Enhancing QA Systems through Retrieval Augmented Generation: The Impact of Embedding Models on Retrieval Results

Student name: HUI Gi Wai Echo, Student ID: 23444584

Abstract

This project examines the impact of fine-tuning sentence transformer models on the information retrieval (IR) performance of Retrieval Augmented Generation (RAG) systems for question-answering (QA) tasks. This RAG approach offers a promising avenue to overcome the limitations of Large Language Models (LLMs) in handling contextually rich or recently updated information, thereby enhancing their applicability in domain-specific QA tasks. With a focus on improving retrieval quality with limited datasets, the project adopts a structured methodology to preprocess NLP research papers as the corpus and employs fine-tuning strategies to tailor a sentence transformer model for enhanced IR tasks. By leveraging traditional metrics and automated evaluation metrics, the project highlights substantial improvements in retrieval quality, demonstrating the potential of fine-tuning embedding models to augment the performance of QA systems.

1 Background Survey

The background survey initiates by exploring methodologies that tackle the inherent limitations faced by foundational Large Language Models (LLMs) in Question-Answering (QA) systems, particularly during the pre-retrieval and retrieval phases in a RAG pipeline. Following this, the discussion shifts towards the evolution of embedding technologies, charting a path from initial word-level embeddings to the context aware word embeddings such as BERT (Devlin et al., 2019) and its successors. These developments have markedly enhanced the semantic interpretation of both queries and documents, significantly benefiting QA systems. The concluding part of the survey delves into the fine-tuning processes for sentence transformer models, outlining various strategies designed to tailor these models for specific domain-related tasks. This background survey sets the stage for the methodology and experiments that follow, providing a comprehensive overview of domain-specific adaptations that further refine the semantic representation of sentences and their impact on QA systems.

1.1 Challenges with Base LLMs in Question-Answering Systems

Large Language Models (LLMs) like T5 (Raffel et al., 2020) have been at the forefront of advancements in question-answering (QA) systems, leveraging their extensive pre-trained knowledge to generate answers. However, a critical limitation of employing base LLMs for QA tasks lies in their lack of interpretability (Roberts et al., 2020). Unlike traditional QA systems, LLMs often fail to provide the contextual basis for their answers, leading to challenges in understanding the rationale behind the generated responses underscore this issue by highlighting the use of the T5 model for QA, where the reliance on the model's internal parameters for information retrieval hinders the provision of context for the answers generated.

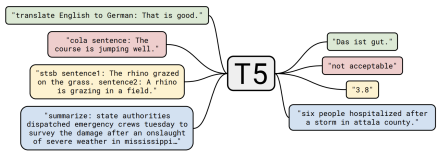


Figure 1: T5 model capabilities of various NLP tasks, including QA

1.2 Addressing LLM Limitations in QA with RAG

To mitigate these limitations, the Retrieval Augmented Generation (RAG) approach presents a promising solution. The RAG system architecture for this project is illustrated in Figure 2. RAG enhances the capabilities of LLMs by incorporating an information retrieval step prior to answer

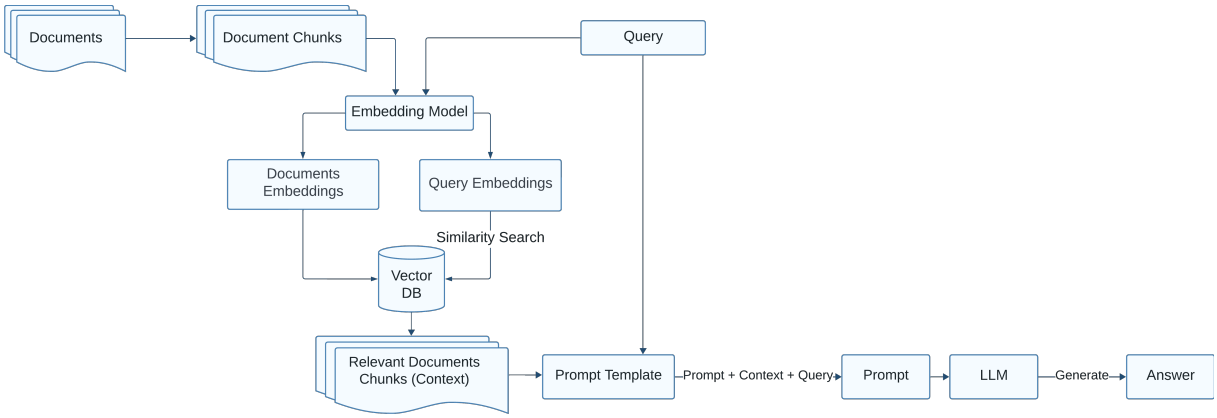


Figure 2: Retrieval Augmented Generation (RAG) system architecture

generation. This methodology involves querying a corpus to retrieve relevant documents based on the user's question, which are then used alongside the query to inform the LLM's response. Such an approach not only aims to improve the accuracy of the answers but also significantly enhances their interpretability by providing a contextually rich basis for the response. The integration of RAG into QA systems represents a pivotal advancement, particularly for domain-specific queries where precision and reliability are paramount. By addressing the "hallucination" issue prevalent in LLMs—where models generate plausible but incorrect or irrelevant information—RAG establishes a more trustworthy and efficient framework for QA tasks, showcasing its potential to revolutionize the field.

