Advanced Passage Retrieval with Lexical and Semantic Matching

Investigating three passage retrieval approaches with the SQuAD 1.1 dataset

Nguyen Thi Linh
20210731
School of Computing, KAIST
nguyenlinh@kaist.ac.kr

Kyaw Ye Thu
20220929
School of Computing, KAIST
kyawyethu@kaist.ac.kr

Shubhangi Kumar
20210768
School of Computing, KAIST
shubh.kumar@kaist.ac.kr

Khadija Rajabova
20210898
School of Computing, KAIST
r.khadija04@kaist.ac.kr

Nguyen Ngoc Mai
20210731
School of Computing
mainguyen@kaist.ac.kr

Abstract—**Knowledge retrieval is incorporated into search engines and chatbot dialogue generation as a solution to mitigate hallucination issues, which refers to when a model generates responses that include information not present in the input, potentially leading to inaccuracies or fictional content. This research will explore the SQuAD1.1 dataset in performing passage retrieval, providing top-k context answers as output from the question input. The goal of the research is to investigate various models that can outperform our benchmark, BM25, a renowned ranking algorithm, on the SQuAD1.1 dataset. Our result suggests that using a bi-encoder with a fine-tuned cross-encoder not only exhibits a better performance than BM25 and but also achieves the highest accuracy among our implemented models.**

I. Introduction

We implemented the passage retrieval task by extracting pertinent information from query inputs and generating the top-k passages from our dataset as the output based on their relevance to a given query. In this work, we will examine the performance of BM25 as our benchmark model. Despite its decent enough performance, BM25 still has many limitations, which will be identified in this project. Then, we will explore other models that can overcome such constraints to achieve a better accuracy. It is suggested in academia that a hybrid model tends to perform better than a standalone model. Hence, we will be examining hybrid models such as DPR (Deep Passage Retrieval) with BM25 and bi-encoder cross-encoder architecture with BM25 to determine the model that gives the highest accuracy with the SQuAD1.1 dataset.

II. Analysis of the Dataset (SQuAD1.1)

The Stanford Question Answering Dataset (SQuAD) is a dataset for comprehensive training and testing of reading comprehension models. Notably, over 90% of the questions in SQuAD have concise answers.

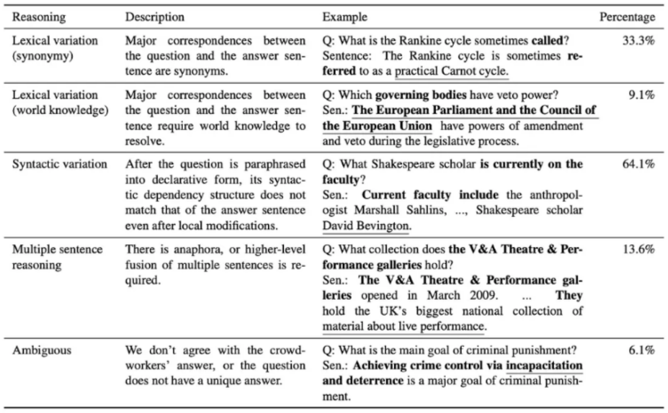
The SQuAD1.1 dataset is intended exclusively for the assessment and development of reading comprehension skills. It consists of a set of crowdsourced questions, all of which are derived from a wide variety of Wikipedia articles. This dataset is special because every question is written so that the answer may be taken straight out of a particular portion of text in the matching reading passage. The dataset also has an interesting feature where some questions are purposefully made unanswerable. This feature adds another level of complexity to the dataset and tests the models' capacity to identify instances in which the text does not include enough information.

In terms of dataset analysis, SQuAD is characterized by diverse categories. Each answer falls into one of several categories: "date," "other numeric," "person," "location," "other entity," "common noun phrase," "adjective phrase," "verb phrase," "clause," or "other".

In addition, the developers carefully selected questions from the development set and manually categorized them according to the kind of logic needed to provide a response. This categorization sheds light on the mental operations required to interpret the questions.

The analysis also includes a measure of syntactic divergence. This entails evaluating how a query and the statement that answers it differ structurally. The creators came up with a metric that determines how complex a question is by counting the number of modifications required to turn it into a sentence that contains the answer. However, it should be known that the response to a certain question can be found in several documents, which would complicate the dataset.

