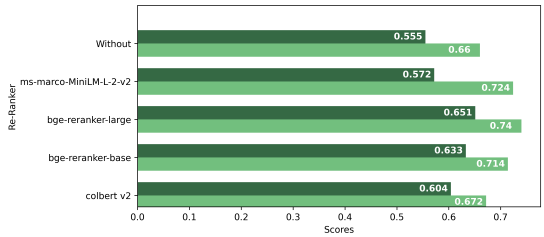

Figure 10: Performance of the RAG model on the ICRC dataset with various re-ranker models. The evaluation used a Semantic Pairs chunking strategy and Faiss VectorStore retrieval, employing the *Stella_en_400M_v5* model. The test set consists of 500 precise queries.

5.4 Summarizer head

The RAG model achieves a final HR of 0.742 and MRR of 0.688 on the test set. As observed in Figure 11, the top-performing summarizer head is Llama 3 8B, with a Faithfulness score of 0.93 and a Relevance score of 0.98 for precise questions. For broader questions, it achieves Faithfulness and Relevance scores of 0.91 and 0.88, respectively.

	HRR	MRR	
Precise queries	0.742	0.668	
	Summarizer head	Faithfulness	Relevancy
Precise queries	Llama 3 8b	0.93	0.98
	Mistral 7b	0.84	0.97
	GPT2	0.32	0.24
Broad queries	Llama 3 8b	0.91	0.88
	Mistral 7b	0.89	0.90
	GPT2	0.22	0.17

Figure 11: The first part of the table presents the Hit Rate and Mean Reciprocal Rank performance of the final model on the ICRC dataset. The second part details the RAG model's Faithfulness and Relevancy scores when using different summarizer heads on the same dataset. The evaluation employed a semantic-pairs chunking strategy; Faiss VectorStore for retrieval, using the fine-tuned *Stella_en_400M_v5* model with a 2-layer neural network adapter; and the bge-large model as a re-ranker. The test queries consisted of 500 precise and 500 broad questions. The retriever selects the top 10 documents, and the re-ranker re-fines this to the 3 most relevant.

6 Discussion

6.1 Analysis

During the selection stage of **chunking techniques**, slight differences in Hit Rate and MRR were observed between Chaining and Semantic Pairs, both of which performed well (Figures 3 and 4). The LDA-based chunking also showed strong performance with lower computational demands.

Figures 6 and 5 and 12 demonstrate that selecting an appropriate re-ranker and embedding model leads to a significant improvement in performance.

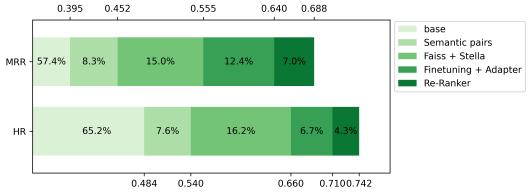

Figure 12: Improvements made at each stage of constructing the RAG model. The "base" model employs simple recursive chunking with retrieval using *bge-small-en*. The percentages reflect the level of improvement achieved at each step. The top axis represents MRR performance, while the bottom axis shows HR.

In the **Retrieval** stage, similarity search focuses on finding information most similar to the query, which can hinder answering questions with multiple topics or where less-similar information is relevant. KGs effectively capture relationships within data but are computationally expensive to construct and retrieve, and they struggle with multilingual documents

(Appendix Figure 21). For example, a KG with Triplex only achieves 27% correct French answers, compared to 48% with VectorStore (Figure 7). While translating documents boosts KG performance, it still lags behind BM25 and similarity search. Fine-tuning the embedding model or using adapters significantly reduces the language performance gap, improving performance by capturing nuanced linguistic and domain-specific features.

6.2 Limitations

One key limitation of this study is the quality of the data extracted using Marker. Moreover, the test set may not fully capture the range and complexity of questions that ICRC staff typically encounter. Consequently, the evaluation pipeline might lack real-world relevance, as it does not incorporate actual queries and documents from ICRC workers. Incorporating real questions and documents would provide a more accurate evaluation of Retrieval-Augmented Generation (RAG) combined with LLMs. Additionally, the resource intensity of this approach poses challenges, particularly with the computational demands of knowledge graph generation and LLM inference when handling large datasets. These factors may limit the practical deployment of these techniques in real-world ICRC applications.