1.3 Enhancing Pre-Retrieval and Retrieval Stages in QA Systems

Chunk Optimization

Identifying the optimal chunk size for documents within the corpus is crucial in the pre-retrieval stage of RAG. Factors such as the nature of the indexed content, the embedding model's optimal block size, the expected length and complexity of user queries, and the specific application's utilization of the retrieved results should be considered when selecting an appropriate chunking strategy. Various chunk optimization techniques, such as sliding window technology, abstract embedding, metadata filtering, and graph indexing, have led to notable improvements in the retrieval system's accuracy and relevancy.

Choosing Embedding Models

The choice of embedding models plays a crucial role in enhancing semantic representations in the retrieval stage in RAG systems. The evolution of embedding models, from static word embeddings like Word2Vec (Mikolov et al., 2013a) (Mikolov et al., 2013b) and GloVe (Pennington et al., 2014) to context-aware embeddings introduced by BERT, has significantly improved the ability to capture semantic information. However, the introduction of sentence transformer models, such as SBERT (Reimers and Gurevych, 2019), has revolutionized the way queries and documents are embedded for retrieval tasks. Selecting an appropriate embedding model that aligns with the specific requirements of the target domain and effectively captures the semantic information of queries and documents is essential for optimal retrieval results in RAG systems.

Fine-tuning Embedding Models

Fine-tuning embedding models is a crucial step in customizing these models for domain-specific contexts. There are two primary paradigms in embedding fine-tuning methods: domain knowledge fine-tuning and fine-tuning for downstream tasks (Gao et al., 2024). Domain knowledge fine-tuning involves utilizing domain-specific datasets to accurately capture domain-specific information, while fine-tuning for downstream tasks leverages the capabilities of LLMs to generate task-specific retrievers and reward signals for data across multiple downstream tasks. This project focuses on experimenting fine-tuning embedding models with domain knowledge, exploring how this approach can effectively embed domain-specific queries and documents.

1.4 Word Embeddings

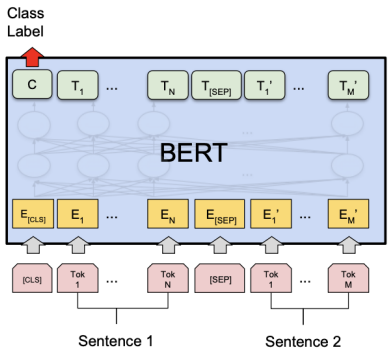
Static word-level embeddings Word2Vec which is a word embedding model that captures the similarity and analogy relationship of words and its surrounding words. However, Word2Vec is limited in capturing the context of words based on the context window. This limitation led to the improving the Word2Vec approach, like the development of GloVe, which is a global word embedding model that leverages the global corpus statistics in the training corpus using word-word co-occurrence counts. However, the static nature of these word-level embeddings makes it independent of the word context, which assign the same pretrained vector to the same word regardless of the context.

The introduction of BERT, a pre-trained transformer network, is capable of producing context-aware embeddings, enhancing word representations by generating different word vectors based on the context in which they are used. However, BERT's design focuses on word or token-level embeddings rather than sentence-level embeddings. This focus on words makes BERT less efficient in determining the semantic similarity between sentences (Reimers and Gurevych, 2019), which is crucial for tasks like information retrieval.

1.5 Sentence Embeddings

The introduction of sentence transformers represents a significant advancement in generating semantically rich sentence embeddings, surpassing traditional word embedding models in capturing semantic information. A notable breakthrough in this domain is the Sentence-BERT (SBERT) model (Reimers and Gurevych, 2019). SBERT, an adaptation of the pre-trained BERT model, employs siamese and triplet network architectures and is fine-tuned on Natural Language Inference (NLI) data, such as the SNLI (Bowman et al., 2015) and the Multi-Genre NLI (Williams et al., 2018) datasets, to produce meaningful sentence embeddings. This method marks a departure from less effective approaches, such as averaging BERT's output layer to obtain sentence embeddings, or using BERT directly for tasks requiring the assessment of semantic similarity between sentences (Figure 3).

One of the main computational challenges addressed by SBERT is the time and resources required for finding semantically similar sentences within large datasets. Traditionally, using BERT



(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG

Figure 3: Using BERT to perform sentence pair classification tasks

directly for sentence pair classification involves feeding both sentences into the network, making it computationally expensive for large collections of sentences. In the contrast, SBERT uses a siamese network architecture during training (Figure 4), hence generating sentence embeddings independently, which can then be compared efficiently using cosine similarity (Figure 5).

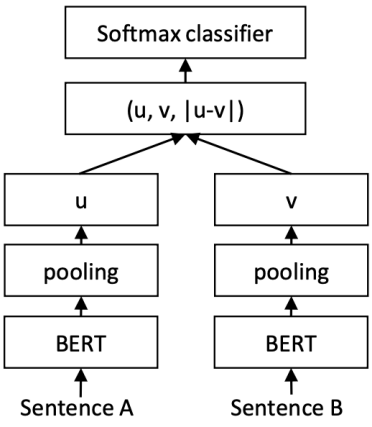


Figure 4: SBERT architecture with classification objective function during training

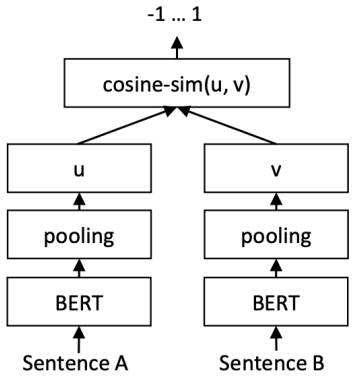


Figure 5: SBERT architecture at inference, for example, to compute similarity scores

SBERT significantly reduces the time required for identifying semantically similar sentence pairs in a collection of 10,000 sentences from approximately 65 hours using BERT to around 5 seconds using SBERT sentence embeddings using modern V100 GPUs (Reimers and Gurevych, 2019). This efficiency gain is crucial for information retrieval tasks in question-answering systems, which aim to retrieve the most relevant information quickly.

