Improving BERT-based FAQ Retrieval System using Query, Question and Answer Simultaneously

Hyunsoo Cho
School of Robotics
Kwangwoon University
Seoul, Republic of Korea
hscho1221@naver.com

Jiwon Choi
School of Robotics
Kwangwoon University
Seoul, Republic of Korea
trwy25@naver.com

Changju Noh
School of Robotics
Kwangwoon University
Seoul, Republic of Korea
ncj0302@naver.com

Kwang-Hyun Park
School of Robotics
Kwangwoon University
Seoul, Republic of Korea
akaii@kw.ac.kr

***Abstract*—FAQ retrieval systems are one of the fields of information retrieval and perform the task of retrieving appropriate question-answer pairs from user queries. Recent studies separately train deep learning models using dense representations of query-question similarity and query-answer relationships. However, since the sentences of a question-answer pair may contain different information, retrieval performance can be improved by using both sentences simultaneously when measuring similarity to the user's query. In this paper, we propose a learning method that can improve retrieval performance by training the relationship between queries and question-answer pairs with a single encoder. We evaluated P@5, MAP, and MRR performance using human-labeled queries on the StackFAQ dataset. To show that performance can be increased by improving the learning method, we trained and verified using the same model as in the previous study and the same query dataset created with GPT-2. Furthermore, we showed that performance can be further improved by training using queries created with GPT-3.5.**

Keywords—FAQ retrieval system, BERT, sentence embedding, joint learning, data augmentation

I. INTRODUCTION

Information retrieval systems are becoming increasingly important in the digital age, enabling rapid search and access to a variety of data. Among them, the Frequently Asked Questions (FAQ) retrieval system is a field of information retrieval system that receives user queries and ranks QA pairs of documents in order of relevance, as shown in Fig. 1, and is mainly used on websites. Candidate documents in an FAQ retrieval system consist of QA pairs, and both question and answer sentences can be used to measure the degree of relevance between the user's query and the candidate documents.

However, there are some difficulties in the FAQ retrieval system. First, even if the QA pair is unrelated to the user's query, the degree of relevance is often high if there are many overlapping words. This problem occurs especially when measuring the similarity between queries and questions. Second, there are difficulties in creating datasets. While QA pairs are data that can be automatically collected through crawling, etc., the user's query dataset to search for the corresponding QA pairs must be directly labeled [1]. We need a query dataset to learn relationships with QA pairs, but this method is expensive. To solve this problem, previous research [2] used the GPT-2 [3] model to automatically label query datasets in QA pairs and use them for training. The data generated in this way is of lower quality than human labeling, and the performance is slightly lower than the results in [4], but the cost of creating the dataset was eliminated and the performance closely caught up.

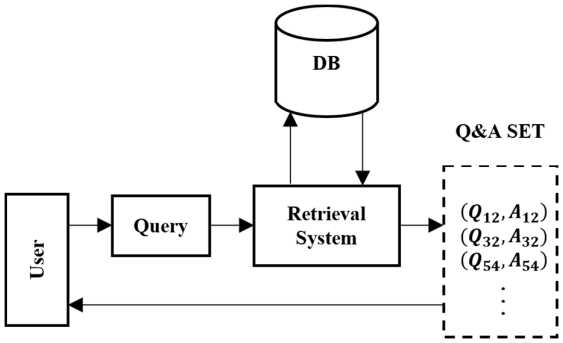

Fig. 1 Process of retrieving Q&A sets from user's query

In this paper, we solve the above two difficulties. First, in existing studies [1], [2], [4], [5], [6], the relationships between queries-questions and queries-answers were independently trained and measured, and the results of both were used to rank the QA pairs. However, we analyzed the dataset and discovered several characteristics. The user's query and the question sentences of the QA pair are similar in nature and have similar sentence lengths. The more overlapping words there are, the more likely they are to actually be related documents, but even if they are unrelated, it is difficult to distinguish them well. On the other hand, the answer sentence is long and contains a lot of information, and the sentence length is very different from the user's query, so it is less sensitive to overlapping words, but generally does not distinguish related QA pairs better than query-question. Through this analysis, we hypothesized that the above problems can be improved by training the questions and answers simultaneously with one neural network model rather than training them independently when learning the relevance to the user's query, and proposed a learning method for this purpose. In addition, we use a query dataset created with GPT-2 [3], which was used in existing research [2], to eliminate the cost of data labeling and at the same time verify that performance is improved just by changing the learning method. Moreover, we use GPT-3.5 to create a better quality query dataset and learn the model to see if performance improves [7], and the performance is compared by training the model using a semi-labeling method that mixes human-labeled queries with automatically generated queries.