6.3 Future work

Several areas for future work have been identified to further enhance the performance and applicability of the methods explored in this study. First, implementing a document selection mechanism before applying Retrieval-Augmented Generation (RAG) could significantly improve the relevance of retrieval when a query targets specific documents, enabling more efficient and focused retrieval, especially when targeted document sets are predefined.

Additionally, testing additional metrics is crucial for a more precise evaluation of model performance. The Normalized Discounted Cumulative Gain (NDCG), a ranking quality metric, measures how closely a model's rankings match the ideal order with all relevant items at the top. The best-performing model on the ICRC dataset has an NDCG of 0.312 and 0.448 for Wikitext. Applying NDCG and other metrics across models would offer deeper insights into retrieval quality and relevance. A combination of metrics ensures a more comprehensive and accurate performance evaluation, particularly for tasks requiring high precision and contextual understanding.

Fine-tuning models on domain-specific data is crucial for enhancing retrieval and generation for NGO tasks. Customizing models like the re-ranker and Triplex for KG creation can improve relevancy and accuracy. If fine-tuning Triplex proves effective, combining embeddings with knowledge graphs as hybrid retrievers, as outlined in Sarmah et al.[44], could offer a robust solution. This approach addresses traditional RAG limitations, such as semantic misalignment and issues with cosine similarity, which may favor irrelevant or overly brief documents. Graph-based RAG can aggregate information across documents to overcome these challenges.

Another direction for future work involves testing different summarization heads and fine-tuning them to improve response quality. Evaluating models such as Llama 3 8B, 70b, and other SOTA LLMs for their ability to generate concise, accurate summaries will provide insights into which models are best suited for summarization tasks within the ICRC context. Fine-tuning these models on ICRC-specific data will further enhance their relevancy and accuracy in producing summaries that meet the organization's needs.

Another potential direction is exploring Reinforcement Learning from Human Feedback (RLHF) to refine the RAG system and the LLM head. RLHF could help capture the nuanced requirements of humanitarian queries more effectively, enhancing the system's ability to provide relevant and contextual responses.

Finally, a comparison with commercially available models such as GPT-4 for summarization tasks and Ada embeddings for retrieval would be valuable. This would establish a strong baseline for assessing future fine-tuned models, such as using Llama 3 8B for the summarization head or embedding-based retrievers. Such comparisons would offer a clearer understanding of the strengths and limitations of each approach. By pursuing these improvements, future work can deliver more efficient, accurate, and contextually relevant tools for ICRC staff, ultimately enhancing the retrieval and generation capabilities for real-world humanitarian tasks.

6.4 Conclusion

In this study, we conducted a thorough evaluation of chunking techniques, retrieval methods, and reranking strategies in RAG for NGO and medical documents. Character-based chunking performed well when subjects were separated by paragraphs but struggled with identifying subjects within the text, while Semantic Pairing and Chaining chunking methods yielded better results. The LDA-based chunking also demonstrated strong performance, offering a balance between accuracy and computational efficiency. Fine-tuning the embedding model or using adapters helped bridge the language performance gap, improving French retrieval accuracy by up to 14% and overall performance by up to 6%. Additionally, employing the right re-ranker and embedding model led to substantial gains in retrieval effectiveness. On the ICRC dataset, the best RAG model achieved a final HR of 0.742, an MMR of 0.688, a Faithfulness score of 0.91, and a Relevancy score of 0.88. Similarly, on the WikiText dataset, the model recorded an HR of 0.986, an MMR of 0.943, a Faithfulness score of 0.89, and a Relevancy score of 0.98.

In conclusion, this study presents a comprehensive evaluation of chunking techniques, retrieval methods, and reranking strategies in RAG for NGO and medical documents. It emphasizes the importance of effective chunking and embedding models for improved retrieval, especially in multilingual and domain-specific contexts. While this study highlights the value of domain-specific fine-tuning, there is still room to improve performance on multilingual data. Several avenues remain for enhancing RAG's effectiveness in specialized fields, including humanitarian operations.

Acknowledgments

This paper and the research behind it would not have been possible without the support of my supervisor, Eduard Durech. I also express my gratitude to Professors Mary-Anne Hartley and Martin Jaggi for their insightful comments and support. Special thanks are due to the ICRC staff who provided the dataset, as their contributions were crucial to the success of this research. Their collective expertise has significantly enhanced this study in countless ways.

