IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 26, 2021, accepted August 5, 2021, date of publication August 11, 2021, date of current version August 17, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3103963

Adaptive Batch Scheduling for Open-Domain Question Answering

DONGHYUN CHOI[1,2], MYEONGCHEOL SHIN[1], EUNGGYUN KIM[1], AND DONG RYEOL SHIN[2]

[1]Kakao Enterprise, Pangyo 13494, South Korea
[2]School of Software, Sungkyunkwan University, Suwon 16419, South Korea

Corresponding author: Donghyun Choi (heuristic.c@kakaoenterprise.com)

ABSTRACT Open-domain question answering aims to get answers for given questions from a set of documents. Recently, dual encoder architecture is widely adopted to dense passage retrieval for question answering. In-batch negative sampling is typically used to gather extra negative samples during training. In this paper, we propose adaptive batch scheduling to enhance the performance of in-batch negative sampling. The proposed algorithm schedules training batches to increase the difficulty of the sampled negatives by in-batch negative sampling during training. We evaluated the proposed approach on the two well-known document retrieval benchmark datasets MSMARCO and Natural Questions. The evaluation result shows that the proposed adaptive batch scheduling could significantly improve the document retrieval performances of dual encoder architecture document retrieval systems.

INDEX TERMS Open-domain question answering, dense passage retrieval, batch scheduling.

I. INTRODUCTION

Open-domain question answering is the task of finding the answers for given natural questions from a large collection of documents. Most question answering systems take pipeline-based approaches [1]–[6]; they first search for documents relevant to the given question and extract answers from the retrieved documents.

Recent advancements in the machine reading comprehension (MRC) task significantly improved the answer extraction performance. When a question and a corresponding paragraph are given, an MRC system could extract an answer to the question from the passage with high accuracy. Due to the emergence of pre-trained language models such as BERT [7], the performances of MRC systems even outperform human performances.[1] For a given question, [1] first tried to retrieve relevant articles from Wikipedia using the traditional keyword-based information retrieval approaches such as TF-IDF or BM25, and analyzed the retrieved documents with a machine reading comprehension system to get an answer.

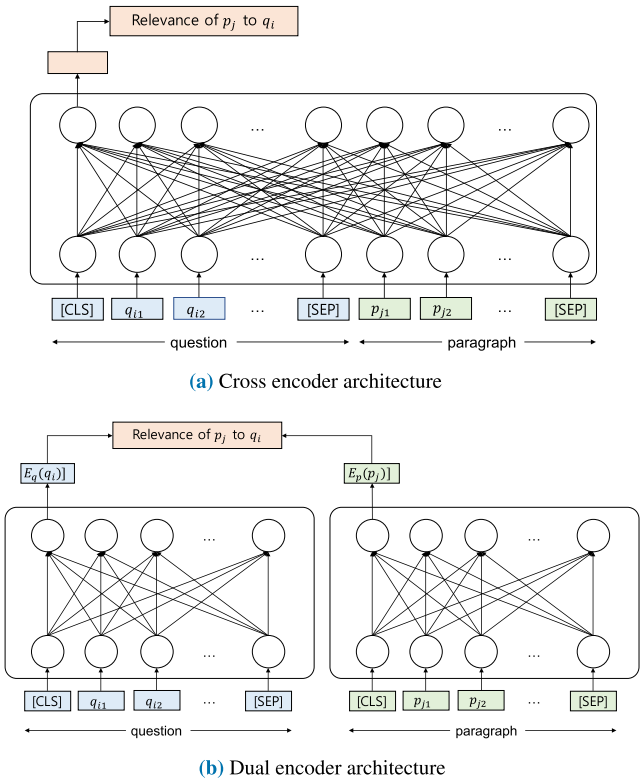
Keyword-based approaches suffer from the term mismatch problem; if a passage does not contain the same terms as in the question, the passage will not be retrieved. Moreover, even when a passage contains the question terms, it is not guaranteed that the passage has the answer for the given question. Many works explored the dense passage retrieval approaches to deal with the term mismatch problem.

