

MrRank: Improving Question Answering Retrieval System through Multi-Result Ranking Model

Danupat Khamnuansin^{†*}, Tawunrat Chalothorn[†], and Ekapol Chuangsuwanich^{*}

[†] Kasikorn Labs Co., Ltd., Kasikorn Business-Technology Group, Thailand

^{*} Department of Computer Engineering, Faculty of Engineering, Chulalongkorn University

danupat.k@kbtg.tech tawunrat.c@kbtg.tech ekapolc@cp.eng.chula.ac.th

Abstract

Large Language Models (LLMs) often struggle with hallucinations and outdated information. To address this, Information Retrieval (IR) systems can be employed to augment LLMs with up-to-date knowledge. However, existing IR techniques contain deficiencies, posing a performance bottleneck. Given the extensive array of IR systems, combining diverse approaches presents a viable strategy. Nevertheless, prior attempts have yielded restricted efficacy. In this work, we propose an approach that leverages learning-to-rank techniques to combine heterogeneous IR systems. We demonstrate the method on two Retrieval Question Answering (ReQA) tasks. Our empirical findings exhibit a significant performance enhancement, outperforming previous approaches and achieving state-of-the-art results on ReQA SQuAD.

1 Introduction

Large Language Models (LLMs) have demonstrated proficiency in comprehending human language across diverse domains. Among these advancements, question answering (QA) systems integrated with LLMs have gained significant attention. Nonetheless, concerns persist regarding the susceptibility of LLMs to hallucinations, potentially leading to inaccuracies in their responses.

To address the aforementioned challenge, the Retrieval Augmented Generation (RAG) (Lewis et al., 2020) is proposed. The RAG system generally consists of two parts: an Information Retrieval (IR) engine which proposes the top k candidate pieces from a document corpus for the query, and the QA engine which is responsible for deriving the final answer from retrieved information.

Although LLMs demonstrate exceptional proficiency in QA tasks, contemporary text embedding techniques employed in IR remain underdeveloped. This deficiency poses a performance bottleneck for

RAG, as shown by the results in the Massive Text Embedding Benchmark (Muennighoff et al., 2023).

To improve retrieval performance, advanced fine-tuning techniques applied to individual retrieval systems (Zhao et al., 2023) or the enhancement of model components (Zhao et al., 2021) emerge as a prominent area of investigation. While fine-tuning approaches have demonstrated efficacy in this domain, they commonly require substantial resources for training, posing a challenge to real-world adoption.

Combining results from multiple IR systems can be quite effective, as demonstrated in the work of Cormack et al. (2009), Reciprocal Rank Fusion (RRF) is a technique that utilizes ranking information to rerank the overall ranking, which has become the industry standard for combining multiple ranking systems. Although the implementation of RRF is straightforward, a notable limitation arises due to its failure to leverage output scores from individual retrievers. Modern techniques have been developed to overcome this limitation; FlexNeuART (Boysov and Nyberg, 2020) employs the Learning to Rank (LTR) framework to amalgamate scores from diverse retrievers into a unified score. However, the application of LTR in the QA retrieval setting requires a modification of the training process to avoid irrelevant samples. Moreover, the exploration and optimization of these techniques within the domain of question answering retrieval settings remain largely unaddressed.

In the work of Liang et al. (2021), a routing method for combining multiple retrievers in the QA setting is introduced. This model employs a routing classifier to select between a neural retriever and a traditional BM25 retriever. However, this method heavily relies on the score generated by BM25 and disregards ranking information from the neural model. Consequently, its performance is limited in modern question answering systems where the neural model typically outperforms BM25.

We propose re-ranking techniques specifically trained to improve question answering retrieval performance. Our investigations involve combining two different types of model architecture (term weighting and neural networks). Additionally, we conducted experiments on two distinct styles of ReQA datasets to demonstrate that combining multiple models using this setup can leverage the advantages of both models through ranking loss objective training. Our findings show that the proposed methodology yields enhanced retrieval performance and ultimately improves the downstream QA task. Our approach outperforms the current state-of-the-art (Zhao et al., 2023) on ReQA SQuAD, surpassing all individual retrieval models, RRF, and the statistical routing strategy, yielding an average enhancement of 13.6% in the mean reciprocal rank (MRR) across datasets.

2 Related Work

Massive Text Embedding Benchmark

Massive Text Embedding Benchmark (MTEB) proposed by Muennighoff et al. (2023) offers comprehensive insights and evaluations across a wide range of modern text embedding models. This benchmark is particularly crucial for understanding the out-of-the-box performance of various models. The evaluation incorporated established text embedding models available on the Hugging Face platform (Wolf et al., 2020). The assessment encompassed retrieval tasks across 15 distinct datasets, each representing diverse domains and dataset characteristics. The evaluation of re-ranking tasks was carried out using four specifically curated re-ranking datasets, providing a robust assessment of performance in real-world applications.

As shown in Table 1, GPT Sentence Embeddings for Semantic Search (SGPT) (Muennighoff, 2022) is one of the top performing models in retrieval tasks. Note that sentence embeddings constructed by simply averaging Global Vectors for Word Representation (GloVe) (Pennington et al., 2014) outperforms BERT (Devlin et al., 2019) and OpenAI’s Ada Similarity. This observation underscores the challenges inherent in retrieval tasks, where mainstream text embedding models still exhibit a deficiency in text retrieval proficiency.

In the context of re-ranking tasks, MPNET (Song et al., 2020) has demonstrated superior performance, highlighting the challenges in achieving a universally best solution in the current landscape.

Notably, a model exhibiting proficiency in retrieval tasks may not necessarily excel in re-ranking tasks.