The data fields in SQuAD are organized as follows:



Reasoning	Description	Example	Percentage
Lexical variation (synonymy)	Major correspondences between the question and the answer sentence are synonyms.	Q: What is the Rankine cycle sometimes **called**? Sentence: The Rankine cycle is sometimes **referred** to as a practical Carnot cycle.	33.3%
Lexical variation (world knowledge)	Major correspondences between the question and the answer sentence require world knowledge to resolve.	Q: Which **governing bodies** have veto power? Sen.: **The European Parliament and the Council of the European Union** have powers of amendment and veto during the legislative process.	9.1%
Syntactic variation	After the question is paraphrased into declarative form, its syntactic dependency structure does not match that of the answer sentence even after local modifications.	Q: What Shakespeare scholar **is currently on the faculty**? Sen.: **Current faculty include** the anthropologist Marshall Sahlins, ..., Shakespeare scholar David Bevington.	64.1%
Multiple sentence reasoning	There is anaphora, or higher-level fusion of multiple sentences is required.	Q: What collection does **the V&A Theatre & Performance galleries** hold? Sen.: **The V&A Theatre & Performance galleries** opened in March 2009. ... **They** hold the UK's biggest national collection of material about live performance.	13.6%
Ambiguous	We don't agree with the crowd-workers' answer, or the question does not have a unique answer.	Q: What is the main goal of criminal punishment? Sen.: **Achieving crime control via incapacitation and deterrence** is a major goal of criminal punishment.	6.1%

Fig. 1. Examples from the development set for each category of reasoning required to answer a question. Image credits to Rajpurkar et al., the original creators of the dataset.

TABLE I
Organization of data

Data fields	Information
id	Tensor holding the ID of the article
title	The article title contained in a tensor
context	A tensor holding an article context extract
query	Tensor with the asked query in it
text	Tensor holding the response's text
answer_start	Tensor representing the answer's context-specific starting index
is_impossible	A binary label of 1 for True and 0 for False that indicates whether or not the query can be answered

III. Benchmark BM25

BM25 is a widely used ranking algorithm in information retrieval that is appreciated for its ease of use and ability to produce search results that are pertinent to the user. The inclusion of term frequency and document length normalization is one of its key benefits. By taking into account both factors, the problem of document length bias is successfully resolved, preventing longer documents from unfairly predominating in search results.

Notwithstanding its advantages, BM25 has several drawbacks. Its disregard for the context or semantic meaning of the query and documents is a significant negative. For queries where contextual awareness is important, this may result in a ranking that is not optimal. The presumption of statistical independence between query phrases is another restriction. Term dependencies do matter in some real-world situations, and the search results may not be as accurate if these relationships are disregarded. Moreover, document length and phrase frequency have a major influence on BM25. This dependence could lead it to ignore other significant elements that could be crucial in figuring out how relevant a document actually is to a query, such document structure and relevance feedback.

IV. Methods

A. *Deep Passage Retrieval (DPR)*

The Dense Passage Retriever (DPR) serves as a technique for passage retrieval, differing from BM-25 or TF-IDF by extracting dense latent representations of passages and queries. DPR aims to index all passages within a low-dimensional and continuous space, allowing efficient retrieval of the top k-passages relevant to a given input question for the reader during run-time. The implementation of DPR involves both passage and encoder networks. While question and passage encoders theoretically can be any neural networks, two independent BERT networks (base, uncased) are utilized, and their representation at the [CLS] token is taken as the output, resulting in d = 768.

DPR employs a dense encoder $E_p()$, translating any text passage into d-dimensional real-valued vectors and constructing an index for all passages for retrieval. At run-time, a different encoder $E_q()$ maps the input question to a d-dimensional vector, facilitating the retrieval of k passages with vectors closest to the question vector. The similarity between the question and passage is determined using the dot product of their vectors:

$$\mathrm{sim}(q,p) = \mathbb{E}_Q[q]^\intercal \mathbb{E}_P[p] \tag{1}$$

During the inference phase, the passage encoder E_p is applied to all passages, and offline indexing using FAISS is carried out. FAISS, an extremely efficient open-source library for similarity search and clustering of dense vectors, can be easily scaled to handle billions of vectors. For a given question q during run-time, its embedding $v_q = E_Q[q]$ is derived, and the top k passages with embeddings closest to v_q are retrieved.