References

[1] A. J. Thirunavukarasu, D. S. J. Ting, K. Elangovan, L. Gutierrez, T. F. Tan, and D. S. W. Ting, "Large language models in medicine," *Nature Medicine*, vol. 29, no. 8, pp. 1930–1940, Aug. 2023, publisher: Nature Publishing Group. [Online]. Available: https://www.nature.com/articles/s41591-023-02448-8

[2] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in *Advances in Neural Information Processing Systems*, vol. 33. Curran Associates, Inc., 2020, pp. 9459–9474. [Online]. Available: https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html

[3] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M.-W. Chang, "REALM: Retrieval-Augmented Language Model Pre-Training," Feb. 2020. [Online]. Available: https://arxiv.org/abs/2002.08909v1

[4] J. Wu, J. Zhu, and Y. Qi, "Medical Graph RAG: Towards Safe Medical Large Language Model via Graph Retrieval-Augmented Generation," Aug. 2024, arXiv:2408.04187 [cs]. [Online]. Available: http://arxiv.org/abs/2408.04187

[5] G. Xiong, Q. Jin, Z. Lu, and A. Zhang, "Benchmarking Retrieval-Augmented Generation for Medicine," Feb. 2024, arXiv:2402.13178 [cs]. [Online]. Available: http://arxiv.org/abs/2402.13178

[6] M. Jeong, J. Sohn, M. Sung, and J. Kang, "Improving medical reasoning through retrieval and self-reflection with retrieval-augmented large language models," *Bioinformatics*, vol. 40, no. Supplement_1, pp. i119–i129, Jun. 2024. [Online]. Available: https://academic.oup.com/bioinformatics/article/40/Supplement_1/i119/7700892

[7] A. Asai, Z. Wu, Y. Wang, A. Sil, and H. Hajishirzi, "Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection," Oct. 2023, arXiv:2310.11511 [cs]. [Online]. Available: http://arxiv.org/abs/2310.11511

[8] T. Zhang, S. G. Patil, N. Jain, S. Shen, M. Zaharia, I. Stoica, and J. E. Gonzalez, "RAFT: Adapting Language Model to Domain Specific RAG," Jun. 2024, arXiv:2403.10131 [cs]. [Online]. Available: http://arxiv.org/abs/2403.10131

[9] S. Ji, S. Pan, E. Cambria, P. Marttinen, and P. S. Yu, "A Survey on Knowledge Graphs: Representation, Acquisition and Applications," Feb. 2020. [Online]. Available: https://arxiv.org/abs/2002.00388v4

[10] H. Paulheim, "Knowledge graph refinement: A survey of approaches and evaluation methods," *Semantic Web*, vol. 8, no. 3, pp. 489–508, Dec. 2016. [Online]. Available: https://www.medra.org/servlet/aliasResolver?alias=iospress&doi=10.3233/SW-160218

[11] L. Ehrlinger and W. Wöß, "Towards a Definition of Knowledge Graphs," Sep. 2016.

[12] S. Xu, M. Chen, and S. Chen, "Enhancing Retrieval-Augmented Generation Models with Knowledge Graphs: Innovative Practices Through a Dual-Pathway Approach," in *Advanced Intelligent Computing Technology and Applications*, D.-S. Huang, Z. Si, and W. Chen, Eds. Singapore: Springer Nature, 2024, pp. 398–409.

[13] N. Chirkova, D. Rau, H. Déjean, T. Formal, S. Clinchant, and V. Nikoulina, "Retrieval-augmented generation in multilingual settings," Jul. 2024, arXiv:2407.01463 [cs]. [Online]. Available: http://arxiv.org/abs/2407.01463

[14] S. Barnett, S. Kurniawan, S. Thudumu, Z. Brannelly, and M. Abdelrazek, "Seven Failure Points When Engineering a Retrieval Augmented Generation System," in *Proceedings of the IEEE/ACM 3rd International Conference on AI Engineering - Software Engineering for AI*. Lisbon Portugal: ACM, Apr. 2024, pp. 194–199. [Online]. Available: https://dl.acm.org/doi/10.1145/3644815.3644945

[15] "Salesforce/wikitext · Datasets at Hugging Face," Apr. 2024. [Online]. Available: https://huggingface.co/datasets/Salesforce/wikitext