Dense passage retrieval approaches encode questions and paragraphs with deep neural models to get the relevance scores between them. Figure 1 shows two encoding network architectures for dense passage retrieval. The cross encoder architecture encodes a question q_i and a paragraph p_j together to get the relevance score between them. The accuracy of the cross encoder is relatively higher compared to the dual encoder architecture. However, it takes considerable amount of time to encode all passages with the given question in runtime. Meanwhile, the dual encoder architecture separately encodes questions and paragraphs. It has the benefit of encoding all the paragraphs before runtime. For a given question, a dual-encoder encodes the question into dense representation and compares it with the indexed paragraph representations. Recent works on dense passage retrieval mainly focus on the dual-encoder architecture [6], [8]–[11].

In-batch negative sampling is a widely used technique in dual encoder architectures to provide extra negative examples during training. Figure 2 briefly illustrates the in-batch negative sampling. For each question and positive paragraph pair (q_i, p_i) in a batch with n elements, positive examples of

The associate editor coordinating the review of this manuscript and approving it for publication was Yiming Tang.

[1]https://rajpurkar.github.io/SQuAD-explorer/



(a) Cross encoder architecture

(b) Dual encoder architecture

FIGURE 1. Encoding architectures for dense passage retrieval.

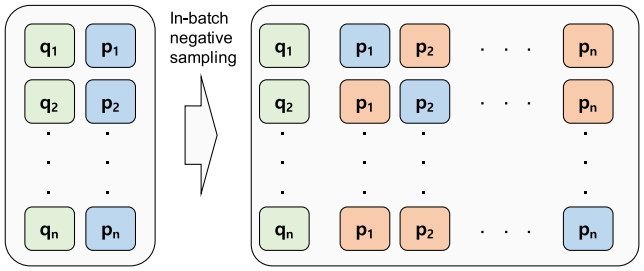


FIGURE 2. In-batch negative sampling.

the questions other than q_i are considered negative examples of q_i. Then the training goal becomes to choose one positive paragraph p_i among the paragraphs contained in the batch for a given question q_i.

Currently, in-batch negative sampling is considered a random negative sampling since batches are generated randomly. As shown in [6], providing hard negatives instead of random negative examples during training could significantly improve the retrieval performance. This paper proposes adaptive batch scheduling (ABS) to provide hard negatives during in-batch negative sampling. The proposed approach first calculates the hardness score for each batch by summarizing the similarities of questions and paragraphs inside the batch. Then training instances are swapped between batches to maximize the sum of batch hardness scores across training batches. We applied the proposed ABS to train dual encoder networks for dense passage retrieval and evaluated the trained models against two well-known document retrieval benchmark datasets, MSMARCO [12] and Natural Questions [13]. The proposed ABS significantly improves the document retrieval performances of dual encoders on both benchmark datasets.

Contributions of this paper could be summarized as follows.

- We propose adaptive batch scheduling, which forces in-batch negative sampling to provide hard negatives instead of random negative samples.
- We experimentally show the effectiveness of our proposed approach using the two well-known document retrieval benchmark datasets.
- We conduct experiments to examine the effectiveness of the proposed approach in detail.

II. RELATED WORK

Document retrieval for open-domain question answering could be classified into two major categories. Sparse representation approaches index terms with inverted index. Those approaches are fast and efficient, but they suffer from the term mismatch problem. [1] retrieves the top 5 relevant articles from Wikipedia using TF-IDF with bigram features and analyzes the retrieved documents with a machine reading comprehension system to get an answer. [2] and [3] expanded each document by generating possible queries of the document using neural language models. The expanded documents are indexed using the BM25 [14] algorithm. [5] augments queries instead of documents through text generation of heuristically discovered relevant contexts. [4] utilized BERT [7] to learn the weights of the terms in a document. The learned term weights replaced the traditional term frequencies during indexing.

