A Hybrid Retrieval Approach for Advancing Retrieval-Augmented Generation Systems

Nguyen Nam Doan[1], **Aki Harma**[1], **Remzi Celebi**[1],**Valeria Gottardo**[2]
[1]Maastricht University, The Netherlands
[2]NLC Health Ventures, The Netherlands
{nam.doan, aki.harma, remzi.celebi}@maastrichtuniversity.nl
valeria.gottardo@nlc.health

Abstract

Retrieval-Augmented Generation (RAG) has become a promising solution for utilizing Large Language Models (LLMs) in domain-intensive question-answering tasks. The performance of RAG is greatly influenced by the retriever component, which typically relies on semantic similarity between the text embeddings of the query and the passages to identify the relevant context. However, text embedding models may only capture the semantic meaning of individual passages, potentially neglecting global relationships between them. To address this limitation, we propose a hybrid retrieval method that integrates embeddings encoded from textual and knowledge graph information. Although in this paper, the knowledge graphs describe the passage relationships in a health-tech industry use case, the hybrid embedding solution is designed to be generic. Furthermore, the proposed retrieval approach aims to offer straightforward implementation without requiring complex joint learning processes. Our results on custom test sets demonstrate significantly enhanced accuracy and ranking of the retriever, thus, supporting the LLM-based reader component in generating more accurate responses.

1 Introduction

The use of pre-trained Large Language Models (LLMs) has gained considerable attention for specific question-answering (QA) tasks, due to their ability to understand and generate natural language (De Angelis et al., 2023). This capability results from their extensive pre-training on diverse text datasets and a large number of parameters, which endows them with the ability to memorize and utilize learned knowledge (Roberts et al., 2020; Petroni et al., 2019). However, implementing pre-trained services within corporate settings faces certain challenges. One notable limitation is the inability to generate accurate and faithful responses for questions related to specific domains and business use cases, often referred to as "hallucination". This constraint stems from knowledge boundaries, which include a lack of specialized domain knowledge and the absence of up-to-date information in the pre-trained data (Huang et al., 2023).

Fine-tuning generative LLMs with a target domain dataset has been proposed as a solution to this problem. This approach adapts the models for specific tasks and often outperforms pre-trained foundation models (Han et al., 2023; Wu et al., 2023; Chen et al., 2023). Nonetheless, training a billion-parameter model often requires significant computational resources and high-quality training datasets to obtain optimal results (Chen et al., 2023). Moreover, this method only offers a temporary solution, as over time the knowledge will be outdated again, leading to a loop of knowledge boundary problem.

Another approach to improve the domain factuality and reduce hallucination of the responses is using the Retrieval-Augmented Generation (RAG) method (Lewis et al., 2021; Izacard and Grave, 2021). The main idea of the RAG method is to use the retriever-reader framework to combine information retrieval (Karpukhin et al., 2020) with LLMs. Specifically, the RAG uses a retriever to select a set of relevant knowledge to the questions, which helps narrow down the answerable evidence for the LLM-based reader. The reader then synthesizes the answer to the query based on the given information like in an "open-book" exam. Therefore, this approach offers the advantage of providing external knowledge to LLMs without requiring the fine-tuning of the models. The responses are also more reliable due to the augmentation of retrieved contexts, which serve as the supporting evidence. Besides, RAG is suitable for both open-domain and closed-domain question-answering tasks, and can also support private use cases depending on the configuration of external data sources.

A recent study showed that the performance of RAG can be hindered by false retrieval, where the system fails to provide accurate information to the

generators (Barnett et al., 2024). Several methods aiming to enhance the retriever's performance involve improving query-text embeddings to re-rank pre-retrieved passages (Nogueira and Cho, 2020; Mao et al., 2021; Askari et al., 2023). However, these methods rely solely on textual methods, which may not always be optimal. For example, text embedding models regularly treat input passages independently and do not capture global relations among them (Yu et al., 2022). This inability to capture the dependencies between related passages can potentially restrain the performance of the system (Min et al., 2020). A possible approach to address this problem is to use knowledge graphs (KG) in conjunction with textual information to enhance the retriever in question-answering systems (QAS), providing more robust text representations. This approach has been investigated by studies such as joint representation learning with two modalities to improve re-ranking and answering performance (Yu et al., 2022; Zhou et al., 2020; Ju et al., 2022). Although these methods have shown better results compared to using text input alone, their algorithms and training processes are often costly and complex.

To address the retrieval challenges, this paper aims to enhance the retriever with a comprehensive embedding component that combines both textual data and knowledge graphs (KGs). Unlike previous studies, we propose a simple hybrid pipeline for generating these representations, rather than training a complex joint learning model. The method and experiment were conducted within the use case of a health-tech venture builder, where questions were posed to find answers in proprietary health-related documents. However, the hybrid embedding method has the potential to be domain-agnostic, as long as its components are customized for specific contexts. The experimental results show that our hybrid method helps improve not only the information retrieval performance but also the generative response quality from different LLM-based readers.

2 Related Work

In QAS using the "retrieve-then-read" mechanism, the text embedding plays an important role in the retriever component. State-of-the-art retrievers use dense text embedding methods, often utilizing neural networks like BERT (Devlin et al., 2019) to encode the semantic meaning of the text into dense vectors (Karpukhin et al., 2020; Reimers and Gurevych, 2019; Xiao et al., 2023). Typically, the Dense Passage Retriever (DPR) (Karpukhin et al., 2020) framework learns embeddings for questions and passages using two separate BERT networks with metric learning. Sentence-BERT (Reimers and Gurevych, 2019) also employs metric learning but allows a single BERT to learn embeddings for two sentences through a shared-weight configuration. DPR and Sentence-BERT have also become fundamental approaches to pre-train other general-purpose BERT-based embedding models such as BAAI General Embedding (BGE) (Xiao et al., 2023), General Text Embedding (GTE) (Li et al., 2023b), E5 (Wang et al., 2022), etc. These models can be further fine-tuned for specific downstream tasks(Choi et al., 2021). However, these frameworks were not originally designed to capture the semantic connections between different passages. This drawback can be tackled by taking advantage of structural information captured by knowledge graph solutions. For instance, the Knowledge-aid open-domain QA (KAQA) framework (Zhou et al., 2020) used two KGs representing the relationship between the question and document, and between retrieved documents to support re-ranking the retriever. Also, to improve the retriever by re-ranking, KG-FiD (Yu et al., 2022) used inter-passage relation KG with the graph attention network (Veličković et al., 2018) to update the representation vectors of the passage. Min et al.(Min et al., 2020) introduced an extended passage-level KG and integrated it into the retriever and reader to improve context coverage and response accuracy. In the health domain, (Wise et al., 2020) introduced a KG describing the relationships between scientific articles on COVID-19 and used TransE-based embeddings for article retrieval and recommendation. While these studies obtained impressive results, their implementation and training processes involved the integration of multiple intensive computational models. Inspired by these works, however, the primary goal of this paper is to achieve a more straightforward implementation approach that leverages the knowledge graph to improve conventional retriever and RAG performance.