[16] B. Sarmah, T. Zhu, D. Mehta, and S. Pasquali, "Towards reducing hallucination in extracting information from financial reports using Large Language Models," Oct. 2023, arXiv:2310.10760 [cs, q-fin, stat]. [Online]. Available: http://arxiv.org/abs/2310.10760

[17] "RetrievalTutorials/tutorials/LevelsOfTextSplitting at main · FullStackRetrieval-com/RetrievalTutorials." [Online]. Available: https://github.com/FullStackRetrieval-com/RetrievalTutorials/tree/main/tutorials/LevelsOfTextSplitting

[18] "w601sxs/b1ade-embed-kd · Hugging Face," May 2024. [Online]. Available: https://huggingface.co/w601sxs/b1ade-embed-kd

[19] L. Wang, N. Yang, X. Huang, L. Yang, R. Majumder, and F. Wei, "Multilingual e5 text embeddings: A technical report," *arXiv preprint arXiv:2402.05672*, 2024.

[20] N. Thakur, N. Reimers, J. Daxenberger, and I. Gurevych, "Augmented sbert: Data augmentation method for improving bi-encoders for pairwise sentence scoring tasks," *arXiv e-prints*, pp. arXiv–2010, 2020.

[21] "Lajavaness/bilingual-embedding-large · Hugging Face." [Online]. Available: https://huggingface.co/Lajavaness/bilingual-embedding-large

[22] "dunzhang/stella_en_400m_v5 · Hugging Face." [Online]. Available: https://huggingface.co/dunzhang/stella_en_400M_v5

[23] D. Blei, A. Ng, and M. Jordan, "Latent Dirichlet Allocation," vol. 3, Jan. 2001, pp. 601–608.

[24] S. Xiao, Z. Liu, P. Zhang, and N. Muennighoff, "C-pack: Packaged resources to advance general chinese embedding," 2023.

[25] "facebookresearch/faiss," Sep. 2024, original-date: 2017-02-07T16:07:05Z. [Online]. Available: https://github.com/facebookresearch/faiss

[26] M. Douze, A. Guzhva, C. Deng, J. Johnson, G. Szilvasy, P.-E. Mazaré, M. Lomeli, L. Hosseini, and H. Jégou, "The Faiss library," Jan. 2024. [Online]. Available: https://arxiv.org/abs/2401.08281v1

[27] X. Zou, "A Survey on Application of Knowledge Graph," *Journal of Physics: Conference Series*, vol. 1487, no. 1, p. 012016, Mar. 2020, publisher: IOP Publishing. [Online]. Available: https://dx.doi.org/10.1088/1742-6596/1487/1/012016

[28] C. Peng, F. Xia, M. Naseriparsa, and F. Osborne, "Knowledge Graphs: Opportunities and Challenges," *Artificial Intelligence Review*, vol. 56, no. 11, pp. 13 071–13 102, Nov. 2023. [Online]. Available: https://doi.org/10.1007/s10462-023-10465-9

[29] S. Pimpalgaonkar, N. Tremelling, and O. Colegrove, "Triplex: a sota llm for knowledge graph construction," 2024. [Online]. Available: https://huggingface.co/sciphi/triplex

[30] B. Potts, "GraphRAG: A new approach for discovery using complex information," Feb. 2024. [Online]. Available: https://www.microsoft.com/en-us/research/blog/graphrag-unlocking-llm-discovery-on-narrative-private-data/

[31] "Neo4j graph database," 2022. [Online]. Available: https://microsoft.github.io/graphrag/

[32] O. Khattab and M. Zaharia, "ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '20. New York, NY, USA: Association for Computing Machinery, 2020, pp. 39–48. [Online]. Available: https://dl.acm.org/doi/10.1145/3397271.3401075

[33] R. Jha, B. Wang, M. Günther, G. Mastrapas, S. Sturua, I. Mohr, A. Koukounas, M. K. Akram, N. Wang, and H. Xiao, "Jina-ColBERT-v2: A General-Purpose Multilingual Late Interaction Retriever," Aug. 2024. [Online]. Available: https://arxiv.org/abs/2408.16672v3

[34] "cross-encoder/ms-marco-MiniLM-L-12-v2 · Hugging Face." [Online]. Available: https://huggingface.co/cross-encoder/ms-marco-MiniLM-L-12-v2

[35] L. Bonifacio, V. Jeronymo, H. Q. Abonizio, I. Campiotti, M. Fadaee, R. Lotufo, and R. Nogueira, "mMARCO: A Multilingual Version of the MS MARCO Passage Ranking Dataset," Aug. 2021. [Online]. Available: https://arxiv.org/abs/2108.13897v5

[36] AI@Meta, "Llama 3 model card," 2024. [Online]. Available: https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md

[37] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," 2019.