Dense representation approaches typically encode each paragraph and question separately into vectors. A similarity between the question encoded vector and paragraph encoded vector measures the relevance of a paragraph to a query. [8]–[10] all tried to pre-train language models for question and paragraph encoding. [8] proposed inverse cloze task to pre-train language models for encoding questions and paragraphs. [9] proposed additional pre-training tasks and showed the effect of each pre-training task experimentally. [10] augmented language model pre-training with a latent knowledge retrieval.

Some other dense representation approaches focused on the fine-tuning. [11] tried to populate negative samples by retrieving top-n passages ranked with BM25. [6] proposed a framework to filter out false negatives from the populated negative samples. [15] uses multiple vectors, i.e., the encoded vectors of each token, instead of one vector to represent each question and paragraph. [16] also uses multiple vectors to encode a paragraph, but only overlapping terms between the question and the paragraph are used to calculate their similarity.

In-batch negative sampling is widely used in dense representation approaches [6], [8]–[11], [15], [16] to feed

negative samples during training. Although in-batch negative sampling is done at random, providing hard negative samples could significantly improve the system performance, as shown in [6]. In this paper, we propose the adaptive batch scheduling method to generate training batches in the way of sampling "hard" negatives instead of random negatives during in-batch negative sampling.

The proposed approach could be considered similar to the boosting algorithms such as AdaBoost [17] in terms of finding out the hard training samples. But there exists a significant difference between boosting algorithms and the proposed algorithm. The boosting algorithms weigh already existing training instances and find out the hard examples among those instances. Meanwhile, the proposed algorithm finds out the new hard training instances by combining different training instances during training.

III. ADAPTIVE BATCH SCHEDULING

In this section, we describe the proposed adaptive batch scheduling in more detail.

A. GOAL

Let the training instances $T = \{d_1, d_2, \ldots, d_t\}$, where d_i is the pair of a question q_i and its corresponding positive paragraph p_i. We set $s_{ij} = sim(q_i, p_j)$ to be the relevance score between a question q_i and a paragraph p_j.

During the training phase, the training corpus is divided into m batches $T^B = \{B_1, B_2, \ldots, B_m\}$. For each batch B_k, we define the hardness score of the batch $h(B_k)$ as follows:

$$h(B_k) = \sum_{d_i \in b_k, d_j \in b_k, i \neq j} s_{ij} \tag{1}$$

Intuitively, $h(B_k)$ is the sum of relevance scores between questions and their negative paragraphs inside the same batch when applying the in-batch negative sampling. Note that a positive paragraph p_i for a question q_i is considered a negative example for the question q_j other than q_i, when the in-batch negative sampling is applied. Higher $h(B_k)$ suggests that during in-batch negative sampling, for a question $q_i \in B_k$, its sampled negatives p_js will probably have higher relevance scores s_{ij} on average. As a result, more difficult negative examples will be provided for q_i during the batch training.

The goal of adaptive batch scheduling is to schedule training data instances for each batch to maximize the sum of batch hardness scores:

$$T^B_{scheduled} = \underset{T^B}{\text{argmax}} \sum_{B_k \in T^B} h(B_k) \tag{2}$$

In the equation, $T^B_{scheduled}$ represents the scheduled training batches $\{B'_1, \ldots, B'_m\}$. With the scheduled training batches $T^B_{scheduled}$, each training question will encounter more difficult negative examples during training.

B. ALGORITHM

We propose a greedy algorithm to generate the ABS-scheduled training batches $T^B_{scheduled}$. For the given training instances $T = \{d_1, d_2, \ldots, d_t\}$, the proposed algorithm iteratively creates member batches of $T^B_{scheduled}$ until all training instances belong to a batch.

Let U be a set of training instances which are not assigned to any batch yet. With batch size n, the proposed algorithm first randomly selects n elements from U to construct an initial batch B. Then the algorithm iteratively replaces an instance $d_r \in B$ with another training instance $d_a \in (U - B)$ to maximize the hardness score $h(B)$. The replacement stops when the algorithm could find no such (d_r, d_a) pair anymore. Once the replacement stops, the resultant batch B is added to the set of scheduled training batches $T^B_{scheduled}$, and the training instances of B are removed from the set U.