3 Proposed Method

The conceptual pipeline of the proposed retriever used in the RAG system is illustrated in Figure 1.

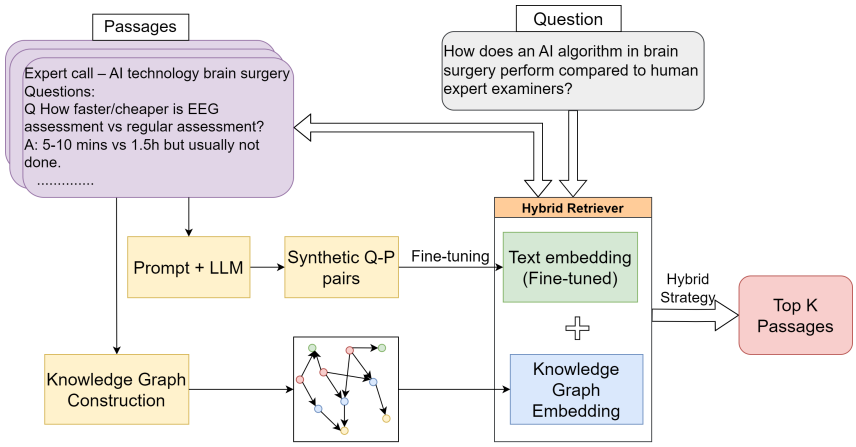


Figure 1: The overall conceptual design of our hybrid retriever implemented in the RAG system.

The goal is to improve the quality of the Top-K passages retrieved in response to a given question. This is achieved by measuring the similarity between hybrid embeddings, which combine text and knowledge graph representations of both the Question and the Passages. To adapt the text embedding model to the use case, a fine-tuning process is performed with a synthetic dataset generated by an LLM. Next, an automatic KG construction is proposed, aiming to present the global relationship between Passages. Lastly, we will introduce a strategy aimed at effectively integrating these two modalities into the hybrid retriever.

3.1 Data collection and description

The data for this study consists of interviews with medical experts from a private use case provided by a venture builder, focusing on the technical and business aspects of their medical innovations. The interviews are documented and categorized into 3 main topics, including medical technology, biotechnology, and digital/AI. These documents serve as the knowledge source for the RAG system for the use case.

Firstly, the interview documents stored in the company's database spanning the last 4 years are collected. Document lengths vary, averaging 697 words. After removing the documents with insufficient content, the dataset contains $1,487$ documents. Next, each document undergoes automated pre-processing steps, including the removal of special tokens, images, and tables, as well as English translation.

Finally, the documents are chunked into smaller passages to enhance searchability in the retrieval stage while optimizing computational resources. We choose the chunk size of 512 tokens to fit the small-size BERT-based models in the text embedding step. Furthermore, two adjacent passages of a document are set to overlap by 20 tokens, ensuring a smooth transition of context between them. After the chunking step, the processed dataset comprises a total of 5607 passages.

3.2 Fine-tuning text embedding model

In the context of this paper, the user input questions and the passages' context are distinct, integrating various aspects of the health-tech industry. Hence, using pre-trained text embedding models in model zoos may not sufficiently capture these nuances for retrieval purposes. To address that issue, a BERT-base embedding model is fine-tuned through the training process of the Sentence-BERT (SBERT) framework (Reimers and Gurevych, 2019) (see Appendix A) on our custom dataset.

3.2.1 Constructing training set

The fine-tuning dataset consists of positive query and source passage pairs. Manually annotating these pairs from interview documents is time-consuming, so to simulate real-life scenarios, we use generative LLMs to comprehend the source passages and automatically generate corresponding queries. In each positive pair, the synthetic query is crafted to reflect the questions related to specific information in the source passage, which serves as the ground truth for the query in the retrieval task.

Given that the documents in this study contain private and sensitive information, local and open-source LLMs were selected for query generation instead of using services like OpenAI API. Concretely, small-size but high-performing generative LLMs such as the Zephyr-7B-beta model (Tunstall

et al., 2023) and Mistral-7B-OpenOrca [1] are chosen to generate synthetic queries. For each source passage, a single query is synthetically generated by inserting this passage into a prompt to instruct the LLMs. For generalization, we use the Zephyr-7B-beta model for generating the training queries and Mistral-7B-OpenOrca for constructing the test ones. In addition, the instruction prompts for the training and test sets are modified to be slightly different. In this work, these instructions follow a role-playing prompting strategy (Li et al., 2023a). For example, the LLM can be instructed to take on the role of a teacher with the task of generating exam questions based on the passages (see Appendix B). To ensure accurate LLM responses, the prompts were carefully designed, and a set of generated questions was reviewed to confirm they resembled real-world queries.