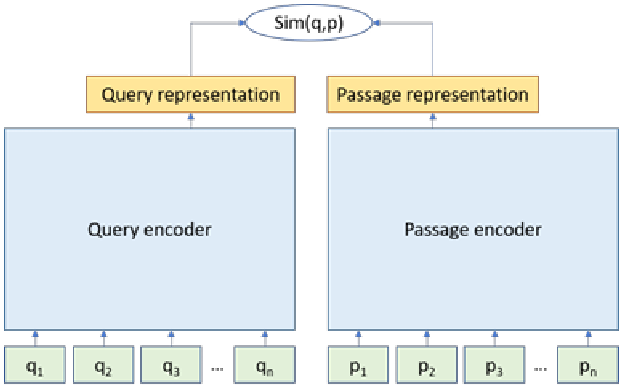


Fig. 2. DPR architecture

In our implementation, we employed a hybrid model that integrates outcomes from both DPR and BM25. This was achieved by calculating the linear combination of their individual scores to reevaluate the combined sets initially retrieved: $\lambda DPR(q) + BM25(q)$, with λ set to 1.1, a value fine-tuned empirically on the development set. This integration was carried out as hybrid methods have shown statistically significant improvements.

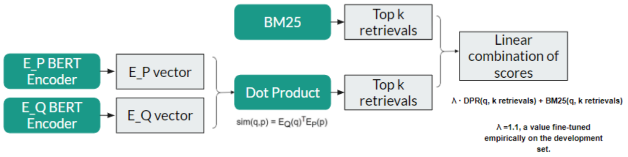

BM25
Top k retrievals
E_P BERT Encoder
E_P vector
E_Q BERT Encoder
E_Q vector
Dot Product
Top k retrievals
Linear combination of scores
$\lambda \cdot$ DPR(q, k retrievals) + BM25(q, k retrievals)
$sim(q,p) = E_Q(q)^T E_P(p)$
λ =1.1, a value fine-tuned empirically on the development set.

Fig. 3. Overall DPR + BM25 architecture

B. Bi-encoder with Cross-encoder re-ranker

In this part, we use a bi-encoder with cross-encoder re-ranker and compare it with the standalone bi-encoder model. The result of the better model will be compared with the benchmark BM25 to discuss later in section V. Results and discussion.

*1) **Bi-encoder:*** A bi-encoder comprises two essential components: the Context Encoder and the Query Encoder. The Context Encoder processes the input passages, converting them into fixed-size vector representations that encapsulate the contextual information. Simultaneously, the Query Encoder transforms the user's query into a corresponding vector representation. The Bi-encoder computes the similarity scores between a user's query and passages. The higher the similarity score, the more relevant the passage is considered.

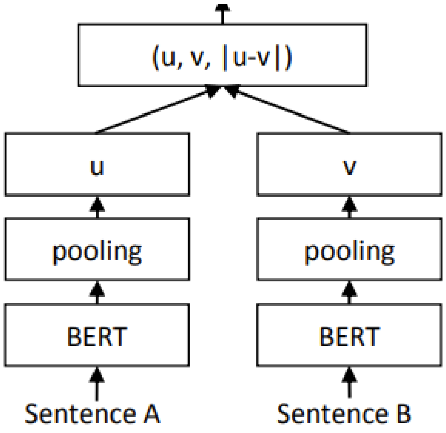


Fig. 4. Bi-encoder architecture

The key point lies in the shared space where both the context and query vectors exist. By encoding passages and queries into this shared space, the bi-encoder facilitates a direct and efficient measure of semantic similarity. The model learns to represent the underlying semantic structure of the text, enabling it to discern relevant information even in the presence of variations in wording.

The use of bi-encoders in passage retrieval represents a significant advancement as it addresses the challenge of capturing nuanced semantic relationships within textual data. The shared embedding space fosters a more robust and context-aware representation, ultimately enhancing the accuracy and effectiveness of passage retrieval systems.

*2) **Cross-encoder:*** The cross-encoder addresses the need for a comprehensive understanding of the relationship between a user query and potential passages. Unlike the bi-encoder, which separates the encoding of context and query, the cross-encoder takes a holistic approach by considering both simultaneously.

The cross-encoder functions by jointly encoding the input passage and query into a shared embedding space. This integration allows the model to capture intricate relationships and dependencies between the context and the query in a more direct manner. The result is a single vector representation that encapsulates the semantic relevance of the passage to the given query.

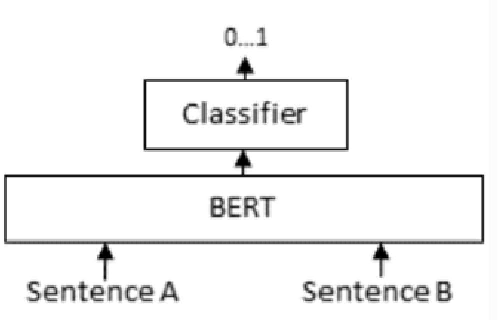


Fig. 5. Cross-encoder architecture

The cross-encoder function is good at capturing nuanced contextual information and semantic similarities. The model is trained to discern not only the relevance of a passage to a query but also to weigh the significance of individual words and phrases in forming that relevance.