II. Related Works

A. Term-Frequency based

To measure relevance, algorithms based on word frequency have been widely used previously [8], [9], [10]. BM25[11] is a representative algorithm among them and calculates the degree of relevance as in equation (1).

$$\sum_{i}^{Q} IDF(q_i) * \frac{TF(q_i,\ D) * (k_1 + 1)}{TF(q_i, D) + k_1 * (1 - b + b * \frac{L_d}{L_{avgd}}} \quad (1)$$

As one of the previously widely used TF-IDF algorithms, it reflects the length of documents and has weights such as k_1 and b. This word frequency-based algorithm has the advantage of fast retrieval execution time, but has the disadvantage of not being able to distinguish between synonyms or understanding the semantic relationship between similar expressions.

B. Deep-Learning based

To solve the shortcomings of word frequency-based algorithms, neural network models using deep learning are being used. Several neural network models were proposed. In the study of [1], the CNN [12] model was used, and in the DPR [13] study, the BERT [14] model was used as a dense representation model to embed queries and candidate documents and calculate similarity, significantly improving retrieval performance. Since then, many studies in the field of information retrieval have used the BERT [14] model, and this method has also been applied to FAQ retrieval systems.

In the study of [4], the fine-tuned BERT [14] model is used to measure the degree of relevance between queries and answers, and the top 10 documents are retrieved as a result. Within it, the similarity between queries is measured using TSUBAKI, a BM25 algorithm family, and the ranking is readjusted for documents whose value is higher than hyper-parameter α. Although performance was improved by combining several methods, a human-labeled dataset was used as the query dataset to learn the relevance of queries and answers. In addition, because it uses the BERT model as a cross-encoder structure, the retrieval execution time is long, making it difficult to use in a real environment [15].

In [2], the top 100 documents are first retrieved using the BM25 algorithm. Next, the relevance of queries and questions, and queries and answers were measured separately with a fine-tuned BERT model, and the 100 initially ranked documents were reranked along with the results of BM25_maxpsg[2], a family of BM25 algorithms. At this time, CombSUM [16], [17] was used as a reranking method. In addition, the query dataset is automatically created with the fine-tuned GPT-2[3] model, eliminating the cost of creating the dataset for training the BERT[14] model, and retrieval execution time was reduced by using the BERT model with the structure of a bi-encoder [15]. However, when comparing the quality of the query dataset created through GPT-2 [3] and the query dataset created using the latest LLM, there is a significant difference, and the performance also has the disadvantage of being relatively low compared to previous research [4].

These deep learning-based methods have the disadvantage of greater memory and execution time costs compared to word frequency-based algorithms. However, through embedding, problems with existing algorithms can be solved by understanding and distinguishing semantic relationships between similar words or sentences.

III. Proposed Method

The studies in [1], [2], [4], [5], and [6] all independently calculate the query-question and query-answer relationships. However, based on the results of analyzing the dataset, we hypothesized that performance can be improved if the relationship between queries, questions, and answers are simultaneously trained in one model, and we propose a learning method for this purpose.