Model	Retr.	Rerank.
BERT	10.59	43.44
OpenAI Ada Similarity	18.36	49.02
GloVe	21.62	43.29
MPNet	43.81	59.36
SGPT-5.8B-msmarco	50.25	<u>56.56</u>

Table 1: Average of the retrieval and re-ranking metric per task per model on MTEB English subsets.

Retrieval Question Answering

Retrieval question answering (ReQA) tasks are proposed in works such as Ahmad et al. (2019) and Chen et al. (2017). They encompass a type of information retrieval system that takes a question as input and responds with the most relevant sentence containing the answer. This task distinguishes itself from traditional Question Answering (QA), where the paragraph containing the answer is provided upfront, leading to notable achievements in machine learning systems, some of which surpass human performance (Rajpurkar et al., 2016). QA retrieval systems typically rely on either neural or term weighting methods. Recent studies, such as SPARTA (Zhao et al., 2021), exhibit significant improvements in retrieval performance by adhering to the concept of tokens-level matching from traditional IR systems. However, the exploration of QA retrieval systems that incorporate hybrid retrieval mechanisms remains relatively limited.

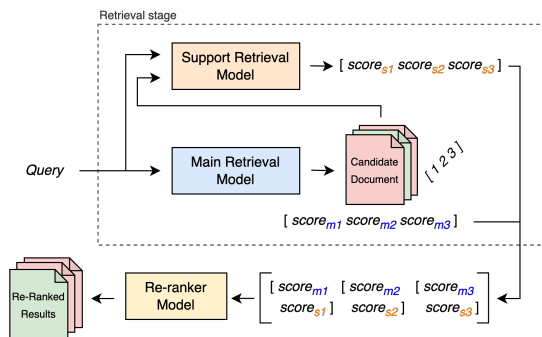
Hybrid Method for Evidence Retrieval for Question Answering

Previous attempts to combine a term weighting model with a neural-based model simply involve routing between retrieval results from each model. This is achieved by training a routing classifier that determines whether to use the traditional IR method (BM25) or the neural-based method (USEQA) based on statistics derived from the training dataset (Liang et al., 2021). The study demonstrates that the hybrid approach outperforms individual retrieval strategies in ReQA SQuAD and ReQA NQ. However, this methodology has limitations in its support for combining multiple neural models, the integration of more than two models, and heavily relies on BM25. These limitations may deter real-world adoption, where there are numerous retrieval models specialized in their own

areas. Notably, USE-QA (Yang et al., 2020), employs fine-tuning techniques to tackle an asymmetric matching task present in QA retrieval. Combining the specialized abilities of these models could lead to better performance.

Thus, we emphasized an approach enhancing model retrieval performance through model-level combination that aims to capitalize on the strengths of multiple model types. Our objective is to present a robust methodology that can effectively combine state-of-the-art or fine-tuned models for better performance improvement.

Figure 1: Our framework for amalgamating multiple ranking information by using the re-ranking approach.



3 Proposed Method

To construct a cohesive framework for merging multiple sets of ranking information, we adopt a re-ranking approach. We have designated a neural-based model as the primary retriever, complemented by BM25 as a supporting model. This choice is based on the observed consistency of the neural-based model, which demonstrates superior performance on R@10. This metric is particularly crucial for assessing the upper limits of performance gain in re-ranking performance. The rankings obtained from the main retriever serve as the initial rankings, subsequently augmented by incorporating scores derived from the support retriever. These concatenated scores are then fed into the re-ranker model to predict the final rankings. The overall process is illustrated in Figure 1.

The frameworks consist of two stages:

1) Retrieval Stage: off-the-shelf retrievers are used in order to generate a candidate pool. We select top candidate retrievers from MTEB benchmarks within the retrieval and re-ranking category. The performance is shown in Table 2.

2) Re-ranking Stage: a re-ranking network is used to construct the final ranking from the candidate pool.

3.1 Retrieval Stage

Our approach focuses on incorporating multiple QA retrieval models. These models have demonstrated significant performance improvements in question answering retrieval tasks compared to base retrieval models. To maximize the applicability of our approach, we have chosen widely recognized and publicly accessible models. In accordance with Muennighoff et al. (2023) results, we also included SGPT-5.8B-msmarco and MPNET. These models represent the current state-of-the-art publicly accessible models in the domains of retrieval and re-ranking tasks, respectively.

Model	MRR	R@1	R@5	R@10
<i>ReQA SQuAD</i>				
BM25	<u>0.670</u>	<u>0.591</u>	0.759	0.814
USE _{qa}	0.665	0.561	<u>0.793</u>	<u>0.854</u>
MPNET _{qa}	0.549	0.399	0.748	0.847
SGPT _{5.8B}	0.783	0.699	0.887	0.926
<i>ReQA NQ</i>				
BM25	0.529	0.378	0.723	0.797
USE _{qa}	0.582	<u>0.447</u>	0.751	0.826
MPNET _{qa}	<u>0.628</u>	0.439	0.902	0.950
SGPT _{5.8B}	0.652	0.528	<u>0.807</u>	<u>0.856</u>

Table 2: The off-the-shelf sentence-level retrieval performance on the ReQA SQuAD and ReQA NQ task.

BM25 Retriever The token statistics for BM25 in the original ReQA paper were computed over the first 10,000 questions of each dataset. This led to significant performance losses, as shown in Table 3. In our implementation of BM25, we have made modifications that include the following key distinctions:

1) The statistics are derived from all documents within the training set, instead of solely utilizing the first 10,000 questions.

2) Our BM25 model is constructed using the `gism` library (Rehurek and Sojka, 2011) with default parameters.

The original ReQA paper provided the rationale behind using only the initial 10,000 questions as to reduce the scoring time. We found that there is no discernable disparity in inference time when comparing the utilization of all question statistics versus using only the first 10,000 statistics for constructing

Model	MRR	R@1	R@5	R@10
<i>ReQA SQuAD</i>				
BM25 _{10k}	0.600	0.523	0.683	0.741
BM25 _{all}	0.670	0.591	0.759	0.814
<i>ReQA NQ</i>				
BM25 _{10k}	0.474	0.327	0.660	0.739
BM25 _{all}	0.529	0.378	0.723	0.797

Table 3: Performances of BM25 on sentence-level ReQA from original paper and our implementation.