The improved quality of sentence embeddings directly translates to improved retrieval performance in the pre-retrieval stage of RAG tasks. Other notable sentence transformer models include Sentence-T5 (Ni et al., 2022), which is fine-tuned on the T5 model architecture, and Instructor (Su et al., 2023), an instruction-based fine-tuning approach that leverages an embedding architecture based on GTR models initialized from T5 models. By capturing more nuanced semantic information at the sentence level, SBERT enhances the semantic understanding of queries and documents, leading to more precise and efficient retrieval of relevant content. This highlights the critical role of high-quality sentence embeddings in advancing the performance of question-answering systems.

1.6 Fine-tuning Sentence transformer models

Fine-tuning sentence transformer models is essential for tailoring these advanced tools to meet the specific requirements of domain-specific applications, aiming with an improved performance of RAG systems in QA tasks. This is achieved by employing various fine-tuning strategies depending on the nature of the training data, which may include pairs or triplets of sentences, with or without explicit semantic similarity labels (Espejel, 2022). The goal of these fine-tuning methods is to capture the nuanced semantic relationships present in domain-specific corpora in order to improve both the accuracy and efficiency of information retrieval. The following are the fine-tuning strategies for sentence transformer models:

Sentence pairs with similarity labels

Loss functions optimize the model to place semantically similar sentences close together in the vector space while pushing dissimilar sentences apart based on the label indicating how similar they are.

Unlabeled similar (positives) sentence pairs

For similar sentence pairs without explicit similarity label, like pairs of paraphrases, pairs of (query, response) or pairs of (query, context), the loss functions focus on bringing similar sentences closer together in the vector space.

Triplets with labels

When working with triplets ([anchor, positive, negative]) with a label for each sentence type, the goal is to fine-tune embeddings so that relevant (anchor and positive) sentences are closer together while maximizing the distance between the anchor and negative sentences.

Triplets without labels

In the absence of specific labels for triplets, the fine-tune objective is similar to the labeled triplet scenario, with the model learning to bring relevant sentences closer together and push irrelevant sentences apart.

2 Methodology Implementation

This section outlines the structured approach taken to assess the impact of fine-tuning pre-trained sentence transformer models on the retrieval performance within RAG systems for question-answering tasks.

The methodology involves several key phases: selecting a specialized corpus of NLP research papers, preprocessing the data, fine-tuning the embedding models, and evaluating the modifications' effectiveness on the retrieval and answer generation capabilities of the system.

The effectiveness of these fine-tuned models is then evaluated through a dual approach. The sentence-transformers *InformationRetrievalEvaluator* is used to directly assess the improvements in retrieval accuracy. Additionally, the Retrieval

Augmented Generation Assessment System (RAGAs) framework is employed to provide a holistic analysis of both retrieval and answer generation performance post-fine-tuning. These evaluations demonstrate the potential of fine-tuned embedding models to significantly augment the efficiency and precision of question-answering systems.

2.1 Corpus Selection

Figure 6: This figure shows an example of corpus selection for NLP research papers, include example of GloVe, BERT, and SBERT papers

The process begins with the selection of a corpus, specifically targeting a collection of PDF documents from NLP research papers. This type of document is chosen as it has a structured and complex structure, including headings, titles, figures, and tables, which are common in research papers. The complex structure allows the project to test the effectiveness of the embedding models in capturing the semantic meaning of the research papers.

2.2 Preprocessing

To ensure the data is optimally prepared for embedding and retrieval processes, PDF parsing and preprocessing are performed. During PDF parsing, the textual content and tabular data are extracted from the complex structure of the PDFs, and the content is preprocessed.

2.3 Chunking

Following preprocessing, the preprocessed documents undergoes chunking. Chunking involves breaking down large documents into smaller pieces,

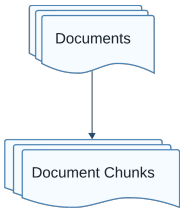


Figure 7: Chunking process for breaking down large documents into smaller pieces

thereby ensuring that each chunk is within the maximum token size that the embedding model can process without truncated information.

2.4 Fine-tuning Embedding Models with Domain Knowledge

Embedding models are fine-tuned using the SentenceTransformers library (Reimers, 2024), which offers a variety of loss functions tailored to different training data scenarios. This fine-tuning is crucial for the RAG system to effectively understand and retrieve relevant content within specialized domains.