[38] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. d. l. Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. L. Scao, T. Lavril, T. Wang, T. Lacroix, and W. E. Sayed, "Mistral 7B," Oct. 2023, arXiv:2310.06825 [cs]. [Online]. Available: http://arxiv.org/abs/2310.06825

[39] H. Yu, A. Gan, K. Zhang, S. Tong, Q. Liu, and Z. Liu, "Evaluation of Retrieval-Augmented Generation: A Survey," Jul. 2024, arXiv:2405.07437 [cs]. [Online]. Available: http://arxiv.org/abs/2405.07437

[40] T. Chen, H. Wang, S. Chen, W. Yu, K. Ma, X. Zhao, H. Zhang, and D. Yu, "Dense X Retrieval: What Retrieval Granularity Should We Use?" Dec. 2023. [Online]. Available: https://arxiv.org/abs/2312.06648v2

[41] N. Muennighoff, N. Tazi, L. Magne, and N. Reimers, "Mteb: Massive text embedding benchmark," *arXiv preprint arXiv:2210.07316*, 2022. [Online]. Available: https://arxiv.org/abs/2210.07316

[42] J. Tiedemann, "Parallel data, tools and interfaces in OPUS," in *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, N. Calzolari, K. Choukri, T. Declerck, M. U. Doğan, B. Maegaard, J. Mariani, A. Moreno, J. Odijk, and S. Piperidis, Eds. Istanbul, Turkey: European Language Resources Association (ELRA), May 2012, pp. 2214–2218. [Online]. Available: http://www.lrec-conf.org/proceedings/lrec2012/pdf/463_Paper.pdf

[43] "BAAI (Beijing Academy of Artificial Intelligence)," Jul. 2024. [Online]. Available: https://huggingface.co/BAAI

[44] B. Sarmah, B. Hall, R. Rao, S. Patel, S. Pasquali, and D. Mehta, "HybridRAG: Integrating Knowledge Graphs and Vector Retrieval Augmented Generation for Efficient Information Extraction," Aug. 2024, arXiv:2408.04948 [cs, q-fin, stat]. [Online]. Available: http://arxiv.org/abs/2408.04948

Appendix

Precise queries prompt	Broad queries prompt
""" Context information is below. --------- *context_str* --------- Given the context information and not prior knowledge, generate only questions based on the below query. You are a Professor. Your task is to setup *num_questions_per_chunk* questions for an upcoming quiz/examination. The questions should be developped and diverse in nature across the document. The questions should not contain options, not start with Q1/ Q2. Restrict the questions to the context information provided. """	""" Context information about the International Committee of the Red Cross (ICRC) and their interventions is below. --------- *context_str* --------- You are a Teacher/ Professor. Your task is to setup *num_question* questions for an upcoming quiz/examination. Given the context information and not prior knowledge, generate only questions based on the previous keyword themes. """

Table 2: Prompting template used to generate queries for both evaluation pipelines.

Topic extraction prompting template
""" You are an AI assistant specialized in classifying PDF documents into broad categories. Your task is to analyze the given text and provide a single tag that best describes the content or purpose of the document. Use the most appropriate tag from the following list, or create a new tag if necessary: education, government, science, health, legal, environment, finance, social-sciences, history, human-rights, community-development, humanitarian-aid, advocacy, sustainability, migration-and-refugees, social-justice, crisis, war. Respond with only the tag, nothing else. Here are some examples: 1. "February 7, 1984 lebanon : the icrc calls for immediate ceasefire. Since the deterioration of the situation in lebanon over the last few days, the civilian population has sustained hundreds of victims, dead and wounded, particularly in the south of the capital, beirut." Humanitarian Aid 2. "M. comelio sommaruga, président du comité international de la croix-rouge (cicr) donnera une conférence de presse le lundi 8 février 1988 à 10 heures 30 m. sommaruga présentera le bilan des principales actions conduites en 1987 ainsi que les objectifs et les grandes options de l'institution pour l'année en cours." Advocacy Now, please classify the following text: *context_str* """

Table 3: Prompting template used to generate queries topics used for training XGboost and finetuning based on topics.