BM25. Consequently, we recommend adopting our approach to implement the BM25 version instead, which yields a notable increase in performance. It is worth mentioning that the training and inference duration of our model remains unaffected by these variations in BM25 configurations.

3.2 Re-ranking Stage

To effectively integrate ranking information derived from multiple retrieval systems, we opted for a neural-based architecture with a pairwise learning-to-rank approach. The method is motivated by two primary factors.

1) QA retrieval models are initially pre-trained with a specific focus on enhancing the ranking of question-answer document pairs. Consequently, applying the pointwise method to train the re-ranker model would lead to redundant training.

2) The unavailability of comprehensive ranking in the question-answer retrieval dataset poses limitations on our ability to fully utilize the advantageous attributes of listwise methods.

Architecture The re-ranker model adopts the RankNet (Burgess et al., 2005) architecture, which comprises of two layers of Siamese feed-forward neural network. (Koch et al., 2015) The network consists of 10 hidden units with leaky ReLU (Maas et al., 2013) and a single sigmoid output node.

Training Hyperparameters We optimized the re-ranker network using the Adam optimizer, (Kingma and Ba, 2015) using a learning rate of $1e - 3$, 1024 batch size, and 100 epochs for every model setting.

Data Preparation The training features are derived from the top 64 ranking scores of the best-performing retrieval model during the retrieval stage (main retriever). Subsequently, the scores from support models are computed and concatenated to form a set of re-ranking features. For the term weighting model, we utilize the scoring output directly from the model. For the neural-based

model, the score is generated by calculating the cosine similarity between the encoded query and the candidate document vectors.

We chose a number of re-ranked documents (k) equal to 64 to facilitate a direct comparison with the routing method. We experimented on how the value of k affects the performance and inference time. The findings indicate that augmenting the value of k beyond 16 yields only marginal improvements in MRR and is not worth trading off with prediction speed. Detailed experimental outcomes are presented in Figure 3 and Figure 4, as outlined in the Appendix.

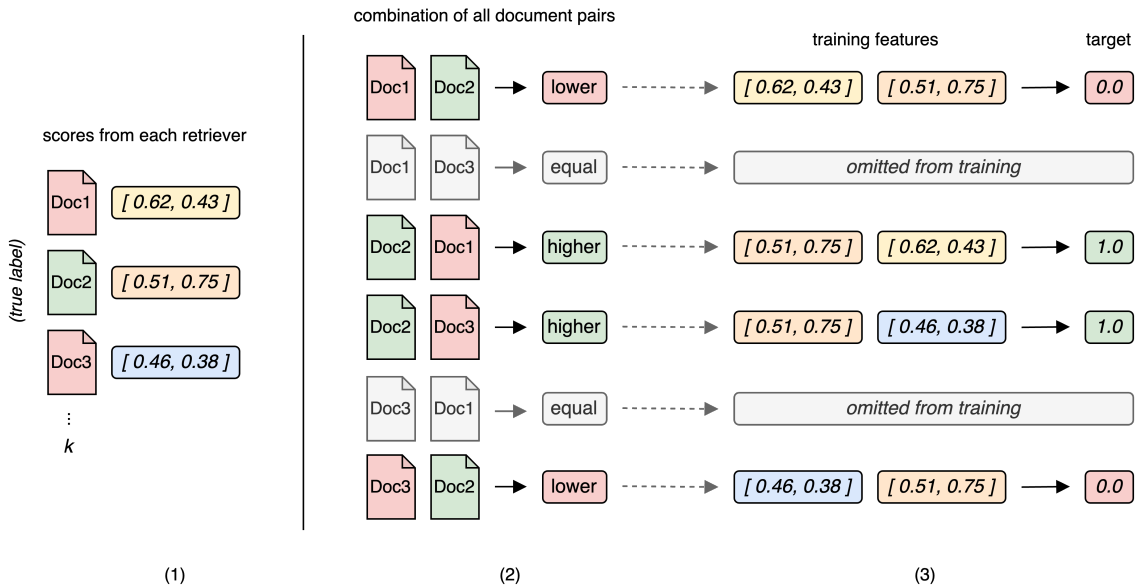
Training Objective The pairwise LTR framework objective involves training the model to produce a value closer to one when it determines the ranking of a particular document to be higher than another. Conversely, it generates a value closer to zero for the opposite scenario. In the event of a tied situation, the conventional RankNet framework designates a target value of 0.5. Since the ReQA dataset has a lot of non-target documents, selecting random pairs will usually result in a tie. To customize the training objective for ReQA tasks, our approach prioritizes sentences containing the correct answer over those that do not. We have modified the conventional RankNet learning-to-rank algorithm by excluding the training sample with a target value of 0.5. The reduction in unnecessary training pairs resulted in a significant decrease in training time while simultaneously enhancing performance.

Inference Phase During the inference phase, the re-ranker model generates ranking information for each candidate document pair to determine which document should rank higher $\{(D_i > D_j), (D_j > D_k), (D_i > D_k)\}$. The final ranking is derived using a topological sorting algorithm $\{D_i > D_j > D_k\}$. In cases where the outcome fails to yield a conclusive ranking, the original results from the main retriever are used as the final ranking. All operations are performed using the networkx library (Hagberg et al., 2008).

4 Results

We assess different retrieval models, including BM25, USE-QA, SGPT-5.8B-msmarco and MPNET-QA (Song et al., 2020). Our baselines include average scoring, thresholding, and Rank Reciprocal Fusion (RRF), a technique commonly employed within modern multi-retriever systems.