3.2.2 Training process

The BERT-based model in SBERT is fine-tuned using the Multiple Negative Ranking loss (MNRL) function (Henderson et al., 2017). Mathematically, the loss function is optimized by minimizing the mean negative log-probability of the positive pairs, shown as follows:

$$
\begin{aligned}
L(\mathbf{q}, \mathbf{p}) &= -\frac{1}{K}\sum_{i=1}^{K} \log\left(P(q_i, p_i)\right) \\
&= -\frac{1}{K}\sum_{i=1}^{K} \log\left(\frac{e^{S(q_i,p_i)}}{\sum_{j=1}^{K} e^{S(q_i,p_j)}}\right)
\end{aligned}
\tag{1}
$$

in which, for a batch size of K, there are K input queries $\mathbf{q} = (q_1, ..., q_K)$ and K corresponding passages $\mathbf{p} = (p_1, ..., p_K)$. The positive pair is denoted as (q_i, p_i) for every $i \leq K$ while the negative pair is indicated as (q_i, p_j) with $i \neq j$ and $i, j \leq K$. To optimize the loss, Adam with decoupled weight decay (AdamW) (Loshchilov and Hutter, 2019) is used along with the warm-up decrease of learning rate enhances for better stability and generalization during training.

In the inference stage, the BERT-based model is taken out of SBERT and used independently. To assess the effectiveness of fine-tuned text embeddings, we then evaluate the retriever's performance on the test set, comparing it to the retrievers using only pre-trained embeddings.

[1] https://huggingface.co/Open-Orca/Mistral-7B-OpenOrca

3.3 Knowledge Graph Construction and Embedding

In the proposed system, the role of the KG is to model the connections and relationships between different passages in the dataset. The KG is then embedded in a vector space by a knowledge graph embedding (KGE) model, such that the structural features between passages are preserved through their vector representations.

3.3.1 Knowledge Graph Construction

The KG consists of a set of triples in the form of *(head, relation, tail)* and is constructed following a schema depicted in Figure 2.

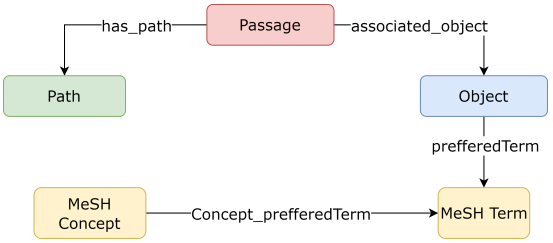


Figure 2: The schema to formalize the knowledge graph construction.

In this schema, the head and tail entities from the triples are categorized into one of five classes, with the Passage, Path, and Object classes serving as general base classes. To enhance the schema for the health-related domain, we incorporate the Medical Subject Headings (MeSH) concepts, MeSH term classes, and their relationships from the MeSH RDF-linked data (Lipscomb, 2000). MeSH, originally a biomedical vocabulary used for indexing and searching PubMed articles, is integrated to enrich the KG with relevant terms. Detailed information about the meaning of entity classes and how these entities are created is described as follows:

Passage entities: The entities belonging to the Passage class represent the text chunks in our dataset. Each Passage entity is defined by its ID in the database.

Path entities: This type of entity represents the path address of the source documents (i.e. the documents before being split into passages) in the database.

Object entities: This type of entity represents the general and bio-medical annotations from the passage. To extract the object entities from the text passages, we leverage the "en_core_sci_lg" model from the SciSpacy library (Neumann et al., 2019) as a Named-entity Recognition (NER) tool. Addi-

tionally, each object is linked to a MeSH Descriptor (i.e. a term that describes the main subject of an article) by a Name Entity Linking function. It leads to a total of $65,282$ objects extracted from our passages, in which 6901 objects have linked MeSH Descriptors.

MeSH Concept entities: The MeSH Concept class describes a unit of meaning. In other words, every term in MeSH which is strictly synonymous with each other is grouped into a "Concept". In MeSH, each Descriptor consists of one or more Concepts. Therefore, the MeSH Concept entities in our KG are then retrieved by using a SPARQL[2] query based on the MeSH Descriptor.

MeSH Term entities: The MeSH Term class describe human-readable names used by a MeSH Concept or MeSH Descriptor. A MeSH Descriptor have one MeSH Term while A MeSH Concept can have one or multiple MeSH Terms and they are strictly synonymous. All MeSH Terms are retrieved by SPARQL query based on MeSH Concepts and MeSH Descriptors.

Additionally, entities of different classes are connected by 4 relations. The descriptions of relations used to link head and tail entities are described as follows:

associated_object: This relation describes the connection between Passage and Object entities. It demonstrates what object entities are mentioned in the text.

has_path: This relation connects between Passage and Path entities, indicating the paths where the passages are located.

preferredTerm: This connection between Object and MeSH Term entities indicates which term the Object entity is preferred to refer to.

Concept_preferredTerm: The connection between MeSH Concept and MeSH Term entities, describing the synonym relation.

An example of a subgraph and detailed statistics of the KG is demonstrated in Appendix C.

3.3.2 Knowledge Graph Embedding

The knowledge graph after being constructed is then represented in the vector space by a knowledge graph embedding (KGE) model such that the graph properties are preserved. Although the methodology is applicable to any KGE model, we opt for translational KGE models because of their simplicity and high efficiency.

[2] https://www.w3.org/TR/sparql11-query/

Generally, translational KGE models operate by using relation embeddings as translations in vector space between head and tail entities. The objective is to learn the embedding of entities and relations in triples to minimize the scoring function $f_r(\mathbf{h}, \mathbf{t})$ of each triple *(h, r, t)* where r is the relation, h and t are head and tail entity embeddings, respectively. Table 1 shows the scoring functions of different KGE models used in this paper.

Table 1: Scoring function equations for TransE, RotatE, and QuatE models. $\circ$ denotes element-wise product, $\otimes$ denotes Hamilton product, $\|\cdot\|_2^2$ denotes the L2 norm.