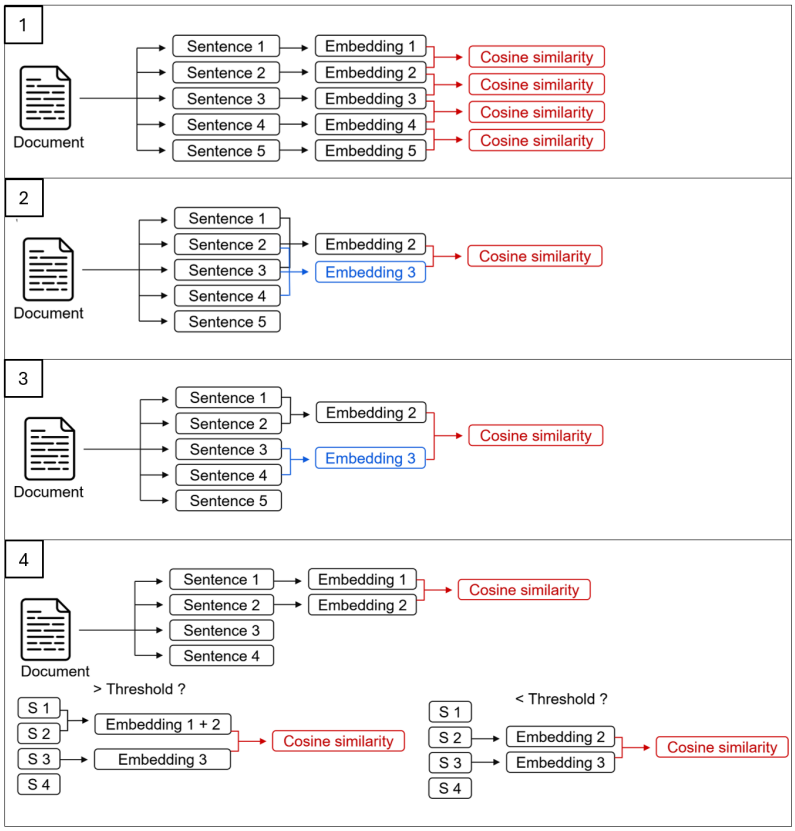

Figure 13: Overview of the semantic chunking methods used: 1, 2 – Original semantic technique without and with a buffer size of 2, respectively. 3 – Modified method using Semantic Pairing. 4 – Chaining technique for chunking with no buffer. A buffer of size 2 was used in the result.

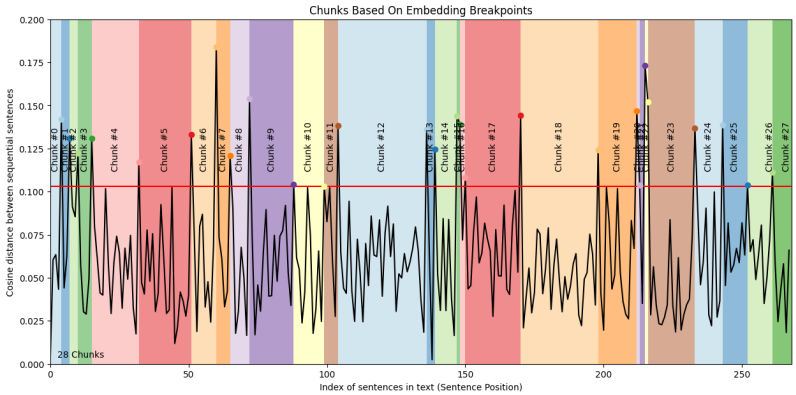

Figure 14: Illustration of the segmentation of two WikiText articles through semantic splitting. The plot displays the cosine similarity of all sentences, with the red line marking the breakpoint percentile threshold for cutting at 90.