Figure 2: Data preparation steps include: 1) retrieving scores for each document candidate, 2) performing permutations on each pair of documents and filtering out tied pairs, and 3) creating negative and positive samples.



We compare our proposed approach mainly on the ReQA SQuAD and ReQA NQ, aligning with the original ReQA paper. Additionally, we conduct experiments on seven question answering datasets from MTEB to assess its effectiveness in contemporary retrieval tasks.

The Stanford Question Answering Dataset (SQuAD) (Rajpurkar et al., 2016) consists of questions on a set of Wikipedia articles, whereas Natural Questions (NQ) (Kwiatkowski et al., 2019) involves user questions issued to Google search, with corresponding answers sourced from Wikipedia by annotators. These two datasets represent distinct characteristics related to lexical overlap between questions and contexts, with SQuAD representing a low-overlap scenario while NQ exhibits the opposite. The publicly available version of both datasets consists of two partitions: a training set and a development set. The number of questions and documents is shown in Table 4.

The development set was used as the evaluation set for both datasets. We focused on the sentence-level retrieval task, following the method outlined in the original ReQA paper. However, our implementation of BM25 included the complete vocabulary of the training documents, deviating slightly from the original ReQA paper which only used the first 10,000 questions. This difference resulted in better performance for our BM25 implementation. See "BM25 Retriever" in Section 3.1 for additional details.

Dataset	Question	Document
ReQA _{SQuAD} (train)	87,599	91,707
ReQA _{SQuAD} (dev)	11,426	10,250
ReQA _{NQ} (train)	74,097	239,013
ReQA _{NQ} (dev)	1,772	7,020

Table 4: The number of questions and candidate sentences on the ReQA SQuAD and ReQA NQ datasets.

We use the Mean Reciprocal Rank (MRR) and Recall at k (R@k) as evaluation metrics. The experiments were replicated five times, and the mean value is reported. We performed significant testing between routing and our method, which showed significant improvements in all settings (one sample t-test, with $p < 0.05$). See Appendix for summary statistics of the results.

Table 5 presents the performance of both individual and combined models. Our model demonstrates better performance across all datasets and evaluation metrics, particularly in ReQA SQuAD where BM25’s performance is comparable to those of the neural-based models. Our approach exhibits a significant improvement compared to individual retrieval methods, RRF and routing strategies. Our model (SGPT+BM25) outperforms MPNET-QA by 48.5%, USE-QA by 22.6%, BM25 by 21.6%, SGPT-5.8B-msmarco by 4.1%, and routing techniques on the same model combination by 7.0%.

On the ReQA NQ dataset, the performance of BM25 is significantly inferior compared to the

	Model	Type	MRR	R@1	R@5	R@10
<i>ReQA SQuAD</i>						
	BM25	Single	0.670	0.591	0.759	0.814
	USE _{qa}	Single	0.665	0.561	0.793	0.854
	MPNET _{qa}	Single	0.549	0.399	0.748	0.847
	SGPT _{5.8B}	Single	0.783	0.699	0.887	0.926
	RRF _{use} (USE _{qa} +BM25)	Hybrid	0.721	0.624	0.839	0.895
	Routing _{use} (USE _{qa} +BM25)	Hybrid	0.705	0.626	0.796	0.851
	Ours _{use} (USE _{qa} +BM25)	Hybrid	0.750	0.663	0.860	0.902
	RRF _{mp} (MPNET _{qa} +BM25)	Hybrid	0.673	0.546	0.827	0.896
	Routing _{mp} (MPNET _{qa} +BM25)	Hybrid	0.709	0.623	0.811	0.869
	Ours _{mp} (MPNET _{qa} +BM25)	Hybrid	0.748	0.658	0.856	0.909
	RRF _{sgpt} (SGPT _{5.8B} +BM25)	Hybrid	0.754	0.664	0.864	0.914
	Routing _{sgpt} (SGPT _{5.8B} +BM25)	Hybrid	0.762	0.686	0.852	0.897
	Ours _{sgpt} (SGPT _{5.8B} +BM25)	Hybrid	0.815	0.736	0.913	0.945
<i>ReQA NQ</i>						
	BM25	Single	0.529	0.378	0.723	0.797
	USE _{qa}	Single	0.582	0.447	0.751	0.826
	MPNET _{qa}	Single	0.628	0.439	0.902	0.950
	SGPT _{5.8B}	Single	0.652	0.528	0.807	0.856
	RRF _{use} (USE _{qa} +BM25)	Hybrid	0.603	0.450	0.802	0.874
	Routing _{use} (USE _{qa} +BM25)	Hybrid	0.552	0.396	0.758	0.831
	Ours _{use} (USE _{qa} +BM25)	Hybrid	0.590	0.452	0.765	0.832
	RRF _{mp} (MPNET _{qa} +BM25)	Hybrid	0.617	0.449	0.843	0.897
	Routing _{mp} (MPNET _{qa} +BM25)	Hybrid	0.638	0.467	0.873	0.931
	Ours _{mp} (MPNET _{qa} +BM25)	Hybrid	0.642	0.457	0.908	0.951
	RRF _{sgpt} (SGPT _{5.8B} +BM25)	Hybrid	0.601	0.443	0.791	0.872
	Routing _{sgpt} (SGPT _{5.8B} +BM25)	Hybrid	0.600	0.440	0.813	0.876
	Ours _{sgpt} (SGPT _{5.8B} +BM25)	Hybrid	0.655	0.529	0.816	0.862

Table 5: Model performances on the sentence-level retrieval of ReQA SQuAD and ReQA NQ task. The routing model refers to the Hybrid (BM25) method described in the original paper.

neural-based systems. Our approach demonstrates the capability to effectively integrate ranking information, resulting in an average performance uplift of 2.6% from individual neural models. This highlights the efficacy of our approach, even in situations where less potent models are combined with their more superior counterparts.