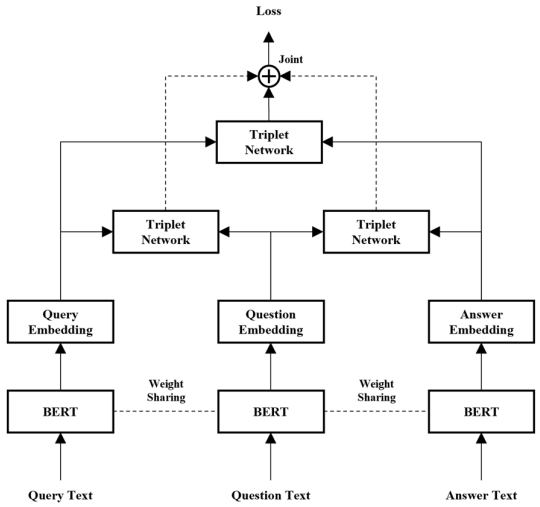

Fig. 2 Model architecture of the first approach

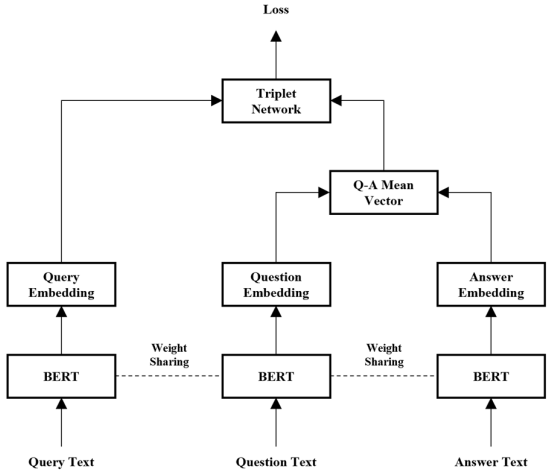

Fig. 3 Model architecture of the second approach

A. Joint Learning of Query-Question, Query-Answer, Question-Answer

The first learning structure of the proposed method is shown in Fig. 2. First, the query, question, and answer sentences are each input into the BERT[14] model and encoded. Here, the BERT[14] model is composed of a siamese-network[18] structure that shares weights. The three sentences are embedded and vectorized, and the loss is calculated using triple-network [19] for query-question, query-answer, and question-answer. At this time, since the query and question are sentences with similar characteristics, learning about question-answer can have a similar effect as

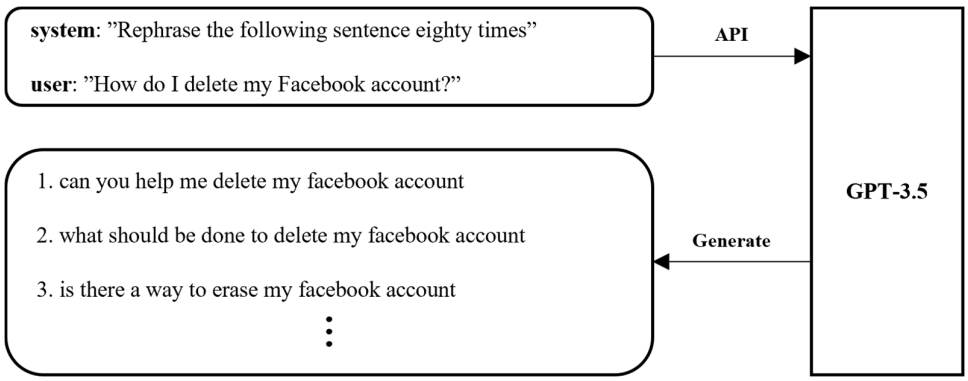

Fig. 4 Diagram of generating queries with GPT-3.5 API

TABLE II EVALUATION RESULTS ON GPT-2 LABELED DATASET

Methods	P@5 ↑			MAP ↑			MRR ↑		
	Min	Max	Mean	Min	Max	Mean	Min	Max	Mean
GPT-2 labeling(Paper)		0.74			0.87			0.88	
GPT-2 labeling (Ours)	0.72	**0.75**	0.74	0.87	**0.89**	0.88	0.89	**0.91**	0.90

TABLE II EVALUATION RESULTS ON GPT-3.5 LABELED DATASET

Methods	P@5 ↑			MAP ↑			MRR ↑		
	Min	Max	Mean	Min	Max	Mean	Min	Max	Mean
GPT-3.5 labeling (Ours) - 10	0.74	0.76	0.75	0.89	0.91	0.90	0.91	0.92	0.92
GPT-3.5 labeling (Ours) - 20	0.75	0.77	0.76	0.90	0.92	0.90	0.91	0.93	0.92
GPT-3.5 labeling (Ours) - 30	0.75	**0.77**	0.76	0.89	**0.92**	0.91	0.91	**0.93**	0.92
GPT-3.5 labeling (Ours) - 40	0.75	0.77	0.76	0.89	0.92	0.90	0.90	0.93	0.92

TABLE III EVALUATION RESULTS ON SEMI-LABELED DATASET

Methods	P@5 ↑			MAP ↑			MRR ↑		
	Min	Max	Mean	Min	Max	Mean	Min	Max	Mean
Only Human labeling (Paper)		0.78			0.90			0.94	
GPT-3.5 + human labeling (Ours) - 30/2	0.75	0.77	0.76	0.90	0.93	0.92	0.91	0.94	0.93
GPT-3.5 + human labeling (Ours) - 30/3	0.77	0.79	0.78	0.92	0.94	0.93	0.93	0.96	0.94
GPT-3.5 + human labeling (Ours) - 30/5	0.80	**0.81**	0.81	0.94	**0.96**	0.95	0.95	**0.97**	0.96

learning about query-answer. Finally, each loss is combined to obtain joint loss [20], [21], and the BERT [14] model is trained. When learning with this structure, the BERT[14] model learns the correlation between query, question, and answer sentences simultaneously.