Algorithm 1 shows the pseudocode of the proposed adaptive batch scheduling algorithm.

Algorithm 1 Adaptive Batch Scheduling

Input Training instances $T = \{d_1, d_2, \ldots, d_t\}$
Question/paragraph encoding models E_q and E_p
Batch size n
Output Scheduled training batches $T^B_{scheduled}$

```
Encode questions and paragraphs using E_q and E_p
Calculate s_ij for all questions q_i and paragraphs p_j
U ← T
T^B_scheduled ← {}
while U ≠ ∅ do
    B ← Randomly selects n instances from U
    while true do
        d_r ← argmax_{d∈B} h(b − d)
        d_a ← argmax_{d∈U, d∉B} h(B − d_r + d)
        if h(B − d_r + d_a) > h(B) then
            Remove d_r from B
            Add d_a to B
        else
            break
        end if
    end while
    Add batch B to T^B_scheduled
    Remove training instances contained in B from U
end while
```

It is straightforward to expand the algorithm to handle the case when additional negative samples are populated before training. For two training instances $d_i = (q_i, p^p_i, p^n_i)$ and $d_j = (q_j, p^p_j, p^n_j)$, we redefine the relevance score $s_{ij} = sim(q_i, p^p_j) + sim(q_i, p^n_j)$. In the equation, p^p_j and p^n_j represent the positive and negative paragraphs of the question q_j, respectively.

A question could have multiple answer paragraphs, or two or more questions could have the same answer paragraph. In those cases, the ABS will group those training instances into the same batch, introducing training noises. To prevent such cases, we maintain a list of positively annotated paragraphs for each training question. During the ABS, for two training instances (q_i, p_i) and (q_j, p_j), we set s_{ij} to 0 if

p_j is the positively annotated paragraph of the question q_i. By doing so, the ABS could prevent the paragraph p_j from being sampled as a (false-) negative sample of the question q_i.

The proposed ABS algorithm is applied before each epoch while training the dual encoder. First, the being-trained model is used to encode questions and paragraphs for relevance score calculation. Then, the inner product is applied to the encoded vectors of a question and a paragraph to get the similarity between them. To reduce the batch scheduling time, only the top 100 paragraphs and their relevance scores are retrieved for each question and used for the scheduling. We used FAISS [18] for fast indexing and maximum inner-product searching of paragraphs.

Our implementation of the proposed ABS algorithm takes about 1 hour to schedule 532,209 MSMARCO training instances with batch size 2048. It takes 45 minutes to get the relevance scores using four Tesla V100 GPUs and 15 minutes to generate scheduled batches using one Intel Xeon(R) Gold 5120 CPU.

IV. EXPERIMENTS

In this section, we describe the evaluation results of the proposed adaptive batch scheduling method.

A. DATASETS

We evaluated the proposed method using two well-known document retrieval benchmark datasets, MSMARCO [12] Passage Ranking and Natural Questions [13]. Table 1 shows the statistics of the datasets.

TABLE 1. Benchmark data statistics. #q and #p means the number of questions and paragraphs, respectively.

Datasets	#q (train)	#q (dev)	#q (test)	#p
MSMARCO	502,939	6,980	6,837	8,841,823
Natural Questions	58,812	-	3,610	21,015,324

The questions of MSMARCO Passage Ranking dataset were gathered from Bing search logs. For each question, paragraphs which contain answers for the question are marked. The dataset contains 8.8 million paragraphs in total; the goal is to find answer-containing paragraphs for a given question. Since the test set of the MSMARCO Passage Ranking dataset is hidden, we used the MSMARCO dev set for testing. For the development set, we randomly choose 512 questions from the training samples. We refer to the 512 questions used to validate the model during training as MSMARCO custom dev set and the original 6,980 development questions as MSMARCO dev set.