*3) **Bi-encoder with cross-encoder re-ranker:*** In the model of Bi-encoder with Cross-encoder re-ranker, with a given search query, we use bi-encoder to encode the query and retrieve k potentially passages relevant for the query with semantic search. With k potentially passages, we use a re-ranker based on a cross-encoder that scores the relevancy of all candidates for the given search query. Output is a top-k ranked list of hit passages that users want.

*4) **Comparison between standalone Bi-encoder vs Combined bi-encoder and cross-encoder:*** To compare the accuracy of the standalone bi-encoder vs combined bi-encoder and cross-encoder, we use the evaluation data set with top-k=5 and top-k=20. k = 32 for the number of potential passages retrieved from bi-encoder in the

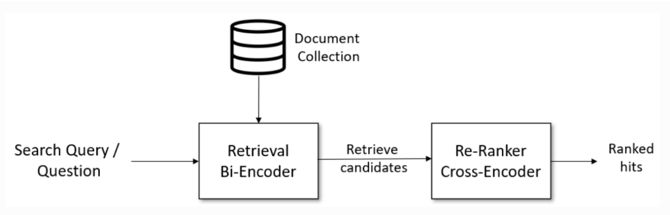


Fig. 6. Bi-encoder with cross-encoder re-ranker architecture

combined model. We truncate long passages to 256 tokens. The result is in the table I.

TABLE II
Accuracy of models with each top-k retrievals

	k = 5 (%)	k = 20 (%)
The standalone Bi-encoder	84.19	93.94
Bi-encoder + cross-encoder	94.47	95.88

*5) **Discussion:*** The incorporation of a cross-encoder re-ranker in conjunction with a bi-encoder has demonstrated notable improvements in the performance of passage retrieval tasks compared to using a standalone bi-encoder. Several key factors contribute to the enhanced effectiveness of this combined approach. Firstly, the bi-encoder, by design, captures contextual information separately for context and query. However, the cross-encoder introduces a holistic approach, jointly considering both the passage and the query during encoding. This simultaneous processing enables a more nuanced understanding of the semantic relationships between the user query and potential passages. Secondly, the cross-encoder excels in fine-grained relevance scoring. By evaluating the entire passage-query pair in a unified representation, it can weigh the significance of individual words and phrases in forming semantic relevance. This fine-grained analysis allows for a more accurate assessment of passage relevance to the user query. Then, the cross-encoder re-ranker acts as a refinement mechanism, allowing for additional contextual considerations after the initial retrieval by the bi-encoder. This refinement step is particularly beneficial in scenarios where the initial retrieval might yield multiple relevant passages, and the cross-encoder helps prioritize the most contextually appropriate results.

The passage retrieval tasks often involve addressing the semantic gap between the way queries are expressed and the content of passages. The combined use of bi-encoder and cross-encoder mitigates this gap by leveraging the strengths of both architectures, resulting in a more comprehensive and aligned representation of semantic content.

C. Bi-encoder with Fine-tuned Cross-encoder

We seek potential improvement by finetuning the cross-encoder (reranker) on the SQuAD dataset in two different ways: 1. Simple finetuning and 2. Fintuning with the BM25 score injection.

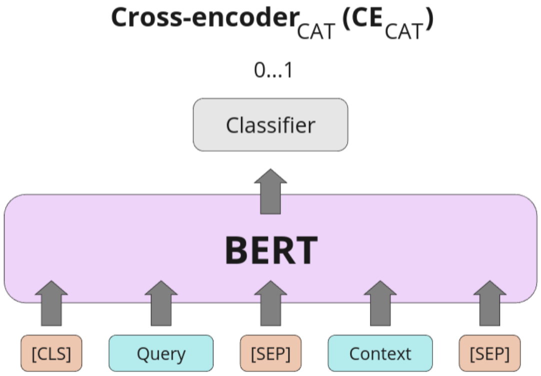


Fig. 7. Regular Cross Encoder Input

*1) **Simple Finetuning (CE$_{CAT}$):*** We concatenate each query and its context to form a single string as an input to the transformer. For every three negative examples (labelled 0), one positive example (labelled 1) is included in the training set, which is the positive-negative-ratio that can be treated as a hyperparameter. - Positive example: A query concatenated with the ground-truth context - Negative example: A query concatenated with the context that has the highest BM25 score among non-ground-truths. The output is a single score between 0 and 1 indicating how relevant the context is for the given query, and the cross entropy loss is used to compute the loss.