Common Ensemble Techniques

We assess standard ensemble techniques including

- 1) Average scoring, re-rank outputs based on normalized average scores from each retriever.
- 2) Thresholding with a predefined criterion set at 0.5, routing between main and support retriever.

Our results indicate that common ensemble techniques demonstrate inferior performance compared to our approach across all settings, as shown in Table 6.

	Model	MRR	R@1	R@10
<i>ReQA SQuAD</i>				
	Avg (USE+BM25)	0.687	0.595	0.869
	Avg (MPNET+BM25)	0.687	0.599	0.859
	Avg (SGPT+BM25)	0.696	0.598	0.901
	Thr (USE+BM25)	0.691	0.594	0.868
	Thr (MPNET+BM25)	0.571	0.424	0.861
	Thr (SGPT+BM25)	0.788	0.707	0.927
<i>ReQA NQ</i>				
	Avg (USE+BM25)	0.345	0.198	0.691
	Avg (MPNET+BM25)	0.357	0.209	0.720
	Avg (SGPT+BM25)	0.346	0.190	0.720
	Thr (USE+BM25)	0.595	0.455	0.846
	Thr (MPNET+BM25)	0.630	0.441	0.949
	Thr (SGPT+BM25)	0.650	0.528	0.853

Table 6: Retrieval performances of common ensemble techniques. *Avg* and *Thr* denote average scoring and thresholding, respectively.

Neural Model Combinations

Our experiments were assessed within the framework of term weighting-neural combination in direct comparison with the routing method. We also included the neural-neural combination experiments. The results demonstrate the efficacy of our approach. The MPNET-QA and SGPT-5.8B-msmarco combination demonstrated notable improvements, surpassing the best performer in the main experiments on ReQA NQ. This is consistent with our prior observations that the fusion of competitive models yields greater performance gains. The results of the model combination are shown in Table 7.

Model	MRR	R@1	R@10
<i>ReQA SQuAD</i>			
USE _{qa} + SGPT	0.784	0.704	0.920
MPNET _{qa} + SGPT	0.786	0.696	0.939
SGPT + USE _{qa}	0.793	0.710	0.934
MPNET _{qa} + USE _{qa}	0.741	0.634	0.924
USE _{qa} + MPNET _{qa}	0.742	0.639	0.915
SGPT + MPNET _{qa}	0.799	0.710	0.948
<i>ReQA NQ</i>			
USE _{qa} + SGPT	0.660	0.536	0.867
MPNET _{qa} + SGPT	0.723	0.569	0.963
SGPT + USE _{qa}	0.665	0.538	0.876
MPNET _{qa} + USE _{qa}	0.712	0.551	0.961
USE _{qa} + MPNET _{qa}	0.698	0.553	0.920
SGPT + MPNET _{qa}	0.714	0.576	0.922

Table 7: The performances of neural-neural models.

Overall Question Answering Performance

In this experiment we investigated how enhancing retrieval performance translates to the overall question answering system performance. We integrated retrieval systems with the BERT reader model and found that an increase in retrieval performance benefits the overall QA system performance in every setting. We reported answer quality using two metrics, including the Exact Match score (EM), where the answer is strictly required to match the label, along with the less strict ROUGE-L score (Lin, 2004). The results show a strong correlation coefficient of 0.865 between R@1 and ROUGE-L scores, confirming that enhancements in retrieval performance positively impact overall QA performance. However, there is a substantial gap between the best retrieval method and the oracle retriever, showing the potential gain for further improvements. The results are presented in Table 8.

Model	ROUGE-L	EM
<i>SQuAD</i>		
BM25	0.519	0.392
MPNET	0.370	0.264
SGPT	0.611	0.458
Ours (MPNET+BM25)	0.573	0.431
Ours (SGPT+BM25)	0.636	0.481
Ours (SGPT+MPNET)	0.618	0.466
Ours (Trio*)	0.641	0.485
Oracle Retriever	0.848	0.651
<i>NQ</i>		
BM25	0.340	0.235
MPNET	0.392	0.259
SGPT	0.450	0.311
Ours (MPNET+BM25)	0.403	0.271
Ours (SGPT+BM25)	0.450	0.310
Ours (SGPT+MPNET)	0.488	0.339
Ours (Trio*)	0.488	0.340
Oracle Retriever	0.781	0.588

Table 8: Overall performances on SQuAD and NQ. *Trio model comprises SGPT, MPNET, and BM25.

Effect of Sample Mining Technique

We trained the the combination model of BM25 and SGPT-5.8B-msmarco using the original pairwise learning-to-rank approach, and compared it to our modified version which includes sample mining. Table 9 shows that by reducing unnecessary training pairs, there is a substantial 97.0% reduction in training samples. At the same time, it yields an average 0.6% marginally improvement in performance across datasets.

Model	MRR	Training Pair
<i>ReQA SQuAD</i>		
Ours _{sgpt}	0.8146 ±0.0004	5,212,691
-mining _{sgpt}	0.8070 ±0.0012	176,599,584
<i>ReQA NQ</i>		
Ours _{sgpt}	0.6554 ±0.0007	4,609,808
-mining _{sgpt}	0.6535 ±0.0005	149,379,552

Table 9: Differences in performance and training pair when unnecessary samples are omitted.

Low-resource Setting

To emulate a low-resource scenario, we ran an experiment using only 10% of the training data. The findings demonstrate that the method can work even in scenarios with limited data. This suggests that our method also works on a smaller training size, as shown in Table 10.