Natural Questions benchmark dataset is first introduced by [13]. Questions of the Natural Questions dataset are collected from Google search logs, and answers for the questions are extracted from Wikipedia. [11] processed all the Wikipedia articles to get 21 million paragraphs in total. They also discarded some questions if the original answer passages failed to match the newly processed paragraphs. [6] used the processed dataset of [11] to populate hard negative examples on each question. In our experiments, we used the datasets with negative samples populated by [6]. 512 questions are randomly chosen from the training questions and used as a development set.

B. EXPERIMENTAL SETTINGS

Mean reciprocal rank (MRR) and recall for top n ranks are used for evaluation metrics, following previous work. Reciprocal rank is the multiplicative inverse of the rank of the first relevant passage; MRR is the average of reciprocal ranks across test questions. Recall for top n ranks is the ratio of questions whose answer passages are contained in top n retrieved documents. During the evaluation, top n paragraphs are retrieved from the whole paragraph pool for each evaluation question.

We set the batch size of 2048 for MSMARCO and 1024 for Natural Questions using four Tesla V100 GPUs. We used gradient checkpointing [19] to fit the large batches into GPU memories. The initial learning rate is experimentally set to $3e^{-5}$. It is selected through grid search with manually chosen values $\{1e^{-5}, 2e^{-5}, 3e^{-5}, 4e^{-5}\}$. Exponential decay with a decay rate of 0.8 is applied to the learning rate for every epoch. The performance on the development set is evaluated after each training epoch; the training stops if the dev set performance does not increase for five consequent epochs.

C. EVALUATION RESULTS

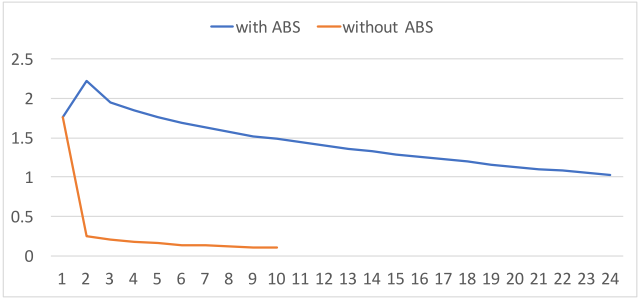
We applied the proposed adaptive batch scheduler to train dual encoders for the two benchmark datasets. For each question, only its positive paragraphs are used as training instances; negative samples are retrieved only from in-batch negative sampling. It is for separately observing the effect of ABS on in-batch negative sampling from the negative samples populated before training. We trained three baseline dual encoders with different pre-trained language models, namely BERT_{base} [7], RoBERTa_{base} [20], and ERNIE_{base} [21].

Table 2 shows the evaluation result. As can be observed from the table, systems trained with ABS outperform those trained without ABS for both benchmark datasets. For the MSMARCO dataset, the proposed ABS increases MRR@10 by 2 to 3 % for all baseline systems. The ABS also increases R@5 for the natural questions dataset from 4 to 6 %. The evaluation result suggests that the application of ABS during training could significantly improve the document retrieval performance of dual encoders, regardless of the pre-trained language models used.

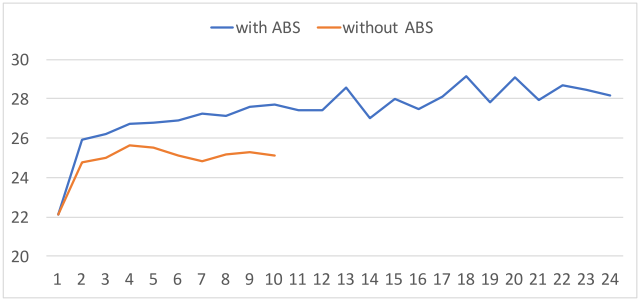
We tried to find out the effect of ABS on the training process in more detail. Figure 3 shows the training loss and custom dev set MRR@10 for each training epoch during ERNIE_{base} training. The ABS is applied before the start of every training epoch, except epoch 1. The training loss decreases much slower when the ABS is applied; the ABS provides negative examples that are

TABLE 2. Evaluation results of the ABS on benchmark datasets. Note that for the MSMARCO dataset, we used the MSMARCO dev set for testing. MRR represents mean reciprocal rank, and R represents recall.