2.5 Document Embeddings and Indexing

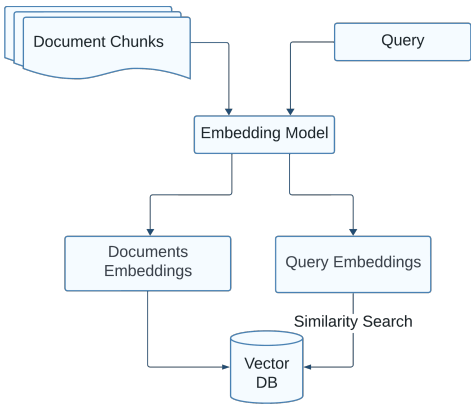


Figure 8: Document and query embeddings

The core of the methodology lies in the exploration of the pretrained and fine-tuned sentence transformer embedding models. The document chunks are embedded using the pre-trained and fine-tuned models, and the embeddings are indexed in a vector store for efficient retrieval.

2.6 Query Embeddings for Retrieval

Queries are embedded using the same models as document embeddings. These embeddings are crucial for the retrieval stage, where they are matched against document chunk embeddings in the vector store.

2.7 Document Retrieval by Similarity Search

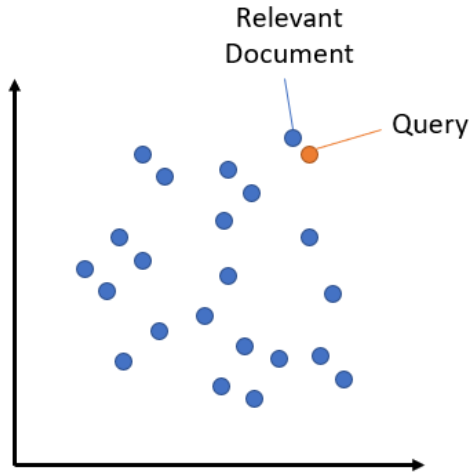


Figure 9: This figure illustrates the similarity search process. Image source: SentenceTransformers

In the document retrieval stage, similarity searches are performed to effectively retrieve relevant document chunks. By making use of similar search using the embeddings of the queries against those of the indexed document chunks in the vector store, the top-k document chunks are retrieved for each query. The retrieval process is designed to retrieve semantically relevant chunks represented in the vector space.

2.8 Answer Generation with LLM

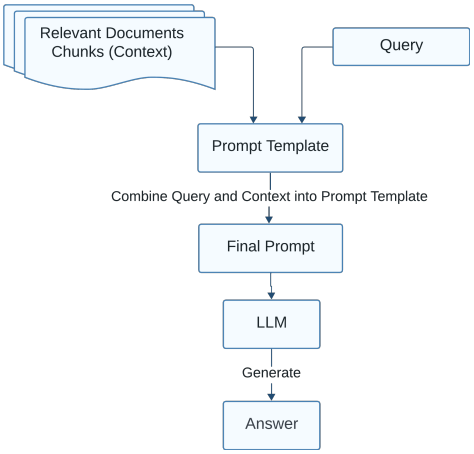


Figure 10: Answer generation process using Large Language Models

In the RAG QA pipeline, after relevant document chunks have been retrieved, they serve to augment the text generation capabilities of Large Language Models (LLMs). The process integrates the contextual information from these chunks into a prompt template, which is then combined with the user query to form a final, comprehensive prompt. This enriched prompt is fed into an LLMs. The final answer generated by the LLM is thus not only based on its internal knowledge it has been trained on but also provided with the relevant context by the retrieved document chunks, improving the contextual relevance of the generated answer.

2.9 Evaluation Metrics

The effectiveness of the fine-tuning approach is assessed using both the *sentence-transformers InformationRetrievalEvaluator* and the LLM-assisted automated evaluation metrics. These tools offer a comprehensive analysis of the retrieval and answer generation performance, providing insights into the impact of pre-trained and fine-tuned embedding models on the retrieval tasks within QA systems.

2.9.1 Information Retrieval Evaluator

The sentence-transformers *InformationRetrievalEvaluator* function is employed for embedding models' retrieval performance assessment, facilitating a direct comparison of IR metrics pre and post fine-tuning. This function computes a range of Information Retrieval (IR) metrics, including traditional metrics such as recall, precision, and accuracy to assess the retrieval performance of the model. The evaluation metrics provide an analysis of the model's effectiveness in retrieving relevant documents based on the input queries.

2.9.2 LLM-assisted Automated Evaluation Metrics

To assess the effectiveness of the fine-tuning approach implemented in this project, the LLM-assisted automated evaluation metrics provided by the RAGAS (Retrieval Augmented Generation Assessment) framework (Es et al., 2024) is also employed to evaluate the retrieval and answer generation performance of the fine-tuned models. This set of metrics extends beyond traditional IR measures such as recall, precision, and accuracy, which are captured by the InformationRetrievalEvaluator. This methodology diverges from conventional metrics that primarily evaluate the direct correlation between queries and retrieved documents. Instead, it offers a holistic assessment strategy that considers the dynamics among queries, retrieved documents, ground truths, and the answers generated. This evaluation strategy reduces the need for manual ground truth annotations. Such an approach is useful for holistic understanding the impact of

pre-trained and fine-tuned embedding models on retrieval tasks within QA systems.