Chunking	max chunk size	Pourcent thresl	buffer size	overlap	model	Hit Rate	MRR	model	Hit Rate	MRR	model	Hit Rate	MRR	nodes nb
Sentence splitter	512	None	None	20	bge - None	0.918	0.814	BM25-None	0.882	0.778	BM25-bge	0.916	0.857	11717
Sentence splitter	1024	None	None	20	bge - None	0.908	0.782	BM25-None	0.866	0.776	BM25-bge	0.9	0.822	6057
Sentence splitter	1024	None	None	0	bge - None	0.908	0.783	BM25-None	0.838	0.740	BM25-bge	0.900	0.853	11640
Recursive splitter	512	None	None	20	bge - None	0.924	0.880	BM25-None	0.772	0.694	BM25-bge	0.832	0.812	69 025
Recursive splitter	1024	None	None	20	bge - None	0.928	0.853	BM25-None	0.784	0.715	BM25-bge	0.886	0.855	33319
Recursive splitter	1024	None	None	0	bge - None	0.896	0.857	BM25-None	0.716	0.662333	BM25-bge	0.784	0.768	68130
Semantic	None	90	2	0	bge - None	0.77	0.707	BM25-None	0.644	0.576333	BM25-bge	0.712	0.693	27611
Semantic + max chunk size	2048	90	2	0	bge - None	0.784	0.724	BM25-None	0.71	0.643	BM25-bge	0.774	0.743	26053
Semantic + min length	None	90	2	0	bge - None	0.839	0.780	BM25-None	0.70	0.618	BM25-bge	0.755	0.746	21008
Semantic pairs + min length	None	80	2	0	bge - None	0.862	0.805	BM25-None	0.698	0.621667	BM25-bge	0.776	0.751	35059
Semantic pairs + min length + para	None	95	2	0	bge - None	0.938	0.846	BM25-None	0.818	0.728	BM25-bge	0.872	0.831	18184
Semantic pairs + min length	None	90	2	0	bge - None	0.9	0.814	BM25-None	0.81	0.723	BM25-bge	0.896	0.87	15937
Semantic pairs + min length	None	95	2	0	bge - None	0.896	0.822	BM25-None	0.8	0.698	BM25-bge	0.858	0.818	11141
Chain	None	None	2	0	bge - None	0.924	0.89	BM25-None	0.776	0.719	BM25-bge	0.854	0.838	16095
Chain	None	None	1	0	bge - None	0.916	0.846	BM25-None	0.82	0.733	BM25-bge	0.869	0.823	39138
Chain + para	2048	0.4	2	0	bge - None	0.958	0.86	BM25-None	0.874	0.785	BM25-bge	0.93	0.878	12225
LDA	None	None	None	0	bge - None	0.92	0.826	BM25-None	0.83	0.751	BM25-bge	0.892	0.86	14525

Figure 15: Test results for HR and MRR using various chunking methods on the WikiText dataset from Hugging Face ("Salesforce/wikitext"). The evaluation involved 500 precise questions generated by the Llama 3 8B model based on these documents. The notation model refers to the embedding model name followed by the re-ranker name. bge represents the bge-small-en embedding model. The green rows represent the baselines, where paragraph separation is followed by method-based splitting. The purple row highlights the best-performing method. "nodes nb" refers to the number of chunks created by the splitting method.

Chunking	model	Hit Rate	MRR	model	Hit Rate	MRR	model	Hit Rate	MRR	nodes nb
agentic + semantic	bge - None	0.969	0.896	BM25-None	0.908	0.821	BM25-bge	0.962	0.937	130
LLM prompting	bge - None	0.981	0.902	BM25-None	0.955	0.872	BM25-bge	0.970	0.948	130

Figure 16: Test results for HR and MRR using different chunking methods on a subset of 100 WikiText articles from Hugging Face ("Salesforce/wikitext"). This subset was selected due to computational constraints. The evaluation was conducted on 100 precise questions generated by the Llama 3 8B model based on these articles. In the results, the term model indicates the embedding model followed by the re-ranker name, with bge representing the bge-small-en embedding model.