Pre-trained language model	ABS	MSMARCO dev			Natural Questions test		
		MRR@10	R@50	R@1000	R@5	R@20	R@100
BERT_{base}	X	25.9	76.2	**95.8**	53.6	71.6	83.0
	O	**27.8**	**77.4**	95.6	**57.6**	**73.1**	**83.2**
RoBERTa_{base}	X	24.3	75.2	96.0	49.1	67.1	80.1
	O	**27.6**	**78.7**	**96.5**	**55.7**	**71.5**	**81.9**
ERNIE_{base}	X	26.2	77.2	96.2	55.4	72.2	83.4
	O	**28.7**	**80.1**	**96.8**	**60.6**	**75.2**	**84.0**



(a) Training loss



(b) Custom development set MRR@10

FIGURE 3. MSMARCO training loss and MRR@10 of custom dev set during training for ERNIE_{base}. The horizontal axis represents the training epoch.

considered difficult by the being-trained encoder before each training epoch. Instead, the custom dev set MRR@10 is much higher when the ABS is applied; it constantly outperforms the case without ABS to 1 to 2 % with the same number of training epochs and takes more epochs to reach the maximal dev MRR@10. In summary, the ABS trains dual encoder models much better by constantly providing hard, confusing negative examples for the being-trained model.

[6] showed that the training batch size is an important parameter when applying the in-batch negative sampling. The larger batch size means more negative samples to train. We tried to see the effect of different training batch sizes on the proposed adaptive batch scheduling. Baseline model ERNIE_{base} is trained with different batch sizes. Table 3 shows the evaluation results.

As can be observed from the table, the document retrieval performance drops as the batch size decreases when the ABS is not used. This result is the same as reported in [6]. Interestingly, the retrieval performance does not drop as the batch size decreases when the ABS is applied for training. The evaluation result suggests that providing a small number of difficult negative examples is better than a large number of random negative samples for training. Since the ABS helps to get the hard negative samples, the dual encoder trainer does not need to have a large training batch size to enhance the document retrieval performance.

TABLE 3. The effect of different training batch sizes on the proposed adaptive batch scheduling.

Batch Size	ABS	MSMarco dev		
		MRR@10	R@50	R@1000
512	X	24.7	74.5	95.4
	O	**29.4**	**79.6**	**96.5**
1024	X	26.2	77.7	96.2
	O	**28.7**	**79.6**	**96.7**
2048	X	26.2	77.2	96.2
	O	**28.7**	**80.1**	**96.8**

The proposed ABS maintains positively annotated paragraphs for each training question to filter out the noises introduced by a question with multiple answer paragraphs or multiple questions sharing the same answer paragraph. Table 4 shows the evaluation results with and without the noise filtering on the MSMARCO dataset. Baseline model ERNIE_{base} is trained with the ABS. As can be observed from the table, the noise filtering slightly improves the performance of the proposed ABS.

TABLE 4. The effect of positive noise filtering for the adaptive batch scheduling on ERNIE_{base} model.

Apply noise filtering	MSMarco dev		
	MRR@10	R@50	R@1000
X	28.4	79.3	96.5
O	**28.7**	**80.1**	**96.8**

Finally, we tried to see the effect of ABS when the separately populated negative examples are present. We used the negative samples proposed in [6]. The random negative

TABLE 5. Evaluation results of the ABS applied on the training set containing negative samples. `Random` and `Hard` represent the dataset with negative samples populated using random negative and hard negative populations, respectively. `None` represents the dataset with no negative samples.

