Improving Dense Passage Retrieval with Multiple Positive Passages

Anonymous ACL submission

Abstract

By leveraging a dual encoder architecture, Dense Passage Retrieval (DPR) has outperformed traditional sparse retrieval algorithms such as BM25 in terms of passage retrieval accuracy. Recently proposed methods have further enhanced DPR's performance. However, these models typically pair each question with only one positive passage during training, and the effect of associating multiple positive passages has not been examined. In this paper, we explore the performance of DPR when additional positive passages are incorporated during training. Experimental results show that equipping each question with multiple positive passages consistently improves retrieval accuracy, even when using a significantly smaller batch size, which enables training on a single GPU.

1 Introduction

In information retrieval (IR), passage retrieval refers to the task of retrieving text segments or passages that are relevant to a given query. Due to its ability to narrow down the searching scope, passage retrieval has become a key component in open-domain question answering (QA) and web search engines. Traditional methods such as TF-IDF and BM25 (Robertson and Walker, 1997) rely on term frequency to measure text relevance, but lack the ability to capture the semantic meaning of sentences. This limitation could lead to poor performance when relative contents are composed of entirely different tokens (Karpukhin et al., 2020).

Pre-trained language models, such as BERT (Devlin et al., 2019) and T5 (Raffel et al., 2020), have significantly enhanced text representation learning and demonstrated superior performance in IR tasks (Nogueira and Cho, 2020; Ni et al., 2022). The Dense Passage Retrieval (DPR) model (Karpukhin et al., 2020) employs a dual-BERT encoder architecture to independently encode questions and passages into dense vector representations and utilizes in-batch negatives to improve training efficiency. DPR outperforms traditional sparse retrieval methods like BM25, thereby boosting the performance of end-to-end QA systems and retrieval-augmented generation models (Lewis et al., 2020). While several refinement methods have been proposed, including optimized training strategies (Qu et al., 2021; Ren et al., 2021b), enhanced similarity measurements (Ren et al., 2021a), and improved training efficiency (Hofstätter et al., 2021), none have investigated the impact of associating multiple positive passages with each question during training.

In this paper, we focus on pairing multiple positive passages with each question when training the dual BERT encoder. The intuition is straightforward: we hypothesize that the dominant number of negative passages during training (e.g., 1 positive vs. 255 negatives per question in DPR) may erode the model's ability to identify relevant passages at inference time. By exposing the model to more positive passages, we reformulate training as a binary classification task, where the model learns to distinguish between positive and negative passages under a smaller positive-negative imbalance. Experimental results on several QA datasets consistently show that our method improves retrieval accuracy while significantly reducing the required batch size, enabling the model to be trained on a single GPU.

2 Methodology

This section presents the method that incorporates multiple positive passages with each question to train the DPR model. As a preliminary step, we first introduce some background of the DPR model.

2.1 The DPR model

DPR uses two separate encoders, $E_P(\cdot)$ and $E_Q(\cdot)$, to map text passages and questions to a shared

vector space. Both encoders are based on the BERT (base, uncased) model, and the representation at the $[CLS]$ token is used as the output.

The training objective is to ensure that the distance between relevant question-passage pairs is smaller than the irrelevant ones. This distance (or similarity) between a question and a passage is measured by the inner product of their vector representations:

$$\text{sim}(q, p) = E_Q(q)^\top E_P(p) \tag{1}$$

Suppose there is one positive (relevant) passage p_i^+ and n negative (irrelevant) passages $p_{i,j}^-$, where $j \in \{1, 2, \ldots, n\}$, for each question q_i. DPR is then optimized using the negative log likelihood (NLL) of the positive passage:

$$\mathcal{L}(q_i, p_i^+, p_{i,1}^-, \ldots, p_{i,n}^-) = -\log \frac{e^{\text{sim}(q_i, p_i^+)}}{e^{\text{sim}(q_i, p_i^+)} + \sum_{j=1}^{n} e^{\text{sim}(q_i, p_{i,j}^-)}}. \tag{2}$$

During the actual training process, each question is paired with one positive passage and one hard negative passage. By taking advantage of the in-batch negatives trick (Yih et al., 2011), all positive and negative passages associated with other questions are treated as negative passages for the current question, enabling efficient computation and improved performance. Assuming the batch size is B, for each question-passage training sample, the ratio of positive passages to negative passages is roughly $1/2B$. In the next subsection, we discuss the differences that after pairing each question with multiple positive passages.