B. Learning Query-Mean Vector of Question and Answer

The second structure of the proposed learning method is shown in Fig. 3. Similar to the learning structure in III.A, query, question, and answer sentences are all embedded through the BERT [14] model, which shares weights with each other. Next, the mean vector of the question and answer vectors is obtained, and the query vector is passed through triplet-network [19] to obtain the loss and train the BERT [14] model. In this way, the query does not independently learn the relationship between questions and answers, but uses the information of questions and answers together with the user's query to rank QA pairs.

C. Reranking Method

The reranking method is to re-rank a document several times using various ranking methods. Similar to [2], this paper performs reranking using the BM25[11] algorithm and BM25_maxpsg. Each document for which similarity is to be sought is divided using a sliding window to obtain each similarity, and the largest value among them is taken as the similarity of the document [2].

Reranking is done in the following order. First, the top 100 QA pairs are retrieved using the BM25 algorithm. Within them, the similarity for each QA pair is calculated and combined with the BM25_maxpsg algorithm and the models trained using the method proposed in this paper. The documents are finally ranked using the similarity of the QA pair finally obtained.

IV. Experiment

A. Dataset

The StackFAQ dataset [1] was used as the dataset. This dataset consists of QA pairs filtered by crawling question-answer websites and human-labeled query sentences for each QA pair. There are 125 types of questions, and each question has an average of 5 answers, for a total of 719 QA pairs. There are 1,249 query data, with humans manually labeling each question with about 10 items, and the query dataset created with GPT-2 released in [2] consists of 855 QA pairs. In this paper, we trained a model with 855 automatically generated query datasets that do not require labeling costs, and evaluated the model with 1,249 human-labeled datasets for more accurate evaluation. Additionally, as the LLM model has recently developed a lot, we also studied it with queries created with GPT3.5 and compared its performance. Fig. 4 shows the process of creating a query dataset using the API. Additional experiments were conducted using a small mix of human-labeled queries and compared with previous research [4] in which training was conducted using only human-labeled queries.

B. Evaluation

As performance evaluation indicators, we use P@5, MAP, and MRR, which are frequently used in recommendation systems and are the same indicators used in previous studies. For all three evaluation indicators, the more related documents there are in the top ranking, the higher the score, and MAP and MRR are indicators that take into account the ranking order of related documents.

C. Experiment Details

In order to verify that performance was improved just by changing the learning method, the hyperparameters, dataset, and loss function were configured as identically as possible to those in the study [2]. Triplet-loss [19] was used as the loss function of the BERT [14] model, and Euclidean distance and cosine distance were used as distance functions. Additionally, five negative data for triplet learning were randomly sampled from among unrelated documents. At this time, considering randomness, the minimum, maximum, and average performance were measured with 256 results. Hyper-parameters were 5 epochs, 8 batch sizes, and 2e-5 learning rate, and RTX3090 24GB GPU and A100 40GB GPU were used for training.

V. Results

Results when using a query dataset created with GPT-2[3] as a training query dataset, when using a query dataset created with GPT-3.5, and when mixing a small amount of human-labeled datasets, were compared.

A. GPT-2 Labeled Query

TABLE I shows the performance specified in [2] and the performance of the model applying the method proposed in this paper, all of which are the results of training with queries generated with GPT-2 [3]. By keeping all experimental details the same, we showed that the results can be improved with the proposed method.

B. GPT-3.5 Labeled Query

TABLE II is the result of applying the proposed learning method and additionally using the query dataset created with GPT-3.5 instead of GPT-2[3] as the training dataset. When the quality of the query dataset was improved with GPT-3.5, performance was improved in all indicators. Additionally, when the performance was compared by increasing the number of queries generated for each question to 10, 20, 30, and 40, the performance gradually increased, but did not increase after 30. This can be seen as overfitting because the diversity of queries that can be generated for each question is limited.

C. Mixed with GPT-3.5 and Human Labeled Query

TABLE III is the result of training with a dataset that is a small mixture of queries created with GPT-3.5 and queries created by humans. Queries created with GPT-3.5 used a dataset with 30 queries per question, which performed best, and human-generated queries were experimented with a mix of 2, 3, and 5 queries per question. At this time, the queries in the existing evaluation dataset are human-labeled datasets with 10 queries per question, and the queries included in training were removed and evaluated. As a result, it was confirmed that the performance gradually increased as the amount of human-labeled data increased, and all performances were higher than the results of [4], which learned using only human-labeled query datasets.