Negative Population	ABS	MSMarco dev		
		MRR@10	R@50	R@1000
`None`	X	26.2	77.2	96.2
	O	**28.7**	**80.1**	**96.8**
`Random`	X	30.4	80.5	**96.9**
	O	**32.6**	**81.4**	96.5
`Hard`	X	33.6	81.6	**97.3**
	O	**34.8**	**82.4**	96.8

population approach selects random paragraphs from the document pool as negative samples for a question. The hard negative population approach first trains a dual encoder and a cross encoder using the training data with negative samples populated using random negative population. Then, for each question, top k paragraphs are retrieved using the dual encoder, and each paragraph is verified with the cross encoder to get the hard negative samples for the question. We downloaded the negative samples populated using the two population approaches from the author's homepage.[2] The dataset contains four negative paragraphs for each question.

Table 5 shows the evaluation results with populated negative samples. We trained ERNIE$_{base}$ with the populated datasets. As can be observed, The performance gains due to the ABS are similar for the random negative populated dataset and the dataset without negative samples. Meanwhile, the effect of ABS decreases with the hard negative populated dataset; since the hard negatives are separately provided in the training dataset itself, the effect of hard negatives provided during the in-batch negative sampling is limited. Still, the application of ABS on training the dual encoder with a hard-negative populated training dataset increases the MRR@10 by 1.2%.

V. CONCLUSION

In this paper, we proposed adaptive batch scheduling to effectively sample hard negatives for in-batch negative sampling during dual encoder training. The proposed ABS is evaluated against two document retrieval benchmark datasets, MSMARCO and Natural Questions. The evaluation results showed that our proposed ABS significantly improves the document retrieval performances of dual encoders when applied during training.

[6] showed that denoising the populated negative samples are vital for performance improvement. Although the proposed ABS improved the overall retrieval performance, we think the system can be further improved by denoising the sampled negatives during training. Our future work will focus on filtering the false negatives sampled by the in-batch negative sampling.

[2]https://github.com/PaddlePaddle/Research/tree/master/NLP/NAACL2021-RocketQA

REFERENCES

[1] D. Chen, A. Fisch, J. Weston, and A. Bordes, "Reading Wikipedia to answer open-domain questions," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, 2017, pp. 1870–1879.
[2] R. Nogueira, W. Yang, J. Lin, and K. Cho, "Document expansion by query prediction," 2019, *arXiv:1904.08375*. [Online]. Available: http://arxiv.org/abs/1904.08375
[3] R. Nogueira, J. Lin, and A. I. Epistemic, "From doc2query to docTTTTTquery," Online Preprint, Tech. Rep., 2019. [Online]. Available: https://cs.uwaterloo.ca/~jimmylin/publications/Nogueira_Lin_2019_docTTTTTquery.pdf
[4] Z. Dai and J. Callan, "Context-aware term weighting for first stage passage retrieval," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2020, pp. 1533–1536.
[5] Y. Mao, P. He, X. Liu, Y. Shen, J. Gao, J. Han, and W. Chen, "Generation-augmented retrieval for open-domain question answering," 2020, *arXiv:2009.08553*. [Online]. Available: http://arxiv.org/abs/2009.08553
[6] Y. Qu, Y. Ding, J. Liu, K. Liu, R. Ren, W. X. Zhao, D. Dong, H. Wu, and H. Wang, "RocketQA: An optimized training approach to dense passage retrieval for open-domain question answering," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2021, pp. 5835–5847.
[7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*. [Online]. Available: http://arxiv.org/abs/1810.04805
[8] K. Lee, M.-W. Chang, and K. Toutanova, "Latent retrieval for weakly supervised open domain question answering," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 6086–6096.
[9] W.-C. Chang, F. X. Yu, Y.-W. Chang, Y. Yang, and S. Kumar, "Pre-training tasks for embedding-based large-scale retrieval," in *Proc. Int. Conf. Learn. Represent.*, 2020, pp. 1–12.
[10] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M.-W. Chang, "REALM: Retrieval-augmented language model pre-training," 2020, *arXiv:2002.08909*. [Online]. Available: http://arxiv.org/abs/2002.08909
[11] V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W.-T. Yih, "Dense passage retrieval for open-domain question answering," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2020, pp. 6769–6781.
[12] T. Nguyen, M. Rosenberg, X. Song, J. Gao, S. Tiwary, R. Majumder, and L. Deng, "MS MARCO: A human generated machine reading comprehension dataset," in *Proc. Workshop Cognit. Comput., Integr. Neural Symbolic Approaches Colocated 30th Annu. Conf. Neural Inf. Process. Syst. (NIPS)*, 2016, pp. 1–10.
[13] T. Kwiatkowski, J. Palomaki, O. Redfield, M. Collins, A. Parikh, C. Alberti, D. Epstein, I. Polosukhin, J. Devlin, K. Lee, K. Toutanova, L. Jones, M. Kelcey, M.-W. Chang, A. M. Dai, J. Uszkoreit, Q. Le, and S. Petrov, "Natural questions: A benchmark for question answering research," *Trans. Assoc. Comput. Linguistics*, vol. 7, pp. 453–466, Aug. 2019.
[14] S. Robertson and H. Zaragoza, "The probabilistic relevance framework: BM25 and beyond," *Found. Trends Inf. Retr.*, vol. 3, no. 4, pp. 333–389, 2009.
[15] O. Khattab and M. Zaharia, "ColBERT: Efficient and effective passage search via contextualized late interaction over bert," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2020, pp. 39–48.
[16] L. Gao, Z. Dai, and J. Callan, "COIL: Revisit exact lexical match in information retrieval with contextualized inverted list," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2021, pp. 3030–3042.
[17] Y. Freund, R. E. Schapire, and N. Abe, "A short introduction to boosting," *J. Jpn. Soc. Artif. Intell.*, vol. 14, no. 5, pp. 771–780, 1999.
[18] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with GPUs," 2017, *arXiv:1702.08734*. [Online]. Available: http://arxiv.org/abs/1702.08734