Chunking	model	Hit Rate	MRR	model	Hit Rate	MRR	model	Hit Rate	MRR	model	Hit Rate	MRR	nodes nb
Recursive 2048	bge - None	0.484	0.395	bge-bge	0.484	0.439	BM25-None	0.574	0.506	BM25-bge	0.622	0.559	28998
Semantic b=1 2048	bge - None	0.518	0.443	bge-bge	0.61	0.554	BM25-None	0.602	0.526	BM25-bge	0.632	0.564	20835
Semantic pairs 1024	bge - None	0.54	0.452	bge-bge	0.612	0.547	BM25-None	0.648	0.574	BM25-bge	0.66	0.586	23823
Semantic pairs 2048	bge - None	0.522	0.439	bge-bge	0.614	0.555	BM25-None	0.608	0.526	BM25-bge	0.646	0.584	20420
Chain 1024	bge - None	0.49	0.41	bge-bge	0.568	0.508	BM25-None	0.568	0.499	BM25-bge	0.61	0.537	21450
Chain 2048	bge - None	0.532	0.444	bge-bge	0.61	0.545	BM25-None	0.616	0.54	BM25-bge	0.642	0.579	21333
LDA 2048	bge - None	0.5	0.433	bge-bge	0.578	0.531	BM25-None	0.612	0.523	BM25-bge	0.64	0.583	23481

Figure 17: Test results for HR and MRR using various chunking methods on the ICRC dataset. The evaluation involved 500 precise questions generated by the Llama 3 8B model based on these documents. The notation model refers to the embedding model name followed by the re-ranker name. bge represents the bge-small-en embedding model. The purple row highlights the best-performing method.

Embedding model for semantic splitting	model	Hit Rate	MRR	model	Hit Rate	MRR	model	Hit Rate	MRR	model	Hit Rate	MRR	nodes nb
w601sxs/b1ade-embed	bge - None	0.54	0.452	bge-bge	0.612	0.547	BM25-None	0.648	0.574	BM25-bge	0.66	0.586	23823
intfloat/multilingual-e5-base	bge - None	0.536	0.459	bge-bge	0.586	0.532	BM25-None	0.620	0.556	BM25-bge	0.648	0.587	23844
Lajavaness/bilingual-embedding-large	bge - None	0.506	0.416	bge-bge	0.582	0.519	BM25-None	0.608	0.525	BM25-bge	0.62	0.557	23795
dunzhang/stella_en_400M_v5	bge - None	0.536	0.44	bge-bge	0.596	0.527	BM25-None	0.61	0.532	BM25-bge	0.64	0.565	23806

Figure 18: Test results for HR and MRR using various embeddings model used for semantic splitting on the ICRC dataset. The evaluation involved 500 precise questions generated by the Llama 3 8B model based on these documents. The notation model refers to the embedding model name followed by the re-ranker name. bge represents the bge-small-en embedding model. The purple row highlights the best-performing method.

Retriever Embedding model	Hit Rate	MRR	Reranker	Hit Rate	MRR
BAAI/bge-small-en	0.54	0.452	bge	0.612	0.547
dunzhang/stella_en_400M_v5	0.66	0.555	bge	0.714	0.633
intfloat/multilingual-e5-large	0.586	0.485	bge	0.64	0.575

Figure 19: Test results for HR and MRR using various embeddings model used for retrieval on the ICRC dataset. The evaluation involved 500 precise questions generated by the Llama 3 8B model-based. bge-small-en was the embedding model used. The purple row highlights the best-performing method.

Model	epochs	batch size	learning rate	HR	MRR
Base	/	/	/	0.638	0.541
Queries Finetuned	4	10	2,00E-05	0.7	0.588
Queries Finetuned	2	10	2,00E-05	0.681	0.589
Queries Finetuned	4	4	2,00E-05	0.671	0.574
Topics Finetuned	4	10	2,00E-05	0.526	0.432
Linear Adapter	4	10	2,00E-05	0.656	0.558
Linear Adapter	4	10	0.01	0.336	0.256
2 layers NN Adapter	25	10	2,00E-05	0.639	0.543
2 layers NN Adapter	4	10	2,00E-05	0.662	0.563
XGBoost	4	10	2,00E-05	0.65	0.54

Figure 20: Test results for HR and MRR with different improvements to the embedding model used for retrieval on the ICRC dataset. Based on these documents, the evaluation was conducted using 500 precise questions generated by the Llama 3 8B model. A 7:3 train test split was used. Stella en 400M v5 was the embedding model used. The purple row highlights the best-performing method.

Figure 21: Part of a Knowledge Graph built using triplets extracted with Sciphi Triplex. A subset of 10 chunks of the ICRC dataset was used.