Model	Data size	MRR	R@1
<i>ReQA SQuAD</i>			
Ours _{use}	100%	0.750	0.663
Ours _{use}	10%	0.749	0.661
Ours _{mp}	100%	0.748	0.658
Ours _{mp}	10%	0.744	0.652
Ours _{sgpt}	100%	0.815	0.736
Ours _{sgpt}	10%	0.815	0.737
<i>ReQA NQ</i>			
Ours _{use}	100%	0.590	0.452
Ours _{use}	10%	0.592	0.454
Ours _{mp}	100%	0.642	0.457
Ours _{mp}	10%	0.641	0.456
Ours _{sgpt}	100%	0.655	0.529
Ours _{sgpt}	10%	0.656	0.529

Table 10: The performances of hybrid models (BM25-neural) on low-resource datasets.

Combining More Than Two Models

Our framework also supports extensions beyond two models. We conducted experiments to combine three retrieval models. The results indicate that combining a model with a different specialization leads to a substantial performance boost, as demonstrated by the combination of SGPT-5.8B-msmarco, MPNET-QA, and BM25. This combination surpasses the current state-of-the-art in ReQA SQuAD, yielding an improvement from 0.801 to 0.823 on the MRR metric (Zhao et al., 2023). The results are shown in Table 11.

Model	MRR	R@1	R@10
<i>ReQA SQuAD</i>			
USE _{mp+bm25}	0.784	0.697	0.926
USE _{sgpt+bm25}	0.807	0.732	0.930
MPNET _{use+bm25}	0.786	0.697	0.934
MPNET _{sgpt+bm25}	0.815	0.734	0.946
SGPT _{use+bm25}	0.819	0.741	0.948
SGPT _{mp+bm25}	0.823	0.742	0.957
<i>ReQA NQ</i>			
USE _{mp+bm25}	0.700	0.555	0.920
USE _{sgpt+bm25}	0.663	0.537	0.867
MPNET _{use+bm25}	0.715	0.556	0.961
MPNET _{sgpt+bm25}	0.729	0.576	0.964
SGPT _{use+bm25}	0.670	0.545	0.876
SGPT _{mp+bm25}	0.714	0.576	0.922

Table 11: Retrieval performances of trio models.

Performance on Modern QA Datasets

We performed experiments using QA datasets from MTEB to assess the efficacy of our methods in modern retrieval tasks. We incorporated MPNET-QA and SGPT-125M-msmarco, utilizing the development set for training and the test set for evaluation. The results presented in Table 12 show the effectiveness of our approach across datasets.

Dataset	MPNET	SGPT	Ours
DBPedia	0.656	0.264	0.669
FEVER	0.542	0.151	0.546
FiQA2018	0.560	0.120	0.560
HotpotQA	0.622	0.234	0.634
MSMARCO	0.352	0.425	0.663
NFCorpus	0.539	0.130	0.539
QuoraRetrieval	0.868	0.797	0.877

Table 12: Model performances on question answering datasets from MTEB, reported in MRR.

End-to-end Evaluation on RAG

We conducted Retrieval-Augmented Generation (RAG) experiments to assess the end-to-end performance of our method using the mistral-7b-instruct-v0.2 (Jiang et al., 2023) as a reader model. The outcomes were evaluated based on METEOR (Banerjee and Lavie, 2005), BERT-F1 (Zhang et al., 2020), and ROUGE-L scores. Our systems demonstrate performance improvements, as shown in Table 13.

Model	ME	BERT	RO-L
<i>SQuAD</i>			
BM25	0.298	0.722	0.194
MPNET	0.225	0.700	0.143
SGPT	0.346	0.737	0.225
MPNET + BM25	0.328	0.731	0.213
SGPT + BM25	0.359	0.741	0.233
SGPT + MPNET	0.348	0.738	0.227
Trio	0.362	0.742	0.236
Oracle Retriever	0.464	0.770	0.298
<i>NQ</i>			
BM25	0.251	0.697	0.155
MPNET	0.286	0.706	0.173
SGPT	0.325	0.719	0.200
MPNET + BM25	0.289	0.707	0.175
SGPT + BM25	0.325	0.719	0.201
SGPT + MPNET	0.344	0.724	0.212
Trio	0.345	0.724	0.213
Oracle Retriever	0.362	0.742	0.236

Table 13: The performances of models in RAG tasks.

Multilingual Settings

We conducted a non-English language experiment to assess the performance of our method when utilizing a multilingual model setup. We performed the experiment in the context of Thai language, utilizing the iapp-wiki-qa-squad datasets (Viriyayudhakorn and Polpanumas, 2021) which adhere to the SQuAD styles. Our modifications to support the Thai language include:

1) Changing USE-QA, which only supports English language, to Multilingual Universal Sentence Encoder Question Answering (mUSE-QA).

2) Substituting the NLTK (Bird and Loper, 2004) tokenizer with the PyThaiNLP (Phatthiyaphaibun et al., 2023) newmm tokenizer.

3) Using BM25 as the main retriever instead of the neural model due to a significant gap in performance on the R@10 metric.

The result shows that our method outperforms routing techniques. Notably, routing strategies only selected the ranking from the BM25 models, yielding a 0% uplift in performance. The results are shown in Table 14.

Model	MRR	R@1	R@10
<i>ReQA iAppTH</i>			
BM25	0.828	0.742	0.951
mUSE _{qa}	0.636	0.533	0.811
Routing _{muse}	0.828	0.742	0.951
BM25 _{muse}	0.835	0.748	0.960

Table 14: Model performances on multilingual settings.

5 Conclusion

In this paper, we propose an effective method to combine term weighting and neural-based ReQA systems. Our approach involves training re-ranker neural networks using a modified pairwise re-ranking objective. Through empirical evaluations, the combined system exhibits a significant performance improvement over other combining strategies, particularly when confronted with significant performance disparities among the individual models. It is noteworthy that this technique is not restricted to a specific type or quantity of retrieval models. Our method focuses on improving the QA system by enhancing the retrieval which we identified as a bottleneck. Further investigation into improving the overall performance by directly optimizing the reader along with the retrieval model might improve the overall pipeline.