2.2 Pairing each question with more positive passages

Instead of a single positive passage, we pair each question with $m > 1$ positive passages to train the model. After applying in-batch training, there are $(m + 1) \times (B - 1) + 1$ negative passages for each question, and the ratio of positive to negative passages is approximately $1/B$, doubling the proportion compared to the original DPR model. We hypothesize that this provides the model with more positive feedback during training, which could be beneficial for improving performance.

Since more positive passages are introduced, we treat the training process as a binary classification task, where the model is expected to judge each passage as either positive or negative with respect to a question. To optimize the model, we discard the NLL loss formulated in Eq. (2) and instead use binary cross-entropy (BCE) loss:

$$\begin{aligned}\mathcal{L}(q_i, p_{i,1}^+, \ldots, p_{i,m}^+, p_{i,1}^-, \ldots, p_{i,n}^-) = \\ -\sum_{k=1}^{m} \log \sigma\left(\text{score}(q_i, p_{i,k}^+)\right) \\ -\sum_{j=1}^{n} \log\left(1 - \sigma\left(\text{score}(q_i, p_{i,j}^-)\right)\right)\end{aligned} \tag{3}$$

In Eq. (3), $\text{score}(q, p)$ refers to the softmax-scaled inner product similarity $\text{sim}(q, p)$, and $\sigma(\cdot)$ is the sigmoid function. These configurations are used to stabilize the training process.

3 Experimental setup

This describes the data used in our experiments and the training configurations.

3.1 Data preparing

The training datasets and source documents are the same as those used in (Karpukhin et al., 2020). The source documents are constructed using Wikipedia English articles (Dec 20, 2018 dump). These documents consist of 21,015,324 passages, with each passage containing 100 words. Details of the training datasets are provided below.

CuratedTREC (TREC) (Baudiš and Šedivý, 2015) is an improved QA training and benchmark dataset derived from the TREC QA tracks. Some answers are expressed using regular expression patterns.

WebQuestions (WebQ) (Berant et al., 2013) was crafted using Google Suggest API, and all questions begin with a wh-word.

SQuAD 1.1 (Rajpurkar et al., 2016) contains 107,785 question-answer pairs derived from 536 Wikipedia articles via crowdsourcing.

TriviaQA (Joshi et al., 2017) is a reading comprehension dataset consisting of 95,000 trivia questions. Each question is associated with six evidence documents on average.

Natural Question (NQ) (Kwiatkowski et al., 2019) is a real-world question answering benchmark dataset with questions mined from Google search queries and answers annotated from Wikipedia articles.

We retain the passage selection strategy as demonstrated in (Karpukhin et al., 2020), and dis-

card samples that could cause the training process failure. Table 1 lists the actual number of questions in each dataset for training the model.

Dataset	questions used for training
TREC	1,117
WebQ	2,448
SQUAD	70,096
TriviaQA	60,368
NQ	58,880

Table 1: Questions used for training the model in each dataset.

3.2 Training setup

We pair each question with up to 3 positive passages and 1 hard negative passage, and use the in-batch negative trick to train the model. Since some questions have fewer than 3 associated positive passages, we dynamically assign the maximum available number of positives for those cases. The batch size is set to 16, with 100 training epochs for TREC and WebQ, and 40 for SQuAD, TriviaQA, and NQ. We use the Adam optimizer with a learning rate of 10^{-5}, linear scheduling with warm-up, and a dropout rate of 0.1. All experiments were conducted on a single NVIDIA RTX 5080 GPU with 16 GB of VRAM.