Model	Scoring Function
TransE (Bordes et al., 2013)	$\|\|\mathbf{h} + \mathbf{r} - \mathbf{t}\|\|_2$
RotatE (Sun et al., 2019)	$\|\|\mathbf{h} \circ \mathbf{r} - \mathbf{t}\|\|_2$
QuatE (ZHANG et al., 2019)	$\|\|\mathbf{h} \otimes \mathbf{r} - \mathbf{t}\|\|_2$

The scoring function is optimised through a Margin Ranking loss function, shown as follows:

$$\mathcal{L}(f_p, f_n) = \max(0, f_p - f_n + \lambda) \qquad (2)$$

in which λ denotes the margin value, f_p and f_n are the scoring function values of a positive triple (i.e. the actual triple in KG) and a negative triple (i.e. the non-existent triple in KG), respectively. This loss function aims to encourage the model to improve its embedding representations and effectively distinguish between positive and negative triples. To sample negative triples, the head or tail of a positive triple is randomly swapped with an entity from another one in the training batch. This process is carefully engineered to ensure that the resulting corrupted triples do not already exist as positive examples in the original KG. Finally, the objective of learning embedding for entities and relationships can be achieved by using the Stochastic Gradient Descent algorithm.

3.4 Hybrid Retrieval Strategy

Figure 3 illustrates the process of our hybrid retrieval strategy. The strategy is divided into five main steps as follows:

Step 1: Given a question, a set of top N (N > K) relevant passages are retrieved using the cosine similarity of their text embeddings. This step aims to narrow the search space by filtering out irrelevant passages based on their semantic nuances.

Step 2: From the top N retrieved passages, their text embeddings and KG embeddings are horizontally concatenated. In the concatenation vectors,

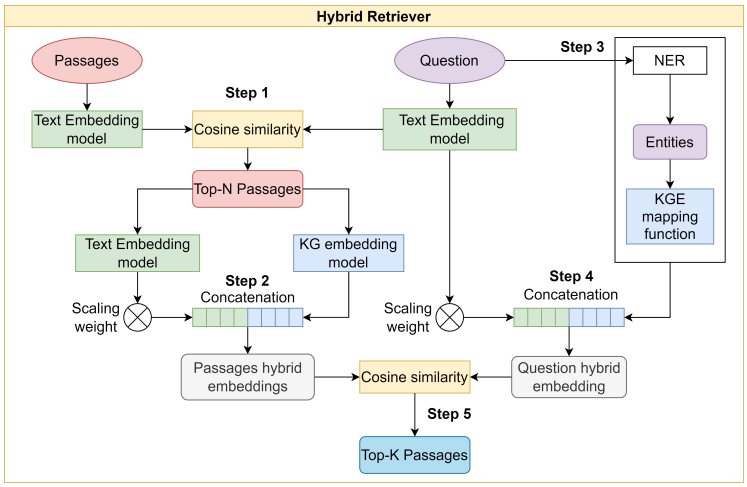


Figure 3: The hybrid retrieval strategy aims to utilise the fusion of text and KG embeddings.

we prioritize the impact of the text embedding component to emphasize the importance of semantic nuances, while utilising the KG embedding component as an auxiliary extension. Thus, text embeddings are multiplied element-wise by a scaling weight α (where $\alpha > 1$) to amplify their significance. The concatenation operation is then specifically formulated as follows:

$$Concat(TE, KGE) = [TE * \alpha, KGE] \quad (3)$$

where TE and KGE indicate a text embedding and a knowledge graph embedding respectively. With the amplification of text embedding, the hybrid representations of passages have more internal semantic features while still containing the global relationship information captured by KG-based vectors.

Step 3: Since the question is not explicitly modelled in our KG, approximating its representation in the KG vector space is needed. Accordingly, all objects in the question are first extracted by using the same NER model in Section 3.3. Next, the question embedding is then approximated by a mapping function, shown as follows:

$$\mathbf{q} = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M}\mathbf{p}_{ij}}{\sum_{i=1}^{N}\sum_{j=1}^{M}\text{passage\_has\_obj}(p_i, o_j)} \quad (4)$$

where M is the number of objects extracted in the question, N is the number of top N retrieved passages, $\mathbf{q}$ is the approximated embedding of a question, $\mathbf{p}_{ij}$ is the embedding of passage p_i in top N that contains object o_i, and

$$\text{passage\_has\_obj}(P_i, O_j) = \begin{cases} 1 & \text{if } p_i \ni o_j \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

Step 4: The text embedding and the approximated KGE of the question are concatenated to form the hybrid vector. Also, the same weight α value is applied to the text embedding of the question before concatenation.

Step 5: Finally, the top K passages are retrieved from the N passages by taking the K highest cosine similarity scores between the hybrid vectors of the question and the N passages.

4 Experiments

4.1 Experimental Setup

4.1.1 Text embedding model setup

For the text embedding component in the retriever, three small-size BERT-based models were selected as the baselines: BGE model (Xiao et al., 2023), E5 model (Wang et al., 2022) and GTE model (Li et al., 2023b). Each model comprises 12 transformer blocks, 12 attention heads per self-attention layer, an input size of 512 tokens, and an embedding size of 384.

The training set to fine-tune the models was generated from 500 random passages, following the method in Section 3.2. The models were trained in 50 epochs, with a batch size of 10 and using the AdamW optimizer. The initial learning rate was configured at 0.001 and with a decay through a warm-up step of 0.1. We trained the models on Google Colab Pro with 15GB VRAM NVIDIA T4.

4.1.2 Knowledge graph embedding configuration

Three KGE models were experimented with, including TransE, RotatE, and QuatE. Each model's embedding size of 200 was configured for optimal

performance based on grid search. The models were trained on 20 epochs. Each training batch consisted of 128 positive triples, where for each positive triple, one negative triple was generated.

4.1.3 Hybrid Retriever configuration

The top-N value was initialized at 20. Experimentation included top-K values of 3 and 5. Various combinations of BGE text embeddings and KGE models were tested for the hybrid embeddings. Through grid search, a weight of $\alpha = 2.5$ was found to optimize performance, ensuring the hybrid retriever achieved its peak effectiveness.

4.2 Evaluation Scheme

4.2.1 Passage Retrieval Evaluation

To evaluate the retriever, three synthetic test sets were created, denoted as the first, second, and third test sets, following the guidelines outlined in Section 3.2. The queries from the first and second test sets were generated from the same random 200 passages. In the first test set, one question was synthesised per passage, while in the second test set, the ratio was 2:1. The third test set was created from a different set of 200 passages, with a 1 question per passage ratio.