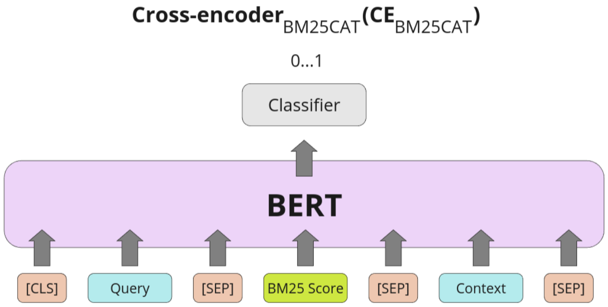


Fig. 8. Injection of BM25 Input

*2) **Finetuning with the BM25 score injection (CE$_{BM25CAT}$):*** In this method, BM25 score is injected into each query-context input as a string. For each query, we have two examples, one with the ground-truth as a positive example and the other as a negative example, which is the context with the highest BM25 scores among the contexts whose scores are lower than that of the ground-truth.

This injection is a method proposed by Arian Askari et al. [7], which is, in turn, inspired by the finding by Wallace et al. [8] that BERT models can capture numeric information. In common re-ranking set-ups, BM25 is widely utilized for finding the top-k documents to be reranked or finding hard negatives. However, the relevance score produced by BM25 based on exact lexical matching is not explicitly used in the downstream reranking stage. On the other hand, an ordinary cross-encoder tends underscore synthetic similarity, such a reranker can be further improved by a better exact word matching given that the presence of query words in the document is a strong signal for relevance in ranking.

Since BM25 scores do not have an upper bound and BERT-based models can process integers better than floats, the scores should be normalized first in order to interpret them. Min-max normalization is applied here as follows.

$$\text{Min-Max}(s_{BM25}) = \frac{s_{BM25} - s_{Min}}{s_{Max} - s_{Min}}$$

V. Results and Discussion

A. *DPR + BM25*

DPR excels at capturing lexical variations and semantic relationships, in contrast to term-matching methods like BM25, which tend to be sensitive to highly specific keywords and phrases. For instance, DPR can map synonyms or paraphrases with distinct tokens to vectors that are close to each other. Initially, we implemented the plain DPR model. Subsequently, it was found in one of the resources that a hybrid model yields superior performance. Our results also align with this observation:

TABLE III
Performance results of DPR and DPR + BM25

	k=5 (%)	k=20 (%)
DPR	74.664	89.772
DPR + BM25	79.044	93.500

These results may be explained by the fact that DPR and BM25 have complementary strengths in information retrieval. DPR is excellent at understanding deep semantic similarities between questions and passages, thus it can capture complex language patterns and nuances that traditional keyword-based methods might miss. On the other hand, BM25, a traditional information retrieval method, excels at keyword matching and is particularly effective with factoid queries where specific terms are important. By combining these approaches, the hybrid model can leverage both semantic understanding and keyword matching, leading to better overall performance than plain DPR.

B. *Bi-encoder with Cross-encoder Re-ranker*

Comparing accuracy with the benchmark in the evaluation data set.

TABLE IV
Accuracy of each models with each top-k

	k = 5 (%)	k = 20 (%)
Bi-encoder with Cross-encoder re-ranker	94.465	95.875
BM25	86.963	94.096

In the retrieving top-5 passages in evaluation data set, we recorded 892 cases that BM25 got wrong but the Bi-encoder with Cross-encoder re-ranker got right and 351 cases BM25 got right but the bi-encoder with Cross-encoder re-ranker got wrong. We found that BM25 lacks understanding of context, semantics, or word relationships. Then it struggles with complex queries. In general, not as accurate as the combined bi-encoder which has high accuracy, understands context and semantics, better at complex queries. But combined bi-encoder model is resource-intensive, slower, and not as scalable as BM25.

C. *Bi-encoder with Fine-tuned Cross-encoder*

TABLE V
Performance Comparison Among Different Methods

Model	**Method**	**Accuracy**	
		k=5 (%)	**k=20 (%)**
	BM25	86.963	94.096
distilroberta-base	Bi-cross encoder without fine-tuning	13.425	12.222
	CE_{CAT}	94.433	95.881
	$\text{CE}_{BM25CAT}$	93.772	95.782
ms-macro-MiniLM-L-6-v2	Bi-cross encoder without fine-tuning	94.465	95.875
	CE_{CAT}	94.863	95.856
	$\text{CE}_{BM25CAT}$	94.484	95.799

TABLE VI
Performance of CE_{CAT} with Different Hyperparameter Values

Warm up steps	**Pos-Neg Ratio**	**Accuracy**	
		k=5 (%)	**k=20 (%)**
5000	1pos-3neg	94.863	95.856
300	**1pos-3neg**	**94.901**	**95.885**
5000	3pos-1neg	94.948	95.856
300	3pos-1neg	94.844	95.875

1) ***Result Interpretation:*** First, we directly apply a distilled version of the RoBERTa-base model, 'distilroberta-base', to observe how much increment in performance can be achieved if it is fine-tuned, which is a mean of verifying our training implementation. While the raw model performs very poorly as expected given that it is mostly intended to be fine-tuned on a downstream task, it even outperforms BM25 in both k values after fine-tuning in either method, proving that our taining implementation is correct.