4 Experiment results

This section demonstrates the evaluation results of the proposed model along with analysis of its effectiveness.

4.1 Main results

The top k ($k \in \{20, 100\}$) retrieval accuracy of different models are shown in Table 2. DPR$^+$ denotes our proposed model, while the others are from (Karpukhin et al., 2020). Single and Multi indicate whether the model was trained on individual or combined datasets (all but except SQuAD). BM25 + DPR is a linear combination of the BM25 and DPR models, as described in (Karpukhin et al., 2020).

The results show that DPR$^+$ achieves the best performance on the NQ, TriviaQA, and WebQ datasets, even without multi-dataset training or BM25 model combination. Notably, our model outperforms the Single DPR baseline on all datasets except SQuAD, clearly confirming its effectiveness. Additionally, thanks to a significantly smaller batch size, our model can be trained on a single GPU with 16 GB of VRAM, whereas training the original DPR model with a batch size of 128 requires 8×32 GB GPUs. This highlights the improved efficiency of DPR$^+$.

We suspect that the low performance of DPR$^+$ on the SQuAD dataset is due to an inadequate number of positive passages. Based on the number of positive passages associated with each question, we classified the questions into three groups, as shown in Table 3. In the table, p_1 denotes the number of questions paired with only one positive passage; the same terminology applies to p_2 and p_3. The symbol δ represents the proportion of p_3 in the total. As illustrated in the table, only 49.1% of the questions in the SQuAD dataset are paired with three positive passages. This data deficiency may hinder DPR$^+$ from fully exploiting the benefits of multiple positive passages during training.

4.2 Ablation study

We selected the TREC dataset and varied the maximum number of positive passages per question to 1, 2, and 3, respectively, to examine the impact of positive passages on performance. We also included the results of DPR-Single (Karpukhin et al., 2020) trained with a batch size of 16 to assess the influence of using the BCE training loss. For simplicity, we reused the same encoded source files from our previous experiments. The final results are presented in Table 4.

In Table 4, DPR$^+_1$, DPR$^+_2$, and DPR$^+_3$ represent DPR$^+$ trained with 1, 2, and 3 positive passages associated with each question, respectively. We observe that performance improves as more positive passages are incorporated, and the difference between DPR-Single and DPR$^+_1$ is relatively small. The improvement becomes more pronounced in terms of top-100 accuracy. These results suggest that the BCE loss has only a subtle impact on performance, whereas pairing each question with more positive passages leads to clear performance gains.

5 Conclusion

In this paper, we present a simple yet effective strategy that pairs multiple positive passages with each question to enhance the DPR model. By formulating training as a binary classification task, where each passage is judged as positive or negative, the model is optimized using the BCE loss. Empirical results demonstrate that the proposed method consistently improves retrieval accuracy while sig-

Training	Retriever	Top-20					Top-100				
		NQ	TriviaQA	WQ	TREC	SQuAD	NQ	TriviaQA	WQ	TREC	SQuAD
None	BM25	59.1	66.9	55.0	70.9	68.8	73.7	76.7	71.1	84.1	80.0
Single	DPR	78.4	79.4	73.2	79.8	63.2	85.4	85.0	81.4	89.1	77.2
	BM25 + DPR	76.6	79.8	71.0	85.2	**71.5**	83.8	84.5	80.5	92.7	**81.3**
	DPR$^+$	**80.4**	**79.9**	**76.5**	83.6	52.7	**86.8**	**85.6**	**83.4**	92.7	69.2
Multi	DPR	79.4	78.8	75.0	**89.1**	51.6	86.0	84.7	82.9	93.9	67.6
	BM25 + DPR	78.0	**79.9**	74.7	88.5	66.2	83.9	84.4	82.3	**94.1**	78.6

Table 2: Top-20 and Top-100 retrieval accuracy across test datasets. The accuracy is calculated as the percentage of top $20/100$ retrieved passages that contain the answer. *Single* and *Multi* denote that the retriever was trained using one or combined datasets (all excluding SQuAD). Bold numbers indicate the best performance.

