Augmenting Document Representations for Dense Retrieval with Interpolation and Perturbation

Anonymous ACL submission

Abstract

Dense retrieval models, which aim at retrieving the most relevant document for an input query on a dense representation space, have gained considerable attention for their remarkable success. Yet, dense models require a vast amount of labeled training data for notable performance, whereas it is often challenging to acquire query-document pairs annotated by humans. To tackle this problem, we propose a simple but effective **D**ocument **A**ugmentation for dense **R**etrieval (DAR) framework, which augments the representations of documents with their interpolation and perturbation. We validate the performance of DAR on retrieval tasks with two benchmark datasets, showing that the proposed DAR significantly outperforms relevant baselines on the dense retrieval of both the labeled and unlabeled documents.

1 Introduction

Retrieval systems aim at retrieving the documents most relevant to the input queries, and have received substantial spotlight since they work as core elements in diverse applications, especially for open-domain question answering (QA) (Voorhees, 1999). Open-domain QA is a task of answering the question from a massive amount of documents, often requiring two components, a retriever and a reader (Chen et al., 2017; Karpukhin et al., 2020). Specifically, a retriever ranks the most question-related documents, and a reader answers the question using the retrieved documents.

Traditional sparse retrieval approaches such as BM25 (Robertson et al., 1994) and TF-IDF rely on term-based matching, hence suffering from the vocabulary mismatch problem: the failure of retrieving relevant documents due to the lexical difference from queries. To tackle such a problem, recent research focuses on dense retrieval models to generate learnable dense representations for queries and documents with a dual encoder structure (Karpukhin et al., 2020; Xiong et al., 2021).

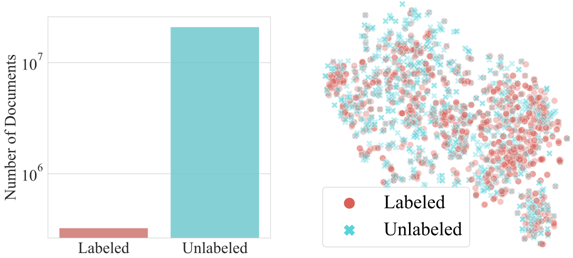

Figure 1: (Left) A number of labeled and unlabeled documents for the Natural Question dataset. (Right) T-SNE (Maaten and Hinton, 2008) visualization of randomly sampled document representations from the DPR model.

Despite their recent successes, some challenges still remain in the dense retrieval scheme for a couple of reasons. First, dense retrieval models need a large amount of labeled training data for a decent performance. However, as Figure 1 shows, the proportion of labeled query-document pairs is extremely small since it is almost impossible to rely on humans for the annotations of a large document corpus. Second, in order to adapt a retrieval model to the real world, where documents constantly emerge, handling unlabeled documents that are not seen during training should obviously be considered, but remains challenging.

To automatically expand the query-document pairs, recent work generates queries from generative models (Liang et al., 2020; Ma et al., 2021) or incorporates queries from other datasets (Qu et al., 2021), and then generates extra pairs of augmented queries and documents. However, these query augmentation schemes have serious obvious drawbacks. First, it is infeasible to augment queries for every document in the dataset (see the number of unlabeled documents in Figure 1), since generating and pairing queries are quite costly. Second, even after obtaining new pairs, we need extra training steps to reflect the generated pairs on the retrieval model. Third, this query augmentation method does not add variations to the documents but only to the queries, thus it may be suboptimal to handle enormous unlabeled documents.

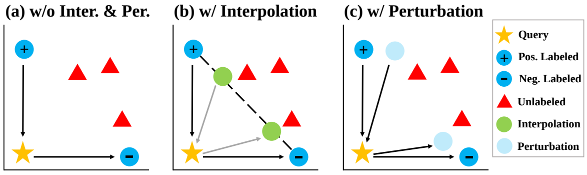

Figure 2: Our document augmenting schemes of interpolation and perturbation on a dense representation space. Pos. and Neg. denote positive and negative documents to the query.

Since augmenting additional queries is costly, the question is then if it is feasible to only manipulate the given query-document pairing to handle numerous unlabeled documents. To answer this question, we first visualize the embeddings of labeled and unlabeled documents. Figure 1 shows that there is no distinct distributional shift between labeled and unlabeled documents. Thus it could be effective to manipulate only the labeled documents to handle the nearby unlabeled documents as well as the labeled documents. Using this observation, we propose a novel document augmentation method for a dense retriever, which not only interpolates two different document representations associated with the labeled query (Figure 2, center), but also stochastically perturbs the representations of labeled documents with a dropout mask (Figure 2, right). One notable advantage of our scheme is that, since it manipulates only the representations of documents, our model does not require explicit annotation steps of query-document pairs, which is efficient. We refer to our overall method as Document Augmentation for dense Retrieval (DAR).