In the hope for achieving higher performance, we tried a cross encoder already trained on the MS Macro Passage Ranking task, ms–macro–MiniLM–L–6–v2. Despite being trained on a different dataset, the model already performs decently with 94.47% and 95.88% for $k = 5$ and $k = 20$ respectively. However, while performance for $k = 5$ sees a noticeable improvement, the opposite is seen in that for $k = 20$ for both methods of finetuning. Regardless, as can be seen from Fig., once the warmup_steps parameter is adjusted to 300 from 5000, the training is found to be more effective, resulting in performance increase for bi-cross without injection. Trying out different hyperparameter values of as in Fig., we have found that fine-tuned bi-encoder cross-encoder method without injection using warmup_steps of 300 and 1–pos–3–neg has the highest accuracy.

*2) **Effect of BM25 score injection:*** In contrast to the findings from the paper by Arian Askari et al. [7], the performance even slightly decreases when the cross encoder is fine-tuned with the BM25 score injection. In order to investigate this phenomena, we need to qualitatively assess the performance between the two models, simply fine-tuned bi-cross encoder and fine-tuned bi-cross encoder with BM25 injection. As can be inferred from Appendix A,

- Among question-context pairs that CE$_{CAT}$ gets right but CE$_{BM25CAT}$ gets wrong, most golden contexts don't explicitly contain the keywords from their respective queries, which requires the model to understand the contextual meaning of the query to be able to produce a correct answer. For example, in row 1 from Table VII, the two keywords of the query, "win/loss ratio" and "2015", are not included in the golden context. Because CE$_{BM25CAT}$ tends to favor exact lexical similarity, it fails to rank the context higher as opposed to CE$_{CAT}$.

- Among question-context pairs that CE$_{BM25CAT}$ gets right but CE$_{CAT}$ gets wrong, the keywords of the queries are usually included in the context. Even so, CE$_{CAT}$ also tends to look for other parts of the query. For example, in the first row of Table VIII, although query key word, "National Anthem", is included in the context, embedding-level close relation between "naming" from the first retrieved context by CE$_{CAT}$ and "who" from the query might have outweighted in the model's prediction.

Thus, we cannot conjuncture that CE$_{BM25CAT}$ is a poorer model than CE$_{CAT}$ in general since the former can still better rerank for many queries that the latter gets wrong. Overall, the nature of the query-context pairs each model gets wrong are different, and it is just that one type of pairs happen to appear more in the development set, which leads us to determine that one CE$_{BM25CAT}$ performs slightly worse statistically.

*3) **Choice of negative examples in fine-tuning:*** One way to improve training is to choose good hard negatives, which, in this case, are the passages that are not the golden context but are really convincing to be the one. In order to find such negatives,

1. For simple fine-tuning, we first retrieve relevant passages for each query using BM25, and choose the one with the highest BM25 score that is not the ground-truth.
2. For fine-tuning with BM25 injection, it becomes a bit tricky, because the negative obtained with the method used for simple fine-tuning usually has a higher BM25 score than that of the ground-truth. Hence, if we train the model with BM25 scores injected, we will be punishing the contexts with high BM25 scores, thereby undermining the importance of lexical similarity. Since we want a very positive-looking negative at the same time, we choose the example with the highest BM25 score that is lower than that of the ground-truth.

VI. Conclusion and Future Work

A. *Conclusion*

From our research, we can conclude that retrieving potentially relevant documents followed by reranking them can outperform the benchmark, BM25. Slight additional accuracy gain can be achieved by finetuning the cross-encoder used for reranking on the dataset. Indeed, among our implementations, the simply fine-tuned bi-cross encoder, CEBM25CAT, (warm up steps=300 and pos-neg ratio=1pos-3neg) can be concluded to perform the best among our implementations for passage retrieval on the SQuAD 1.1 dataset.