Dataset	δ	question count			
		p_1	p_2	p_3	total
TREC	82.1%	89	111	920	1,120
WebQ	74.1%	365	273	1,826	2,464
SQUAD	49.1%	14,842	20,834	34,420	70,096
TriviaQA	79.5%	6,336	6,029	48,035	60,400
NQ	67.7%	9,577	9,455	39,848	58,880

Table 3: Question counts with respect to the number of positive passages associated with each question in the training datasets. p_1, p_2, and p_3 denote the number of questions paired with 1, 2, and 3 positive passages, respectively. δ represents the proportion of p_3 in the total number of questions.

Model	Top-20 accuracy	Top-100 accuracy
DPR-Single	80.8	89
DPR$_1^+$	80.4	89.6
DPR$_2^+$	83.9	91.6
DPR$_3^+$	83.6	92.7

Table 4: Top-20 and Top-100 retrieval accuracy for different methods. DPR$_i^+$ denotes the DPR$^+$ model trained with i positive passages paired with each question. The results for DPR-Single are reported using a batch size of 16 during training.

nificantly reducing the hardware requirements for training.

Limitations

Due to hardware constraints, we were only able to train the model with up to three positive passages per question and a batch size of 16. The effects of using more positive passages or larger batch sizes remain unexplored. Further investigation is required to understand the trade-off between batch size and the number of positive passages for optimal performance.

Moreover, while our method is simple and easy to implement, it may have limitations in further improving retrieval accuracy compared to more sophisticated approaches.

Acknowledgments

The author would like to thank anyone who offered constructive suggestions.

References

Petr Baudiš and Jan Šedivý. 2015. Modeling of the question answering task in the yodaqa system. In *Proceedings of the 6th International Conference on Experimental IR Meets Multilinguality, Multimodality, and Interaction - Volume 9283*, CLEF'15, page 222–228, Berlin, Heidelberg. Springer-Verlag.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on Freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544, Seattle, Washington, USA. 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.

Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. 2021. Efficiently teaching an effective dense retriever with balanced topic aware sampling. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 113–122, New York, NY, USA. Association for Computing Machinery.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernandez Abrego, Ji Ma, Vincent Zhao, Yi Luan, Keith Hall, Ming-Wei Chang, and Yinfei Yang. 2022. Large dual encoders are generalizable retrievers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9844–9855, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Rodrigo Nogueira and Kyunghyun Cho. 2020. Passage re-ranking with bert. *Preprint*, arXiv:1901.04085.

Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. 2021. RocketQA: An optimized training approach to dense passage retrieval for open-domain question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5835–5847, Online. 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. *J. Mach. Learn. Res.*, 21(1).

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Ruiyang Ren, Shangwen Lv, Yingqi Qu, Jing Liu, Wayne Xin Zhao, QiaoQiao She, Hua Wu, Haifeng Wang, and Ji-Rong Wen. 2021a. PAIR: Leveraging passage-centric similarity relation for improving dense passage retrieval. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2173–2183, Online. Association for Computational Linguistics.

Ruiyang Ren, Yingqi Qu, Jing Liu, Wayne Xin Zhao, QiaoQiao She, Hua Wu, Haifeng Wang, and Ji-Rong Wen. 2021b. RocketQAv2: A joint training method for dense passage retrieval and passage re-ranking. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2825–2835, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

S. E. Robertson and S. Walker. 1997. *Some simple effective approximations to the 2–Poisson model for probabilistic weighted retrieval*, page 345–354. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Wen-tau Yih, Kristina Toutanova, John C. Platt, and Christopher Meek. 2011. Learning discriminative projections for text similarity measures. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, pages 247–256, Portland, Oregon, USA. Association for Computational Linguistics.