We experimentally validate our method on standard open-domain QA datasets, namely Natural Question (NQ) (Kwiatkowski et al., 2019) and TriviaQA (Joshi et al., 2017) (TQA), against various evaluation metrics for retrieval models. The experimental results show that our method significantly improves the retrieval performances on both the unlabeled and labeled documents. Furthermore, a detailed analysis of the proposed model shows that interpolation and stochastic perturbation positively contribute to the overall performance.

Our contributions in this work are threefold:

- We propose to augment documents for dense retrieval models to tackle the problem of insufficient labels of query-document pairs.
- We present two novel document augmentation schemes for dense retrievers: interpolation and perturbation of document representations.
- We show that our method achieves outstanding retrieval performances on both labeled and unlabeled documents on open-domain QA tasks.

2 Related Work

Dense Retriever Dense retrieval models (Lee et al., 2019; Karpukhin et al., 2020) have gained much attention, which generate dense representations for queries and documents. However, dense retrieval faces a critical challenge from limited training data. Recent work has addressed such a problem by generating extra query-document pairs to augment those pairs to the original dense retrieval model (Liang et al., 2020; Ma et al., 2021; Qu et al., 2021), or by regularizing the model (Rosset et al., 2019). However, unlike ours that automatically augments data during a training phase, these methods require extensive computational resources for an additional generation step of explicitly query-document pairing before retriever's training.

Data Augmentation Since enlarging the volume of training data is crucial to the performance of deep neural networks, data augmentation is applied to diverse domains (Shorten and Khoshgoftaar, 2019; Hedderich et al., 2021), where interpolation and perturbation are dominant methods. Mixup interpolates two different items, such as pixels of images, to augment the training data (Zhang et al., 2018; Verma et al., 2019), which is also recently adopted for NLP tasks (Chen et al., 2020a; Yin et al., 2021). However, none of the previous work has shown the effectiveness of mixup when applied to retrieval tasks. Besides interpolation, Wei and Zou (2019) and (Ma, 2019) proposed perturbation over words, and Lee et al. (2021) proposed perturbation over word embeddings. Jeong et al. (2021) perturbed document embeddings to generate diverse summaries. In contrast, we address the dense retrieval task by perturbing document representations with dropout (Srivastava et al., 2014).

3 Method

We begin with the definition of dense retrieval.

Dense Retrieval Given a pair of query q and document d, the goal of dense retrieval is to correctly calculate a similarity score between them from the dense representations $\boldsymbol{q}$ and $\boldsymbol{d}$, as follows:

$$\begin{aligned} &f(q, d) = \text{sim}(\boldsymbol{q}, \boldsymbol{d}), \\ &\boldsymbol{q} = E_Q(q; \theta_q) \quad \text{and} \quad \boldsymbol{d} = E_D(d; \theta_d), \end{aligned} \tag{1}$$

where f is a scoring function that measures the similarity between a query-document pair, sim is a similarity metric such as cosine similarity, and E_Q and E_D are dense encoders for a query and document, respectively, with parameters $\theta = (\theta_q, \theta_d)$.

	Natural Questions (NQ)						TriviaQA (TQA)					
	MRR	MAP	T-100	T-20	T-5	T-1	MRR	MAP	T-100	T-20	T-5	T-1
BM25	32.46	20.78	78.25	62.94	43.77	22.11	55.28	34.85	83.15	76.41	66.28	46.30
DPR	39.55	25.61	83.77	72.94	54.02	27.45	44.29	27.24	80.50	71.07	57.74	33.63
DPR w/ QA	40.00	24.93	83.46	72.13	55.46	27.67	46.27	28.08	80.76	71.88	59.14	35.90
DPR w/ DA	41.28	26.60	83.68	72.83	55.51	29.31	46.08	27.82	80.42	71.55	58.64	35.85
DPR w/ AR	41.18	26.04	83.60	73.41	55.51	29.11	45.13	27.57	80.65	71.68	58.09	34.52
DAR (Ours)	**42.92**	**27.12**	**84.18**	**75.04**	**57.62**	**30.42**	**47.32**	**28.70**	**81.30**	**72.66**	**59.88**	**36.94**

Table 1: Retrieval results on NQ and TQA datasets. BM25 is the sparse retrieval model, whereas others are dense retrieval models. The numbers in bold denote the best model among dense retrieval models, which we aim to improve in this work.

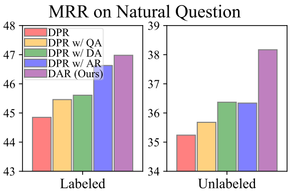

Figure 3: Retrieval results on the labeled and unlabeled documents in the NQ dataset.