The evaluation is conducted through a series of specifically designed metrics:

Context Precision: Measures whether all of the ground-truth present in the contexts are ranked higher or not. It makes use of the question, ground truth answer, and retrieved contexts for evaluation.

$$\text{Context Precision@K} = \frac{\sum_{k=1}^{K} (\text{Precision@k} \times v_k)}{\text{Total number of relevant items in the top } K \text{ results}}$$

$$\text{Precision@k} = \frac{\text{true positives@k}}{(\text{true positives@k} + \text{false positives@k})}$$

Figure 11: Context Precision formula

Context Recall: Measures the retrieved context aligns with the ground truth. It uses the ground truth answer and retrieved contexts for evaluation.

$$\text{context recall} = \frac{|\text{GT sentences that can be attributed to context}|}{|\text{Number of sentences in GT}|}$$

Figure 12: Context Recall formula

Context Relevancy: Measures the relevancy of the retrieved context to the question. It uses the question and retrieved contexts for evaluation.

$$\text{context relevancy} = \frac{|S|}{|\text{Total number of sentences in retrieved context}|}$$

Figure 13: Context Relevancy formula

Faithfulness: Measures whether the answer is grounded in the given retrieved contexts. It uses the generated answer and retrieved contexts for evaluation.

$$\text{Faithfulness score} = \frac{|\text{Number of claims in the generated answer that can be inferred from given context}|}{|\text{Total number of claims in the generated answer}|}$$

Figure 14: Faithfulness formula

Answer Relevancy: Measures answer generation relevancy to the query. It uses the question, generated answer, and retrieved contexts for evaluation.

$$\text{answer relevancy} = \frac{1}{N} \sum_{i=1}^{N} \frac{E_{g_i} \cdot E_o}{\|E_{g_i}\| \|E_o\|}$$

Figure 15: Answer Relevancy formula

3 Experiments

This section outlines the experiments designed to evaluate the impact of fine-tuning sentence transformer models on the retrieval capabilities of a QA system using a Retrieval Augmented Generation (RAG) pipeline. The Sentence-BERT paper is utilized as the corpus for these experiments, focusing on the fine-tuning of the sentence-transformers/all-mpnet-base-v2 model to assess its information retrieval performance.

3.1 Dataset - Corpus Selection

The corpus for the retrieval task comprises pre-processed PDF content from the Sentence-BERT research paper - Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. This selection aims to enhance the model's ability to retrieve information directly related to the content such as methodologies and findings discussed in the Sentence-BERT (SBERT) paper.

3.2 Dataset: Structure of the fine-tuning dataset

The fine-tuning dataset is organized into tuples of queries and relevant documents. Document chunks are extracted from the Sentence-BERT PDF. Each document chunk is limited to a maximum of 1800 characters or 360 tokens to ensure optimal processing by the embedding models. The queries are generated based on the document chunks, which are regarded as the relevant documents for the synthetic queries.

3.3 Data Pre-processing

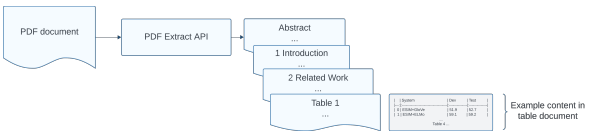


Figure 16: Data Pre-processing steps

Text extraction from the NLP research papers is performed using the PDF Extract API, which outputs a structured JSON file containing textual and tabular data. The preprocessing steps include removing newline characters, special characters, and lowercasing the text.

3.3.1 Chunk Optimization

This project also perform as simple chunk optimization to ensure that the document chunks are not sliced randomly. It involves the following steps:

- Structuring the JSON data into separate raw text documents based on the document's inherent headings, such as "Abstract," "1 Introduction," "2 Related Work," etc.

- Tables are extracted and stored as separate raw text files using markdown table syntax to ensure that document chunks are not randomly sliced.

- Content sections like Acknowledgements and References, which may not contain semantically rich information, are omitted to refine the data quality for fine-tuning.

- Larger sections are processed into smaller chunks using *RecursiveCharacterTextSplitter* and *SentenceTransformersTokenTextSplitter*, adhering to the 360 token limit.

3.4 Generating datasets for fine-tuning sentence transformer models

The datasets for fine-tuning are structured as tuples of queries and relevant documents. The chunked documents serve as context for the generation of synthetic queries. This fine-tuning process and the subsequent evaluation are conducted using the InformationRetrievalEvaluator.

The LLM employed for query generation is gpt-3.5-turbo-0125 with a temperature setting of 0.8. A total of 150 questions are generated, with five questions per chunk and two per table chunk, placing higher emphasis on textual than tabular content.

3.4.1 Prompt template generates queries:

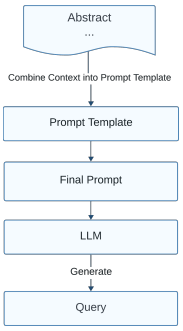


Figure 17: Generating queries from the context

You are a smart assistant designed to come up with meaningful questions. Given the context information and not prior knowledge, create

`{num_questions_per_chunk}`

questions to be fully answered from the given context. The question should be framed from a part that contains non-trivial information. The questions should be of moderate difficulty, reasonable, and must be understood by humans. Do not use phrases like provided context in the question.

`Context: {context}`

Example of a generated query: How was sbert trained and fine-tuned in the given context?

Example of the relevant document as the context: ... 3.1 training details we train sbert on the combination of the snli bowman et al., 2015 and the multi genre nli williams et al., 2018 dataset. the snli is a collection of 570, 000 sentence pairs annotated with the labels contradiction, eintailment, and neutral. multinli contains 430,000 sentence pairs and covers a range of genres of spoken and written text. we fine tune sbert ...