Limitations

In our work, we have focused on implementing our methodology within the context of the retrieval question answering task. Nonetheless, it is reasonable to anticipate that the application of our techniques to a broader retrieval task would yield favorable results. Our methodology necessitated the selection of a main retriever. This is helpful in reducing the computing load of the supporting retriever, yet it can introduce a cap on the final performance. In the future, we may try to eliminate the need for main retrieval model selection. We also focus our experiment on techniques that do not require full-weight updates. Nevertheless, our approach can complement other full-weight update fine-tuning techniques to further enhance performance. Lastly, our model’s computational cost scales with the number of re-ranking indexes fed through the re-ranker (k), as the process requires the prediction of k^2 pairs of attributes to determine the final ranking. This aspect may present challenges when deploying the model in situations with a tight compute budget.

Acknowledgements

This work is under the joint research initiative between KASIKORN Business-Technology Group (KBTG) and the Faculty of Engineering, Chulalongkorn University. We would like to thank KASIKORNBANK PUBLIC COMPANY LIMITED (KBank) and KBTG for supporting the research and providing computational resources to conduct the experiments.

References

- Amin Ahmad, Noah Constant, Yinfei Yang, and Daniel Cer. 2019. *ReQA: An evaluation for end-to-end answer retrieval models*. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 137–146, Hong Kong, China. Association for Computational Linguistics.
- Satanjeev Banerjee and Alon Lavie. 2005. *METEOR: An automatic metric for MT evaluation with improved correlation with human judgments*. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.
- Steven Bird and Edward Loper. 2004. *NLTK: The natural language toolkit*. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages

- 214–217, Barcelona, Spain. Association for Computational Linguistics.
- Leonid Boytsov and Eric Nyberg. 2020. [Flexible retrieval with NMSLIB and FlexNeuART](#). In *Proceedings of Second Workshop for NLP Open Source Software (NLP-OSS)*, pages 32–43, Online. Association for Computational Linguistics.
- Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. 2005. [Learning to rank using gradient descent](#). In *Proceedings of the 22nd International Conference on Machine Learning, ICML '05*, page 89–96, New York, NY, USA. Association for Computing Machinery.
- Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. [Reading Wikipedia to answer open-domain questions](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada. Association for Computational Linguistics.
- Gordon V. Cormack, Charles L A Clarke, and Stefan Buettcher. 2009. [Reciprocal rank fusion outperforms condorcet and individual rank learning methods](#). In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '09*, page 758–759, New York, NY, USA. Association for Computing Machinery.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. 2008. Exploring network structure, dynamics, and function using networkx. In *Proceedings of the 7th Python in Science Conference*, pages 11 – 15, Pasadena, CA USA.
- Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.
- Diederik P. Kingma and Jimmy Ba. 2015. [Adam: A method for stochastic optimization](#). In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.
- Gregory Koch, Richard Zemel, Ruslan Salakhutdinov, et al. 2015. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2(1). Lille.
- Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. [Natural questions: A benchmark for question answering research](#). *Transactions of the Association for Computational Linguistics*, 7:452–466.
- 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. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS'20*, Red Hook, NY, USA. Curran Associates Inc.
- Zhengzhong Liang, Yiyun Zhao, and Mihai Surdeanu. 2021. [Using the hammer only on nails: A hybrid method for representation-based evidence retrieval for question answering](#). In *Advances in Information Retrieval: 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28 – April 1, 2021, Proceedings, Part I*, page 327–341, Berlin, Heidelberg. Springer-Verlag.
- Chin-Yew Lin. 2004. [ROUGE: A package for automatic evaluation of summaries](#). In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.
- Andrew L Maas, Awni Y Hannun, Andrew Y Ng, et al. 2013. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30(1), page 3. Atlanta, Georgia, USA.
- Niklas Muennighoff. 2022. [Sgpt: Gpt sentence embeddings for semantic search](#).
- Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. 2023. [MTEB: Massive text embedding benchmark](#). In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2014–2037, Dubrovnik, Croatia. Association for Computational Linguistics.
- Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. [GloVe: Global vectors for word representation](#). In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.
- Wannaphong Phatthiyaphaibun, Korakot Chaovavanich, Charin Polpanumas, Arthit Suriyawongkul, Lalita Lowphansirikul, Pattarawat Chormai, Peerat Limkonchotiwat, Thanathip Suntornthip, and Can Udomcharoenchaikit. 2023. [PyThaiNLP: Thai natural language processing in python](#). In *Proceedings of the 3rd Workshop for Natural Language Processing Open Source Software (NLP-OSS 2023)*, pages 25–36, Singapore, Singapore. Empirical Methods in Natural Language Processing.

- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. [SQuAD: 100,000+ questions for machine comprehension of text](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics. 565–575, Online. Association for Computational Linguistics.
- Radim Rehurek and Petr Sojka. 2011. Gensim–python framework for vector space modelling. *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic*, 3(2).
- Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. In *Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS’20*, Red Hook, NY, USA. Curran Associates Inc.
- Kobkritt Viriyayudhakorn and Charin Polpanumas. 2021. [iapp_wiki_qa_squad](#).
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. [Transformers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.
- Yinfei Yang, Daniel Cer, Amin Ahmad, Mandy Guo, Jax Law, Noah Constant, Gustavo Hernandez Abrego, Steve Yuan, Chris Tar, Yun-hsuan Sung, Brian Strope, and Ray Kurzweil. 2020. [Multilingual universal sentence encoder for semantic retrieval](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 87–94, Online. Association for Computational Linguistics.
- Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. [Bertscore: Evaluating text generation with BERT](#). In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.
- Bo Zhao, Jun Bai, Chen Li, Jianfei Zhang, Wenge Rong, Yuanxin Ouyang, and Zhang Xiong. 2023. [Enhancing biomedical reqa with adversarial hard in-batch negative samples](#). *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 20(5):2933–2944.
- Tiancheng Zhao, Xiaopeng Lu, and Kyusong Lee. 2021. [SPARTA: Efficient open-domain question answering via sparse transformer matching retrieval](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages

A Effect of the Amount of Re-ranked Documents

We conducted an experiment to study the effect of the number of re-ranked documents on the retrieval performance (Figure 3) and runtime (Figure 4). The processing was conducted on a standalone machine featuring Intel(R) Core(TM) i7-7700 CPU @ 3.60GHz, with no GPU acceleration.