The retriever performance was measured by two metrics: Hit Rate (HR) for retrieval accuracy and Mean Reciprocal Rank (MRR) for ranking ability. The higher the metric values, the better the performance.

4.2.2 RAG performance evaluation

To evaluate an end-to-end RAG, another test set was used that included 50 golden question-answer pairs manually extracted from the documents. Given the question, the correctness of RAG's response was measured by comparing it to the golden answer, using Semantic Answer Similarity (SAS) score (Risch et al., 2021). SAS score is calculated by a cross-encoder model given the predicted answer and golden answer pairs. In this paper, the cross-encoder version of the BGE model was used. The SAS score also measured the level of relevancy between the RAG responses and the retrieved passages.

4.3 Experimental Results

4.3.1 Passage Retrieval Results

The comparison of retrievers using different embedding models is demonstrated in Table 2. Across all test sets, retrievers using the fine-tuned text embedding models exhibited superior performance compared to their pre-trained counterparts in both HitRate@K and MRR@K metrics. Notably, the fine-tuning of GTE and E5 models resulted in more pronounced enhancements. Although the fine-tuned BGE model showed only a slight improvement over its pre-trained version, both iterations of BGE outperformed the E5 and GTE models. These improvements were consistent across all top-K scenarios. These findings also suggest that relying solely on general-purpose text embedding models, which leverage pre-trained knowledge, may not yield optimal results for domain-specific datasets. The proposed fine-tuning solution can significantly enhance performance and stabilize results in such cases.

On the other hand, our method of integrating fine-tuned BGE with any knowledge graph embedding model demonstrated notable enhancements compared to BGE-only retrievers across all test sets and Top-K settings. Particularly, the hybrid approach showed increases of up to 8.1% in HitRate and up to 8.7% in MRR. Notably, leveraging TransE embeddings, the hybrid retriever produced the highest results compared to other hybrid variations. This emphasizes the effectiveness of leveraging global semantic relationships to enrich the textual representation of both passages and queries, hence, enhancing the overall retrieval performance.

4.3.2 End-to-end RAG performance

The effectiveness of the hybrid retriever was further evaluated through its impact on the answering performance of RAG systems. For this experiment, the hybrid retriever employed fine-tuned BGE and TransE models and returned the top 3 passages. The performance of LLM-based readers was then compared across three conversational LLMs: LLaMA-2-13B-chat (Touvron et al., 2023), Zephyr-7B-beta and Mistral-7B-OpenOrca.

Table 3 displays the response quality scores for various RAG settings. In terms of Relevancy, the scores remained relatively consistent, suggesting that all LLM baselines could properly answer questions following the retrieved contexts. However, our analysis of Correctness scores revealed a notable enhancement in RAG performance when utilizing hybrid retrievers, with improvements of up to 13.1%. This highlights the significant impact of hybrid retrievers on the accuracy of RAG-generated responses.

Table 2: The performance comparison between retrievers using hybrid embeddings and those using only text embeddings.

Retriver	1st Test set		2nd Test set		3rd Test set	
	HR	MRR	HR	MRR	HR	MRR
Top K = 3						
$GTE_{pre\text{-}trained}$	0.3865	0.2938	0.3291	0.2324	0.4690	0.3427
$E5_{pre\text{-}trained}$	0.6546	0.5567	0.5753	0.4882	0.6804	0.5506
$GTE_{fine\text{-}tuned}$	0.7989	0.6993	0.6934	0.5644	0.7886	0.6683
$E5_{fine\text{-}tuned}$	0.7525	0.6125	0.6231	0.5201	0.7474	0.6091
$BGE_{pre\text{-}trained}$	0.8247	0.6941	0.7311	0.6139	0.8195	0.6821
$BGE_{fine\text{-}tunned}$	0.8350	0.7164	0.7437	0.6335	0.8350	0.6941
BGE_{ft} + TransE	**0.8917**	**0.7506**	**0.8040**	**0.6892**	0.8814	0.7336
BGE_{ft} + RotatE	0.8763	0.7526	0.7839	0.6570	**0.8865**	**0.7431**
BGE_{ft} + QuatE	0.8711	0.7465	0.7814	0.6440	0.8763	0.7250
Top K =5						
$GTE_{pre\text{-}trained}$	0.4742	0.3136	0.3994	0.2487	0.5515	0.3610
$E5_{pre\text{-}trained}$	0.7577	0.5798	0.6482	0.5051	0.7989	0.5774
$GTE_{fine\text{-}tuned}$	0.8608	0.7129	0.7587	0.5795	0.8556	0.6838
$E5_{fine\text{-}tuned}$	0.7938	0.6223	0.6909	0.5356	0.7886	0.6189
$BGE_{pre\text{-}trained}$	0.8505	0.6993	0.7989	0.6292	0.8659	0.6926
$BGE_{fine\text{-}tunned}$	0.8917	0.7298	0.8090	0.6485	0.8917	0.7080
BGE_{ft} + TransE	**0.9329**	**0.7652**	**0.8542**	**0.7007**	**0.9381**	**0.7435**
BGE_{ft} + RotatE	0.9175	0.7616	0.8517	0.6726	0.9175	0.7498
BGE_{ft} + QuatE	0.9072	0.7542	0.8316	0.6560	0.9175	0.7341

Table 3: Responses comparison between different RAG's combinations.

LLM reader	Retriever	Relevancy	Correctness
LLama-2-13B-chat	BGE ft	0.9846	0.7655
Mistral-7B-OpenOrca	BGE ft	0.9909	0.7193
Zephyr-7B-beta	BGE ft	0.9708	0.7486
LLama-2-13B-chat	hybrid	0.9921	0.7729
Mistral-7B-OpenOrca	hybrid	0.9891	0.8136
Zephyr-7B-beta	**hybrid**	**0.9878**	**0.8283**

The experimental results also correlated with the correctness of answer examples shown in Table 7 (see Appendix F). The answers from the Zephyr-7B-beta model based on hybrid retriever contexts were more detailed and aligned better with the golden answer than those from BGE-only retriever contexts, which contained less information and thus, had a lower correctness score.