3.5 Fine-tuning Datasets Setup

The experiments utilize distinct datasets for different stages of the model training process—fine-tuning, validation, and testing. These datasets comprise question-document pairs generated from the SBERT corpus, segmented as follows:

- **Training Dataset:** Constitutes 70% of the pairs, used to train the model to encode the nuances and detailed information from the SBERT paper effectively.

- **Validation Dataset:** Comprises 20% of the pairs, utilized for periodic evaluation during the training process to monitor and optimize model performance.

- **Testing Dataset:** Makes up the remaining 10% and includes the corpus (all the document chunks), queries, and relevant documents. This dataset serves as the final evaluation set to assess the model's retrieval capabilities and the impact of fine-tuning, ensuring the model's ability to generalize to new, unseen questions related to the SBERT context.

3.6 Fine-tuning Configurations

Table 1 details the configurations used in the fine-tuning process of the sentence transformer model.

Table 1: Fine-tuning Configurations

	Configuration
Model	sentence-transformers/all-mpnet-base-v2
DataLoader	Tuples of (queries, relevant documents) using Pytorch DataLoader
Loss Function	MultipleNegativesRankingLoss
Training Epochs	3
Batch Size	10

3.7 Evaluation Metrics using InformationRetrievalEvaluator

The Information Retrieval Evaluator function is employed to assess the retrieval performance of both the pre-trained and the fine-tuned models. As mentioned earlier in this section, the baseline model is the pre-trained sentence-transformers/all-mpnet-base-v2. The evaluation metrics, which include recall, precision, and accuracy at varying levels of k, along with Mean Reciprocal Rank (MRR), Normalized Discounted Cumulative Gain (NDCG), and Mean Average Precision (MAP), provide a comprehensive analysis of each model's effectiveness in retrieving relevant documents based on the input queries.

For the evaluation, 10% of the SBERT PDF prepared dataset, encompassing the entire corpus of document chunks, queries, and relevant documents, is utilized.

3.7.1 Experiment Results and Findings using InformationRetrievalEvaluator

The comparative analysis of the pre-trained and fine-tuned models reveals significant enhancements in information retrieval performance post fine-tuning is shown in Table 2. Key findings include:

Accuracy: There is a noticeable improvement in the model's accuracy, with cosine similarity-based accuracy at the top retrieved document (cos_sim-acc@1) increasing from 37% to 60%.

Precision and Recall: Both metrics exhibit substantial improvements; for instance, cos_sim-precision@1 and cos_sim-recall@1 both rise to 60%.

Other Performance Metrics: The fine-tuning process leads to notable enhancements in other significant metrics. The MRR improved from 45% to 70%, NDCG from 51% to 76%, and MAP from 47% to 70% when considering the top 10 retrieved documents using the dot score.

Table 2: Comparison of pre-trained and fine-tuning results using InformationRetrievalEvaluator

Metric	Baseline	Fine-tuned
cos_sim-Accuracy@1	0.37	0.60
cos_sim-Accuracy@3	0.47	0.77
cos_sim-Accuracy@5	0.50	0.87
cos_sim-Accuracy@10	0.73	0.93
cos_sim-Precision@1	0.37	0.60
cos_sim-Recall@1	0.37	0.60
cos_sim-Precision@3	0.16	0.26
cos_sim-Recall@3	0.47	0.77
cos_sim-Precision@5	0.10	0.17
cos_sim-Recall@5	0.50	0.87
cos_sim-Precision@10	0.073	0.093
cos_sim-Recall@10	0.73	0.93
cos_sim-MRR@10	0.45	0.70
cos_sim-NDCG@10	0.51	0.76
cos_sim-MAP@100	0.47	0.70
dot_score-Accuracy@1	0.37	0.60
dot_score-Accuracy@3	0.47	0.77
dot_score-Accuracy@5	0.50	0.87
dot_score-Accuracy@10	0.73	0.93
dot_score-Precision@1	0.37	0.60
dot_score-Recall@1	0.37	0.60
dot_score-Precision@3	0.16	0.26
dot_score-Recall@3	0.47	0.77
dot_score-Precision@5	0.10	0.17
dot_score-Recall@5	0.50	0.87
dot_score-Precision@10	0.073	0.093
dot_score-Recall@10	0.73	0.93
dot_score-MRR@10	0.45	0.70
dot_score-NDCG@10	0.51	0.76
dot_score-MAP@100	0.47	0.70

3.8 Evaluation Metrics using Automated Evaluation

In this phase, the fine-tuning approach's effectiveness is further assessed through an automated evaluation setup, leveraging the RAGAs framework. This extends the evaluation beyond conventional metrics, using Context Precision, Context Recall, Context Relevancy, Faithfulness, and Answer Relevancy. The assessment assesses the fine-tuned model's performance against the baseline pre-trained model, highlighting the impact of

fine-tuning on the QA system's efficacy.

3.8.1 Generating Datasets for Automated Evaluation

A distinct dataset has been curated for the automated evaluation, comprising triplets of queries, context, and ground truth answers. These triplets are derived from the *Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks* PDF to yield new questions that are reflective of the document's essence but distinct from the queries used in the Information Retrieval evaluation.