B. *Future Work*

As can be inferred from Table V, even a human might not be able to relate the golden context with the query. This is because in order to realize the golden passage includes the answer, one needs to know the "literal context" of the golden passage as well, which is included in other passages. For example, just by reading, *"The league eventually narrowed the bids to three sites:",* it is virtually impossible to know that it is going to specify *"the name of the stadium where Super Bowl 50 was played?"*, which explains why none of CE$_{CAT}$ and CE$_{BM25CAT}$ gets it right. Hence, in order to achieve higher accuracy on the SQuAD dataset, one viable improvement is to connect each passage with its previous n passages and its following n passages to better instill the context of the passage in the model.

(Here, *'passage'* is used instead of *'context'* to differentiate it from the literal context of the passage.)

Since the natural next step of document retrieval is question-answering, we can further work on applying our best model for the downstream task of narrowing down the scope of documents that a question-answering model has to look for in order to seek answers. As a case in point, by filtering out irrelevant documents or websites using CE$_{CAT}$, we can try to mitigate hallucination exhibited in Large Language Model (LLM) in the future.

References

[1] Xueguang Ma, Kai Sun, Ronak Pradeep, Minghan Li, and Jimmy Lin, 2022, “Another Look at DPR: Reproduction of Training and Replication of Retrieval”, https://dl.acm.org/doi/10.1007/978-3-030-99736-6_41

[2] Vladimir Karpukhin, Barlas Ouz, Sewon Min, Patrick Lewis, Ledell Wu, “Dense Passage Retrieval for Open-Domain Question Answering”, 2020, https://arxiv.org/abs/2004.04906

[3] Guilherme Rosa, Luiz Bonifacio, Vitor Jeronymo, Hugo Abonizio, Marzieh Fadaee, Roberto Lotufo, Rodrigo Nogueira, “In Defense of Cross-Encoders for Zero-Shot Retrieval”, 2022, https://arxiv.org/pdf/2212.06121.pdf

[4] Hyun Seung Lee, Seungtaek Choi, Yunsung Lee, Hyeongdon Moon, Shinhyeok Oh, Myeongho Jeong, Hyojun Go, Christian Wallraven. Cross Encoding as Augmentation: “Towards Effective Educational Text Classification”, 2023, https://aclanthology.org/2023.findings-acl.137.pdf

[5] Nils Reimers, Iryna Gurevych. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks, 2019, https://arxiv.org/pdf/1908.10084.pdf

[6] Bi-encoder with cross-encoder re-ranker, https://www.sbert.net/examples/applications/information-retrieval/README.html

[7] A. Askari, A. Abolghasemi, G. Pasi, W. Kraaij & S. Verberne, "Injecting the BM25 score as text improves BERT-based re-rankers", 2023, https://arxiv.org/abs/2301.09728

[8] Wallace, E., Wang, Y., Li, S., Singh, S., Gardner, M., "Do NLP models know numbers? Probing numeracy in embeddings.", 2019, https://arxiv.org/abs/1909.07940

[9] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, Percy Liang, “SQuAD: 100,000+ Questions for Machine Comprehension of Text”, 2016, https://doi.org/10.48550/arXiv.1606.05250

[10] Understanding the BM25 Ranking Algorithm | by Everton Gomede, PhD | Medium, https://medium.com/@evertongomede/understanding-the-bm25-ranking-algorithm-19f6d45c6ce

[11] How to build a smart search engine (Part II) | by Josh Taylor | Towards Data Science, https://towardsdatascience.com/how-to-build-a-smart-search-engine-a86fca0d0795

[12] R. Nogueira, K. Cho, “Passage Re-Ranking with BERT", 2019, https://arxiv.org/pdf/1901.04085.pdf

[13] S. Paul, “Information Retrieval with document Re-ranking with BERT and BM25”, 2020, https://medium.com/@papai143/information-retrieval-with-document-re-ranking-with-bert-and-bm25-7c29d738df73

Github repo:
https://github.com/linhmonkaist/Passage_retriever

Appendix

A. Outputs of Fine-tuned Models

TABLE VII
Queries that $\mathrm{CE}_{BM25CAT}$ gets wrong but CE_{CAT} gets right.