5 Discussion

In this section, we will discuss how the hybrid embeddings help to improve the retriever by analyzing the impacts of the weight α. Additionally, a comparison of the proposed method with other re-ranking mechanisms will be demonstrated. In these analyses, we experimented with a sub-case using our 1^{st} retriever test set introduced in Section 4.1.

As shown in Figure 6 (see Appendix D), it is clear that the value of α greatly affects the performance of the hybrid retriever. When α was set to 0, it was equivalent to the case of a retriever only using TransE embeddings. However, the performance of the retriever in that case was poorly underperformed, especially when the top-N value increased. This observation suggests that when the retriever relies solely on KGE and retrieves information from a larger pool, it is more prone to noise and irrelevant information. When α was set to 1, text and KG embeddings were equally concatenated. While this improved hybrid retriever performance, it decreased at higher top-N values. This finding indicates that balanced weights might cause KG embeddings to diminish the semantic meaning of text embeddings in their hybrid vectors, resulting in unstable outcomes.

Conversely, as the parameter α increases, it boosts the influence of text embeddings. This, in turn, strengthened the semantic features in the combined vectors, resulting in better performance for the retriever. Notably, with larger α values, the performances remained relatively unchanged by variations in top-N values. This observation indicates that the α factor can aid hybrid vectors in differentiating dissimilar ones. However, it is crucial to keep α values within an appropriate range. If α is too large, the text embedding features can overshadow the global features from KGE, causing the hybrid vectors to resemble text embeddings too closely.

Additionally, our two-stage hybrid retrieval strategy can be considered to be similar to the "retrieve then re-rank" mechanism. However, instead of using a re-ranking model, the hybrid vectors take the KGE component to re-rank the pre-defined passage orders. The results show that our hybrid retriever had comparative performance compared to other "retrieve then rerank" paradigms despite not being intentionally designed for re-ranking purposes (see Appendix E). Furthermore, the proposed hybrid retriever only uses cosine similarity to retrieve passages, which is computationally lighter for inference than using neural-based re-ranker models.

6 Conclusion

In this work, a hybrid method is introduced which leverages both text and knowledge graph embeddings to advance the retriever in Retrieval-Augmented Generation systems. The text embedding model is fine-tuned with a synthetic dataset

to adapt to downstream tasks. Meanwhile, the KGE component is trained from a KG presenting the global relationships between passages in the dataset. Additionally, the proposed hybrid retrieval strategy efficiently integrates these two representation types without using complex architecture or training processes typical in other KG-based retrieval methods.

The experimental results demonstrate that the method can significantly improve the retriever's performance in both accuracy and ranking in comparison to the baseline methods. This improvement subsequently results in the higher correctness of the RAG's responses.

In this paper, the methods were tested in an application in the health-tech domain where knowledge can be represented based on Medical Subject Headings (MeSH) classes. However, the proposed method is generic and can be applied to any domain where the KG can be meaningfully constructed to describe the relations between different passages.

References

Arian Askari, Amin Abolghasemi, Gabriella Pasi, Wessel Kraaij, and Suzan Verberne. 2023. Injecting the BM25 Score as Text Improves BERT-Based Rerankers. ArXiv:2301.09728 [cs].

Scott Barnett, Stefanus Kurniawan, Srikanth Thudumu, Zach Brannelly, and Mohamed Abdelrazek. 2024. Seven Failure Points When Engineering a Retrieval Augmented Generation System. ArXiv:2401.05856 [cs].

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating Embeddings for Modeling Multi-relational Data. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Zeming Chen, Alejandro Hernández Cano, Angelika Romanou, Antoine Bonnet, Kyle Matoba, Francesco Salvi, Matteo Pagliardini, Simin Fan, Andreas Köpf, Amirkeivan Mohtashami, Alexandre Sallinen, Alireza Sakhaeirad, Vinitra Swamy, Igor Krawczuk, Deniz Bayazit, Axel Marmet, Syrielle Montariol, Mary-Anne Hartley, Martin Jaggi, and Antoine Bosselut. 2023. MEDITRON-70B: Scaling Medical Pretraining for Large Language Models. ArXiv:2311.16079 [cs].

Hyunjin Choi, Judong Kim, Seongho Joe, and Youngjune Gwon. 2021. Evaluation of BERT and ALBERT Sentence Embedding Performance on Downstream NLP Tasks. ArXiv:2101.10642 [cs].

Luigi De Angelis, Francesco Baglivo, Guglielmo Arzilli, Gaetano Pierpaolo Privitera, Paolo Ferragina, Alberto Eugenio Tozzi, and Caterina Rizzo. 2023. ChatGPT and the rise of large language models: the new AI-driven infodemic threat in public health. *Frontiers in Public Health*, 11:1166120.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv:1810.04805 [cs]*. ArXiv: 1810.04805.

Tianyu Han, Lisa C. Adams, Jens-Michalis Papaioannou, Paul Grundmann, Tom Oberhauser, Alexander Löser, Daniel Truhn, and Keno K. Bressem. 2023. MedAlpaca – An Open-Source Collection of Medical Conversational AI Models and Training Data. ArXiv:2304.08247 [cs].

Matthew Henderson, Rami Al-Rfou, Brian Strope, Yunhsuan Sung, Laszlo Lukacs, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. 2017. Efficient Natural Language Response Suggestion for Smart Reply. ArXiv:1705.00652 [cs].

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2023. A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions. ArXiv:2311.05232 [cs].

Gautier Izacard and Edouard Grave. 2021. Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering. ArXiv:2007.01282 [cs].

Mingxuan Ju, Wenhao Yu, Tong Zhao, Chuxu Zhang, and Yanfang Ye. 2022. Grape: Knowledge Graph Enhanced Passage Reader for Open-domain Question Answering. ArXiv:2210.02933 [cs].

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. ArXiv:2004.04906 [cs].

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2021. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. ArXiv:2005.11401 [cs] version: 4.

Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023a. CAMEL: Communicative Agents for "Mind" Exploration of Large Language Model Society. ArXiv:2303.17760 [cs].