Synthetic Queries and Ground Truth Answers Queries are synthetically generated using gpt-3.5-turbo-0125 with a set temperature of 0.8. This process results in 50 new questions, which are then paired with ground truth answers. These answers are formulated using gpt-4-turbo-preview with zero temperature settings, ensuring their accuracy and relevance to the provided context.

3.8.2 Prompt template generates ground truth answers:

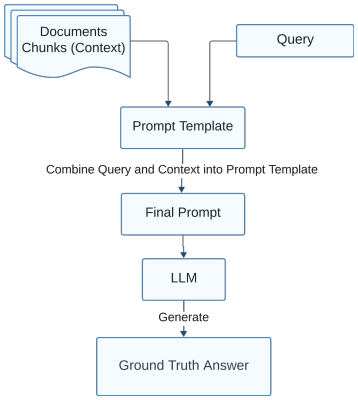


Figure 18: Generating ground truth answers from the context

You are a research assistant specializing in NLP research. You should come up with an answer to the given question using the information from the given context from a research paper.

answer: a answer about the context.

Format the output as JSON with the following keys:

```
question: {question}
context: {context}
answer:
```

Example of a triplet dataset (query, context, ground truth answer): An example triplet from the generated dataset includes:

- **Generated query:** What is the main difference between the common practice of deriving sentence embeddings using BERT and the approach developed in SBERT?
- **Relevant document as context:** a common method to address clustering and semantic search is to map each sentence to a vector space such that semantically similar sentences are close. researchers have started to input individual sentences into bert and to derive fixed size sentence embeddings. the most commonly used approach is to average the bert output layer known as bert embeddings or by using the output of the first token the cls token. as we will show, this common practice yields rather bad sentence embeddings, often worse than averaging glove embeddings pennington et al., 2014. to alleviate this issue, we developed sbert. the siamese network . . .
- **Generated ground truth answer:** The main difference between the common practice of deriving sentence embeddings using BERT and the approach developed in SBERT is the methodology for generating fixed-size sentence embeddings. In the common practice, sentence embeddings are derived by either averaging the BERT output layer or using the output of the first token CLS token, which often yields poor results. In contrast, SBERT utilizes a siamese network architecture to produce fixed-sized vectors for input sentences, which allows for efficient semantic similarity searches and clustering by using similarity measures like cosine similarity or Manhattan/Euclidean distance. This approach significantly. . .

3.8.3 Automated Evaluation Dataset

With the synthetic triplet dataset prepared, the automated metrics can be used. The questions prepared are used as queries to retrieve the context from the document chunks. The retrieved context is then used to generate the answer to the question. With the prepared ground truth answer, the evaluation metrics are calculated. The automated metric calculates the evaluation based on the question, retrieved

context, generated answer, and ground truth answer.

This structured approach underpins a comprehensive evaluation, allowing for a holistic understanding of the fine-tuning's effects on the QA system's performance.

Example of a automated evaluation dataset For the generation of the evaluation dataset, document chunks are embedded into embeddings using both the pre-trained baseline and the fine-tuned models. The queries are embedded using the either the baseline or the fine-tuned embedding model, and the top-k document chunks are retrieved for each query.

- **Vector Database:** Use to store the embedded document chunks for retrieval and use as the retriever for retrieving the top-k document chunks for each query.
- **Number of Retrieved Documents:** 2

Example of the automated evaluation dataset:

- **Example of the query:** What is the main difference between the common practice of deriving sentence embeddings using BERT and the approach developed in SBERT?
- **Example of the retrieved documents as the relevant context:**

  '2 related work... a large disadvantage of the bert network structure is that no independent sentence embeddings are computed, which makes it difficult to derive sentence embeddings from bert. to bypass this limitations, researchers passed single sentences through bert and then derive a fixed sized vector by either averaging the outputs . . . '

  '. . . we present sentence bert sbert, a modification of the pretrained bert network that use siamese and triplet network structures to derive semantically meaningful sentence embeddings that can be compared using cosine similarity. . . '
- **Example of the generated answer:** The main difference between the common practice of deriving sentence embeddings using BERT and the approach developed in SBERT is that SBERT utilizes siamese and triplet network structures to derive semantically meaningful sentence embeddings that can be compared using cosine-similarity. This approach reduces the computational effort significantly compared to BERT, making it much faster to find the most similar pair of sentences while maintaining the accuracy of BERT embeddings.
- **Example of the ground truth answer:** The main difference between the common practice of deriving sentence embeddings using BERT and the approach developed in SBERT is the methodology for generating fixed-size sentence embeddings. In the common practice, sentence embeddings are derived by either averaging the BERT output layer or using the output of the first token CLS token, which often yields poor results. In contrast, SBERT utilizes a siamese network architecture to produce fixed-sized vectors for input sentences, which allows for efficient semantic similarity searches and clustering by using similarity measures like cosine similarity or Manhattan/Euclidean distance. This approach significantly. . .

3.8.4 Experiment Results and Findings using Automated Evaluation

Analyzed through the lens of automated evaluation framework, the fine-tuned model demonstrates improvements across the majority of assessed metrics. A notable exception is observed in the metric of context relevancy, which showed a marginal decrease.

This further analysis complements the findings from information retrieval evaluator result, further substantiating the benefits of fine-tuning for improving the performance of retrieval-augmented question-answering systems. Table 3 encapsulates the comparative performance of the pre-trained and fine-tuned models.