VI. Conclusion

This paper proposes a learning method for the BERT model that can improve retrieval performance by simultaneously training query, question, and answer sentences in single model. To verify the proposed method, we made the experimental environment as identical as possible to the research in [2], used a training query dataset generated by GPT-2, and were able to increase performance in all three indicators: P@5, MAP, and MRR. In addition, by creating a new training query dataset with GPT-3.5, we were able to improve the quality of the query and improve all three evaluation indicators. At this time, compared to the performance of [4], which used a dataset in which all training queries were directly labeled by humans, P@5 and MRR were close, but MAP was higher, and in the experiment using semi-labeling, all performance indicators were high.

References

[1] M. Karan and J. Šnajder, "Paraphrase-focused learning to rank for domain-specific frequently asked questions retrieval," Expert Systems with Applications, vol. 91, pp. 418-433, January 2018.

[2] Y. Mass, B. Carmeli, H. Roitman, and D. Konopnicki, "Unsupervised FAQ retrieval with question generation and BERT," Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 807-812, July 2020.

[3] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," OpenAI blog, vol. 1, no. 8, pp. 9, 2019.

[4] W. Sakata, T. Shibata, R. Tanaka, and S. Kurohashi, "FAQ retrieval using query-question similarity and BERT-based query-answer relevance," Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, Paris, France, July 21-25, 2019. pp. 1113-1116.

[5] S. Gupta, and V. R. Carvalho, "FAQ retrieval using attentive matching," Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, Paris, France, July 21-25, 2019, pp. 929-932.

[6] J. Seo, T. Lee, H. Moon, C. Park, S. Eo, I. D. Aiyanyo, K. Park, A. So, S. Ahn, and J. Park, "Dense-to-question and sparse-to-answer: Hybrid retriever system for industrial frequently asked questions," Mathematics, vol. 10, no. 8,pp. 1335, 2022.

[7] H. Dai, Z. Liu, W. Liao, X. Huang, Z. Wu, L. Zhao, W. Liu, N. Liu, S. Li, D. Zhu, H. Cai, Q. Li, D. Shen, T. Liu, and X. Li, "ChatAug:

Leveraging ChatGPT for text data augmentation," arXiv:2302.13007, 2023.

[8] C.-H. Huang, J. Yin, and F. Hou, "A text similarity measurement combining word semantic information with TF-IDF method," Jisuanji Xuebao(Chinese Journal of Computers), vol. 34, no. 5, pp. 856-864, 2011.

[9] S. Albitar, S. Fournier, and B. Espinasse, "An effective TF/IDF-based text-to-text semantic similarity measure for text classification," Proceedings of 15th International Conference on Web Information Systems Engineering (WISE 2014), Thessaloniki, Greece, October 12-14, 2014.

[10] B. Bakiyev, "Method for determining the similarity of text documents for the Kazakh language, taking into account synonyms: extension to TF-IDF," Proceedings of 2022 International Conference on Smart Information Systems and Technologies (SIST), Nur-Sultan, Kazakhstan, April 28-30, 2022.

[11] S. Robertson and H. Zaragoza, "The probabilistic relevance framework: BM25 and beyond," Foundations and Trends in Information Retrieval, vol. 3, no. 4, pp. 333-389, 2009.

[12] K. T. O'Shea and R. Nash, "An introduction to convolutional neural networks," arXiv:1511.08458, 2015.

[13] V. Karpukhin, B. Oğuz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W.-t. Yih, "Dense passage retrieval for open-domain question answering," arXiv:2004.04906, 2020.

[14] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," arXiv:1810.04805, 2018.

[15] S. Humeau, K. Shuster, M.-A. Lachaux, and J. Weston, "Poly-encoders: Transformer architectures and pre-training strategies for fast and accurate multi-sentence scoring," arXiv:1905.01969, 2019.

[16] E. Fox and J. A. Shaw, "Combination of multiple searches," NIST Special Publication, SP, pp. 243-243, 1994.

[17] A. K. Kozorovitsky and O. Kurland, "From identical to similar: Fusing retrieved lists based on inter-document similarities," Journal of Artificial Intelligence Research, vol. 41, pp. 267-296, 2011.

[18] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," Proceedings of ICML Deep Learning Workshop, vol. 2, no. 1, 2015.

[19] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," Proceedings of International Workshop on Similarity-Based Pattern Recognition (SIMBAD 2015), Copenhagen, Denmark, October 12-14, 2015, pp. 84-92.

[20] P. H. Martins, Z. Marinho, and A. F. T. Martins, "Joint learning of named entity recognition and entity linking," arXiv:1907.08243, 2019.

[21] Q. Chen, Z. Zhuo, and W. Wang, "BERT for joint intent classification and slot filling," arXiv:1902.10909, 2019.