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023b. Towards General Text Embeddings with Multi-stage Contrastive Learning. ArXiv:2308.03281 [cs].

Carolyn E. Lipscomb. 2000. Medical Subject Headings (MeSH). *Bulletin of the Medical Library Association*, 88(3):265–266.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. ArXiv:1711.05101 [cs, math].

Yuning Mao, Pengcheng He, Xiaodong Liu, Yelong Shen, Jianfeng Gao, Jiawei Han, and Weizhu Chen. 2021. Reader-Guided Passage Reranking for Open-Domain Question Answering. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 344–350, Online. Association for Computational Linguistics.

Sewon Min, Danqi Chen, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2020. Knowledge Guided Text Retrieval and Reading for Open Domain Question Answering. ArXiv:1911.03868 [cs].

Mark Neumann, Daniel King, Iz Beltagy, and Waleed Ammar. 2019. ScispaCy: Fast and Robust Models for Biomedical Natural Language Processing. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 319–327. ArXiv:1902.07669 [cs].

Rodrigo Nogueira and Kyunghyun Cho. 2020. Passage Re-ranking with BERT. ArXiv:1901.04085 [cs].

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language Models as Knowledge Bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. ArXiv:1908.10084 [cs].

Julian Risch, Timo Möller, Julian Gutsch, and Malte Pietsch. 2021. Semantic Answer Similarity for Evaluating Question Answering Models. ArXiv:2108.06130 [cs].

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How Much Knowledge Can You Pack Into the Parameters of a Language Model? ArXiv:2002.08910 [cs, stat].

Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. FaceNet: A Unified Embedding for Face Recognition and Clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823. ArXiv:1503.03832 [cs].

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. ArXiv:1902.10197 [cs, stat].

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. ArXiv:2302.13971 [cs].

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M. Rush, and Thomas Wolf. 2023. Zephyr: Direct Distillation of LM Alignment. ArXiv:2310.16944 [cs].

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text Embeddings by Weakly-Supervised Contrastive Pre-training. ArXiv:2212.03533 [cs].

Colby Wise, Vassilis N. Ioannidis, Miguel Romero Calvo, Xiang Song, George Price, Ninad Kulkarni, Ryan Brand, Parminder Bhatia, and George Karypis. 2020. COVID-19 Knowledge Graph: Accelerating Information Retrieval and Discovery for Scientific Literature. ArXiv:2007.12731 [cs].

Chaoyi Wu, Weixiong Lin, Xiaoman Zhang, Ya Zhang, Yanfeng Wang, and Weidi Xie. 2023. PMC-LLaMA: Towards Building Open-source Language Models for Medicine. ArXiv:2304.14454 [cs].

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighof. 2023. C-Pack: Packaged Resources To Advance General Chinese Embedding. ArXiv:2309.07597 [cs].

Donghan Yu, Chenguang Zhu, Yuwei Fang, Wenhao Yu, Shuohang Wang, Yichong Xu, Xiang Ren, Yiming Yang, and Michael Zeng. 2022. KG-FiD: Infusing Knowledge Graph in Fusion-in-Decoder for Open-Domain Question Answering. ArXiv:2110.04330 [cs].

SHUAI ZHANG, Yi Tay, Lina Yao, and Qi Liu. 2019. Quaternion Knowledge Graph Embeddings. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Mantong Zhou, Zhouxing Shi, Minlie Huang, and Xiaoyan Zhu. 2020. Knowledge-Aided Open-Domain Question Answering. ArXiv:2006.05244 [cs].

A Sentence-BERT architecture

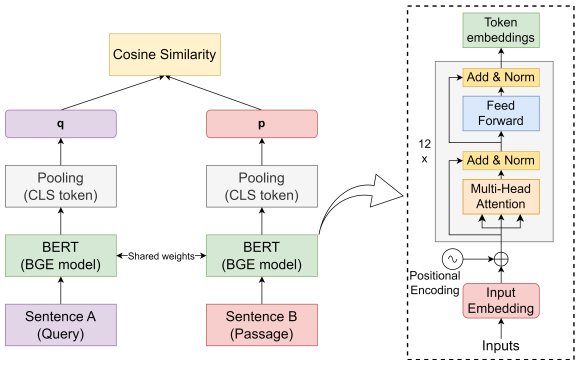


Figure 4: A structure of Sentence-BERT using a Siamese network. Query (Sentence A) and Passage (Sentence B) are encoded separately by two shared-weight BERT-base blocks.

As shown in Figure 4, SBERT uses a Siamese network (Schroff et al., 2015) with two shared-weight BERT-based models. Given the query and passage pairs, SBERT will learn to give higher cosine similarity scores for matched pairs and vice versa. In principle, the embedding model used in SBERT can be any variation of BERT.

In this paper, all layers of the BERT-based model are fine-tuned to ensure complete adaptation to the training set. The *Pooling* layers in SBERT are configured depending on pre-trained BERT to aggregate contextualized word embeddings of query and passage resulting vectors q and p, respectively. Finally, the cosine similarity is then computed between them.

B Prompts for synthetic question generation

The instructional prompt templates used for the Zephyr 7B and Mistral 7B models for generating train and test sets in our case are depicted in Table 4. In both prompts, a role-playing prompting strategy (Li et al., 2023a) was chosen to guide the models' behaviours and responses.

C Graph statistics

An example of a subgraph is shown in Figure 5. Table 5 shows the number of entities in each class and the relations in our KG. It is built by $455,737$ triples including $98,524$ entities, in which there are $5,607$ Passage entities. In $65,282$ objects, each Object can be associated by an average of 5.6 Passages and by at least 1 Passage. Besides, there are $23,634$ Objects linked with MeSH Terms. On average, each Object is connected to 2.7 MeSH Terms while each MeSH Term is referred to by 3.26 Objects. Furthermore, each MeSH Concept is referred to by an average of 2.1 MeSH Term. For the Path entity, on average, each Path is connected by 3.77 Passages, in which the highest number of Passage originating from a Path is 20.

Table 4: Two prompt templates to generate synthetic training and testing sets. The prompt for the train set is inputted to the Zephyr-7B-beta model while the prompt for the test set is used to guide Mistral-7B-OpenOrca.