Table 3: Automated evaluation result: Comparison of QA System Performance Metrics Before and After Fine-tuning Embedding Models

Metric	Pre-trained	Fine-tuned
Context Precision (%)	86.5	93.0
Context Recall (%)	83.5	94.5
Context Relevancy (%)	9.7	9.4
Faithfulness (%)	93.2	96.3
Answer Relevancy (%)	84.5	90

3.9 Overall Findings and Implications

The cumulative evidence from both the InformationRetrievalEvaluator and automated evaluations points to a clear trend: the fine-tuning of embedding models significantly enhances performance on the targeted corpus. This trend is consistent and substantial, indicating the robustness of the fine-tuning method applied to the specific dataset.

Directions for Future Experiments:

- Investigating the effects of varying document chunk sizes, for instance, contrasting the retrieval quality of specific questions arising from 1-2 sentence chunks against more general questions sourced from paragraph-sized chunks.

- Diversifying the synthetic query generation methods, such as creating queries derived from multiple document chunks as opposed to single-chunk queries, may yield further insights into the optimization of fine-tuning practices for QA systems.

4 Application: Fine-Tuned Embedding Model in a QA System

4.1 Selection of the Embedding Model

The embedding model fine-tuned in the preceding experiments was chosen for its enhanced information retrieval capabilities. This section details the practical application of the fine-tuned model within a QA system designed to operate in real-world settings.

4.2 QA System Development

A QA system was developed by leveraging contemporary technologies such as Langchain, ChainLit, and the Chroma vector database. This system, utilizing the Sentence BERT paper's corpus, showcases the efficacy of the fine-tuned embedding model in accurately retrieving relevant document chunks. These chunks provide the context for generating answers through a large language model (LLM). The architecture of this system allows for its application across diverse academic research papers, enabling researchers and students to access relevant information within a domain-specific corpus.

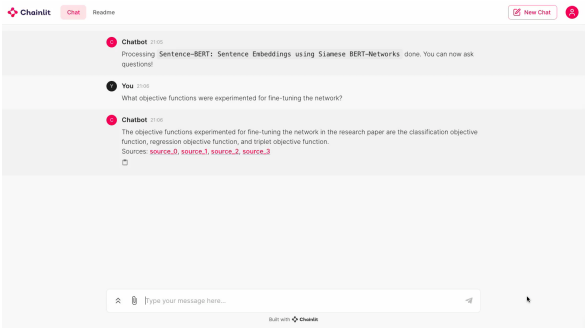


Figure 19: QA Application

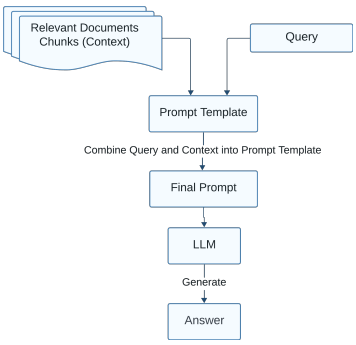


Figure 20: Generating answers using Large Language Models

4.2.1 Prompt template generates the answer in a RAG pipeline:

You are an AI assistant designed to provide accurate and helpful responses to questions related to the NLP research papers. Your goal is to always provide conversational answers based solely on the context information provided and not rely on prior knowledge. If you cannot answer the question with the context, please respond with 'Hmm, I'm not sure'.

```
context: {context}
```

Given the context information and not prior knowledge, answer the question.

```
Question: {question}
Final Answer:
```

Sample Query and Answer

- **Query:** What objective functions were experimented for fine-tuning the network?

- **Answer:** The objective functions experimented for fine-tuning the network in the research paper are the classification function, regression objective function, and triplet objective function.

Discussion: The response accurately captures the key objective functions explored in the research paper, demonstrating the model's ability to retrieve and summarize relevant information effectively.

4.3 Demonstration and Real-World Implications

The demonstration, while centered on the Sentence BERT paper for the QA pipeline, reflects a scalable process applicable to a wide range of academic research papers in PDF format. The methodology consists of the following steps:

1. Preprocessing and indexing the target document for the QA application.

2. Automatically prepare the fine-tuning dataset as described in the previous sections.

3. Fine-tune the embedding model.

4. Use the fine-tuned model for query embedding and document retrieval in the QA process.

This approach illustrates that substantial improvements in information retrieval can be achieved even with a limited dataset. The successful application of a fine-tuned embedding model to structured scientific papers underscores its potential to enhance academic research settings through improved information retrieval performance in a Retrieval Augmented Generation QA system.

5 References

References

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

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.

Shahul Es, Jithin James, Luis Espinosa Anke, and Steven Schockaert. 2024. RAGAs: Automated evaluation of retrieval augmented generation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 150–158, St. Julians, Malta. Association for Computational Linguistics.

Omar Espejel. 2022. Train and fine-tune sentence transformers models.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality.

Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith Hall, Daniel Cer, and Yinfei Yang. 2022. Sentence-t5: Scalable sentence encoders from pretrained text-to-text models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1864–1874, Dublin, Ireland. 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.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer.

Nils Reimers. 2024. Sentence-transformers training overview.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426, Online. Association for Computational Linguistics.

Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. 2023. One embedder, any task: Instruction-finetuned text embeddings. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1102–1121, Toronto, Canada. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.