To measure the latency, we ran our trio model with $k = 16$. The re-ranker network adds a latency of 0.0133 seconds per query on average across datasets.

Figure 3: Relationship between the number of re-ranked documents (k) and retrieval performance (MRR).

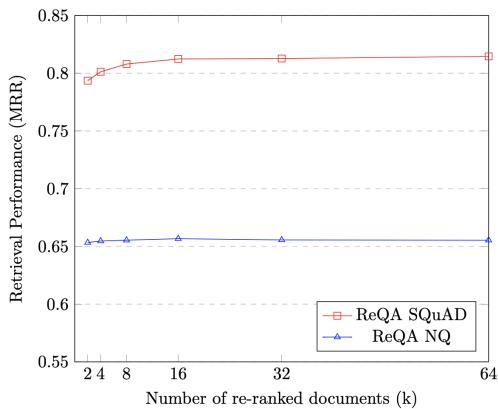
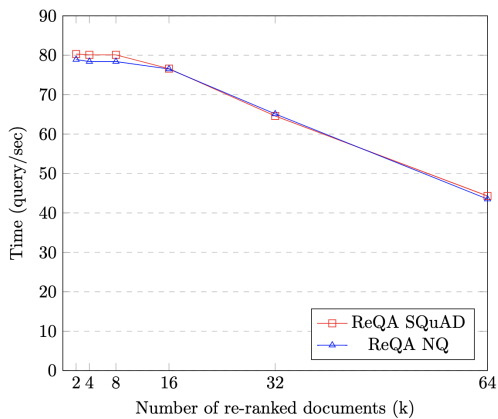


Figure 4: Relationship between the number of re-ranked documents (k) and inference time (query/sec).



B Correlation Between Retrieval and Overall QA Performance

We assess the correlation between retrieval and overall QA performance. The findings reveal a strong correlation across metrics, suggesting that improvements in retrieval performance have a positive influence on overall QA performance.

Metric	MRR	R@1
<i>SQuAD</i>		
ROUGE-L	0.8225	0.8392
EM	0.8182	0.8385
<i>NQ</i>		
ROUGE-L	0.8806	0.8909
EM	0.8430	0.8751

Table 15: Correlation coefficient between retrieval and overall question answering performance.

Model	MRR	Recall@1	Recall@5	Recall@10
<i>ReQA SQuAD</i>				
Ours _{use}	0.7503 ±0.0002	0.6629 ±0.0002	0.8598 ±0.0006	0.9015 ±0.0004
Ours _{mpnet}	0.7476 ±0.0004	0.6576 ±0.0008	0.8561 ±0.0004	0.9090 ±0.0003
Ours _{sgpt}	0.8146 ±0.0004	0.7364 ±0.0005	0.9131 ±0.0006	0.9451 ±0.0005
<i>ReQA NQ</i>				
Ours _{use}	0.5902 ±0.0012	0.4518 ±0.0022	0.7646 ±0.0005	0.8319 ±0.0008
Ours _{mpnet}	0.6422 ±0.0014	0.4568 ±0.0026	0.9077 ±0.0008	0.9510 ±0.0006
Ours _{sgpt}	0.6554 ±0.0007	0.5292 ±0.0011	0.8160 ±0.0006	0.8621 ±0.0019

Table 16: Summary statistics for experimental results on ReQA SQuAD and ReQA NQ.

Query	Routing	Ours
What two artists came out with Coldplay during the half-time show?	On December 3, the league confirmed that the show would be headlined by the British rock group Coldplay.	The Super Bowl 50 halftime show was headlined by the British rock group Coldplay with special guest performers Beyoncé and Bruno Mars, who headlined the Super Bowl XLVII and Super Bowl XLVIII halftime shows, respectively.
What type of relationships do enthusiastic teachers cause?	Students who had enthusiastic teachers tend to rate them higher than teachers who didn't show much enthusiasm for the course materials.	Enthusiastic teachers are particularly good at creating beneficial relations with their students.
When did Luther return to Wittenberg?	When the town council asked Luther to return, he decided it was his duty to act.	Luther secretly returned to Wittenberg on 6 March 1522.
What is the mortality rate of pneumonic plague?	Septicemic plague is the least common of the three forms, with a mortality rate near 100%.	Pneumonic plague has a mortality rate of 90 to 95 percent.
What was the population Jacksonville city as of 2010?	Jacksonville has Florida's largest Filipino American community, with 25,033 in the metropolitan area as of the 2010 Census.	Jacksonville is the principal city in the Jacksonville metropolitan area, with a population of 1,345,596 in 2010.
What was the result of the 1967 referendum?	However, as a result of the referendum in France and the referendum in the Netherlands, the 2004 Treaty establishing a Constitution for Europe never came into force.	When a consolidation referendum was held in 1967, voters approved the plan.

Table 17: Comparative analysis between Routing and our model reveals distinct characteristics. The outcomes obtained from the Routing technique demonstrate the limitations inherent in the BM25 model, as it heavily relies on the presence of overlapping evidence while inadequately addressing the critical aspect of question-answer alignment. Conversely, our model exhibits a robust approach by incorporating ranking information from the neural model.