Prompt for train set
Context information is below. _____ {context_input} _____ Given the context information and no prior knowledge, you are a Teacher/Professor. Your task is to set up {num_questions_per_chunk} questions for an upcoming quiz/examination. The question must be based on the main context. Additionally, the question must have a clear answer indicated in the context information. Finally, return the question with a question mark at the end.
Prompt for test set
Context information is below. _____ {context_input} _____ Given the context information and no prior knowledge, you are a Teacher/Professor. Your task is to set up {num_questions_per_chunk} questions for an upcoming quiz/examination. The questions should be diverse in nature across the document. Restrict the questions to the context information provided.

Table 5: Summary of knowledge graph details

Entity Class	**Count**	**Relation Type**	**N.o triples using relation**
Passage	5607	associated_object	370936
Path	1487	has_path	5607
Object	65282	preferredTerm	64053
MeSH Concept	6901	Concept_prefferedTerm	15141
MeSH Term	19247		
Total	98524	Total	455737

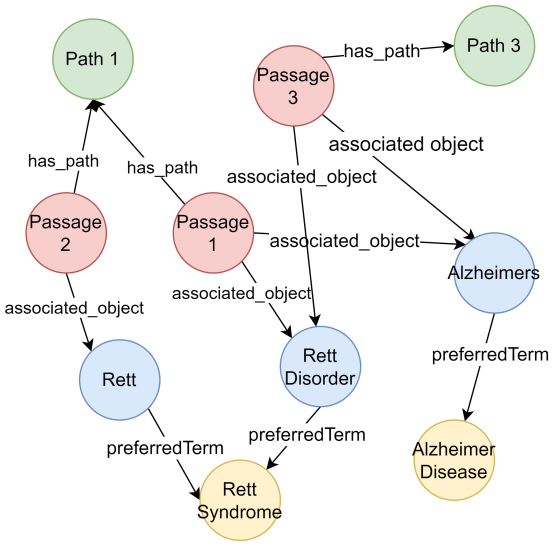


Figure 5: An example of a sub-graph illustrates the relationship between different passages. Intuitively, Passages with more common connections are located closer in the graph.

D Influence of scaling weight α

Figure 6 presents the results of the hybrid retriever's performance employing fine-tuned BGE and TransE embeddings, concerning HitRate@3 across varying α values.

E Comparision to "Retrieve then Re-rank" mechanism

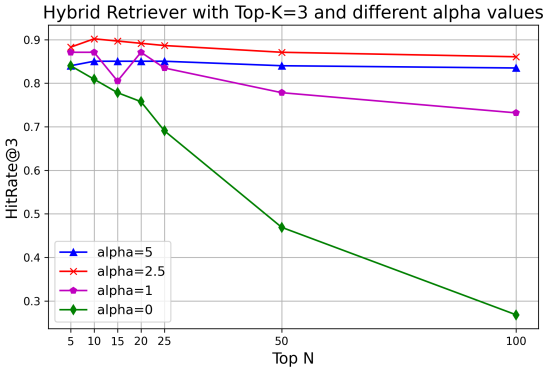
Table 6 shows the comparison between the proposed methodology and retrievers integrating re-ranking models.

Table 6: The performance comparison on the first test set between our hybrid retriever and different combinations of retriever and re-ranker models.

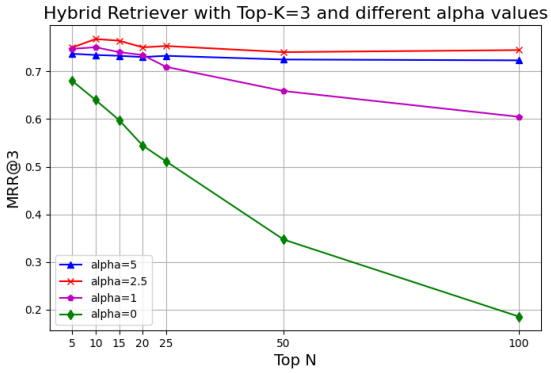
Retriever + Reranker	HR@3	MRR@3
BGE_{ft}	0.8350	0.7164
BGE_{ft} + Stsb-roberta-base	0.3762	0.2689
BGE_{ft} + ms-marco-miniLM-L-2-v2	0.8195	0.6701
BGE_{ft} + BGE-rerank-base	0.9123	0.7439
BGE_{ft} + ms-marco-miniLM-L-6-v2	**0.9175**	**0.8092**
BGE_{ft} + TransE (ours)	0.8917	0.7506

F Examples of system's response

Table 7 illustrates an example of answers from the Zephyr-7B-beta model for a question based on the contexts provided by BGE-only and hybrid retrievers. The answer generated from the hybrid retriever's contexts exhibited greater detail and closer alignment with the golden answer. In contrast, the answer obtained from the BGE retriever's passages contained only half of the information leading to a lower correctness score.



(a) Performance Analysis with HitRate@3



(b) Performance Analysis with MRR@3

Figure 6: Overall Performance Analysis on HitRate@3 and MRR@3 with different alpha values

Table 7: Comparison of answers from Zephyr-7B-beta reader with different retrievers

Question: What is the main reason for revision surgery?
Golden Answer: Infection, infection and infection. We already have antibacterial coatings. However, these do not cover for the 100%. Furthermore, the proximal and distal parts are coated, but the valve is not.
BGE + Zephyr-7B-beta: According to the expert call, the main reason for revision surgery in brain shunts is infection, infection, and infection. **Correctness score:** 0.4096
Hybrid + Zephyr-7B-beta: The main reason for revision surgery in the context provided is infection, infection, and infection. The expert mentions that if there is no native increased pressure, then the shunts get clogged due to pathology from the past such as high protein levels, haemorrhages, or circulating tumour cells. The valve is prone to erythrocyte cells, and the main reason for revision surgery is infection. The expert also mentions that they already have antibacterial coatings, but these do not cover 100% and the proximal and distal parts are coated, but the valve is not. Therefore, infection remains a significant issue in shunt revision surgeries. **Correctness score:** 0.9975