Query	**Golden Context**	**First Retrieved Context by CE$_{BM25CAT}$**
What was the **win/loss ratio** in **2015** for the Carolina Panthers during their regular season?	The Panthers finished the regular season with a 15-1 record, and quarterback Cam Newton was named the NFL Most Valuable Player (MVP). They defeated the Arizona Cardinals 49-15 in the NFC Championship Game and advanced to their second Super Bowl appearance since the franchise was founded in 1995. The Broncos finished the regular season with a 12-4 record, and denied the New England Patriots a chance to defend their title from Super Bowl XLIX by defeating them 20-18 in the AFC Championship Game. They joined the Patriots, Dallas Cowboys, and Pittsburgh Steelers as one of four teams that have made eight appearances in the Super Bowl.	Super Bowl 50 was an American football game to determine the champion of the National Football League (NFL) for the 2015 season. .... As this was the 50th Super Bowl, the league emphasized the "golden anniversary" with various gold-themed initiatives, as well as temporarily suspending the tradition of naming each Super Bowl game with Roman numerals (under which the game would have been known as "Super Bowl L"), so that the logo could prominently feature the Arabic numerals 50
What Denver player caused two fumbles for the Panthers?	The Broncos took an early lead in Super Bowl 50 and never trailed. Newton was limited by Denver's defense, which sacked him seven times and forced him into three turnovers, including a fumble which they recovered for a touchdown. Denver linebacker Von Miller was named Super Bowl MVP, recording five solo tackles, 2½ sacks, and two forced fumbles.	The Panthers seemed primed to score on their opening drive of the second half when Newton completed a 45-yard pass to Ted Ginn Jr. on the Denver 35-yard line on their second offensive play. .... But once again they came up empty, this time as a result of a Newton pass that bounced off the hands of Ginn and was intercepted by safety T. J. Ward. Ward fumbled the ball during the return, but Trevathan recovered it to enable Denver to keep possession

TABLE VIII
Queries that CE_{CAT} gets wrong but $\mathrm{CE}_{BM25CAT}$ gets right.

Query	**Golden Context**	**First Retrieved Context by CE$_{CAT}$**
Who did the **National Anthem** at Super Bowl 50?	Six-time Grammy winner and Academy Award nominee Lady Gaga performed the national anthem, while Academy Award winner Marlee Matlin provided American Sign Language (ASL) translation.	Super Bowl 50 was an American football game to determine the champion of the National Football League (NFL) for the 2015 season. .... suspending the tradition of **naming** each Super Bowl game with Roman numerals (under which the game would have been known as "Super Bowl L"), so that the logo could prominently feature the Arabic numerals 50.
Which team held the scoring lead throughout the entire game?	The Broncos took an early lead in Super Bowl 50 and never trailed. Newton was limited by Denver's defense, which sacked him seven times and forced him into three turnovers, including a fumble which they recovered for a touchdown. Denver linebacker Von Miller was named Super Bowl MVP, recording five solo tackles, 2½ sacks, and two forced fumbles.	Denver took the opening kickoff and started out strong with Peyton Manning completing an 18-yard pass to tight end Owen Daniels and a 22-yard throw to receiver Andre Caldwell. A pair of carries by C. J. Anderson moved the ball up 20 yards to the Panthers 14-yard line, but Carolina's defense dug in over the next three plays. First, linebacker Shaq Thompson tackled Ronnie Hillman for a 3-yard loss. Then after an incompletion, Thomas Davis tackled Anderson for a 1-yard gain on third down, forcing Denver to settle for a 3-0 lead on a Brandon McManus 34-yard field goal. The score marked the first time in the entire postseason that Carolina was facing a deficit.

TABLE IX
Queries that both CE$_{CAT}$ and CE$_{BM25CAT}$ get wrong.

Query	**Golden Context**	**First Retrieved Context by CE$_{CAT}$**
What did the next three drives result in?	There would be no more scoring in the third quarter, but early in the fourth, the Broncos drove to the Panthers 41-yard line. On the next play, Early knocked the ball out of Manning's hand as he was winding up for a pass, and then recovered it for Carolina on the 50-yard line. A 16-yard reception by Devin Funchess and a 12-yard run by Stewart then set up Gano's 39-yard field goal, cutting the Panthers deficit to one score at 16-10. The next three drives of the game would end in punts.	The further decline of Byzantine state-of-affairs paved the road to a third attack in 1185, when a large Norman army invaded Dyrrachium, owing to the betrayal of high Byzantine officials. Some time later, Dyrrachium—one of the most important naval bases of the Adriatic—fell again to Byzantine hands.
What is the name of the stadium where Super Bowl 50 was played?	The league eventually narrowed the bids to three sites: New Orleans' Mercedes-Benz Superdome, Miami's Sun Life Stadium, and the San Francisco Bay Area's Levi's Stadium.	Super Bowl 50 was an American football game to determine the champion of the National Football League (NFL) for the 2015 season. .... As this was the 50th Super Bowl, the league emphasized the "golden anniversary" with various gold-themed initiatives, as well as temporarily suspending the tradition of naming each Super Bowl game with Roman numerals (under which the game would have been known as "Super Bowl L"), so that the logo could prominently feature the Arabic numerals 50.