[19] T. Chen, B. Xu, C. Zhang, and C. Guestrin, "Training deep nets with sublinear memory cost," 2016, *arXiv:1604.06174*. [Online]. Available: http://arxiv.org/abs/1604.06174
[20] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A robustly optimized BERT pretraining approach," 2019, *arXiv:1907.11692*. [Online]. Available: http://arxiv.org/abs/1907.11692
[21] Z. Zhang, X. Han, Z. Liu, X. Jiang, M. Sun, and Q. Liu, "ERNIE: Enhanced language representation with informative entities," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 1441–1451.

DONGHYUN CHOI received the B.S. degree in computer science from Korean Advanced Institute of Science and Technology, South Korea, in 2006. He is currently pursuing the Ph.D. degree in computer science with Sungkyunkwan University, Suwon, South Korea. He is currently working as a Researcher with Kakao Enterprise, Pangyo, South Korea.

MYEONGCHEOL SHIN received the B.S. degree in computer science and engineering from POSTECH, South Korea, in 2006. He is currently leading a group of NLP Researchers, Kakao Enterprise, Pangyo, South Korea. His research interests include entity extraction, dialog engine, language modeling, and transfer learning.

EUNGGYUN KIM is currently leading the NLP Team, Kakao Enterprise, Pangyo, South Korea. He is also working on Korean natural language processing systems. His research interests include chatbots, sentiment analysis, and open-domain question answering.

DONG RYEOL SHIN is currently a Professor with the College of Information and Communication Engineering, Sungkyunkwan University, South Korea. Currently, he does research on big data and its applications with a particular emphasis on platform implementation that is the basis for big data analytics and performance analysis. It includes healthcare platform based on cloud, in-vehicle e-call system, educational platform buildup and demonstration for big data analytics, and healthcare data analysis. He is involved in the development of scheduling algorithm, as well as protocol analysis, performance evaluation from the wireless network viewpoint. His research interests include distributed systems, middleware, mobile computing, wireless networks, and communication systems.