# Query		MRR	R@1k
10K	ANCE	42.62	94.60
	+ DAR	**46.31**	**94.81**
50K	ANCE	46.88	**95.58**
	+ DAR	**48.20**	**95.58**

Table 2: Results on the MS MARCO subsets with ANCE as a denser retriever.

	Time (Min.)	Relative Time
DPR	19	1.00
DPR w/ QA	41	2.16
DPR w/ DA	38	2.00
DPR w/ AR	29	1.53
DAR (Ours)	21	1.11

Table 3: Wall-clock time and its relative time for training a DPR model per epoch.

A dense retrieval scheme generally uses the negative sampling strategy to distinguish the relevant query-document pairs from irrelevant pairs, which generates an effective representation space for queries and documents. We specify a relevant query-document pair as $(q, d^+) \in \tau^+$, and an irrelevant pair as $(q, d^-) \in \tau^-$, where $\tau^+ \cap \tau^- = \emptyset$. The objective function is as follows:

$$\min_{\theta} \sum_{(q,d^+)\in\tau^+} \sum_{(q,d^-)\in\tau^-} \mathcal{L}(f(q,d^+), f(q,d^-)), \quad (2)$$

where a loss function $\mathcal{L}$ is a negative log-likelihood of the positive document. Our goal is to augment a set of query-document pairs, by manipulating documents with their interpolation or perturbation, which we explain in the next paragraphs.

Interpolation with Mixup As shown in interpolation of Figure 2, we aim at augmenting the document representation located between two labeled documents to obtain more query-document pairs, which could be useful to handle unlabeled documents in the middle of two labeled documents. To achieve this goal, we propose to interpolate the positive and negative documents (d^+, d^-) for the given query q, adopting mixup (Zhang et al., 2018). Note that, since the input documents to the encoder E_D are discrete, we use the output embeddings of documents to interpolate them, as follows:

$$\tilde{\boldsymbol{d}} = \lambda \boldsymbol{d}^+ + (1-\lambda)\boldsymbol{d}^-, \quad (3)$$

where $\tilde{\boldsymbol{d}}$ is the mixed representation of positive and negative documents for the given query q, and $\lambda \in [0, 1]$. We then optimize the model to estimate the similarity $\text{sim}(\boldsymbol{q}, \tilde{\boldsymbol{d}})$ between the interpolated document and the query as the soft label λ with a binary cross-entropy loss. The output of the cross-entropy loss is added to the original loss in equation 2. One notable advantage of our scheme is that the negative log-likelihood loss in equation 2 maximizes the similarity score of the positive pair, while minimizing the score of the negative pair; thus there are no intermediate similarities between arbitrary query-document pairs. However, ours can obtain query-document pairs having soft labels, rather than strict positive or negative classes, by interpolating the positive and negative documents.

Stochastic Perturbation with Dropout In addition to our interpolation scheme to handle unlabeled documents in the space of interpolation of two labeled documents, we further aim at perturbing the labeled document to handle its nearby unlabeled documents as shown in Figure 2 right. In order to do so, we randomly mask the representation of the labeled document, obtained by the document encoder E_D, with dropout, where we sample masks from a Bernoulli distribution. In other words, if we sample n different masks from the distribution, we obtain n different query-document pairs $\left\{(\boldsymbol{q}, \boldsymbol{d}_i^+)\right\}_{i=1}^{i=n}$ from one positive pair $(\boldsymbol{q}, \boldsymbol{d}^+)$. By doing so, we augment n times more positive query-document pairs by replacing a single positive pair (q, d^+) in equation 2. Moreover, since the document perturbation is orthogonal to the interpolation, we further interpolate between the perturbed positive document $\boldsymbol{d}_i^+$ and the negative document $\boldsymbol{d}^-$ for the given query in equation 3, to augment a soft query-document pair from perturbation.

Efficiency Data augmentation methods are generally vulnerable to efficiency, since they need a vast amount of resources to generate data and to forward the generated data into the large language model. However, since our interpolation and perturbation methods only manipulate the already obtained representations of the documents from the encoder E_D, we don't have to newly generate document texts and also to forward generated documents into the model, which greatly saves time (See Table 3). We provide a detailed analysis and discussion of efficiency across models in **Appendix** B.2.

	MRR	MAP	T-20	T-5
DAR (Ours)	**42.92**	**27.12**	**75.04**	**57.62**
w/o Perturbation	41.26	26.19	73.68	55.37
w/o Interpolation	40.40	25.70	73.41	55.29
DPR	39.55	25.61	72.94	54.02

Table 4: Ablation studies of our DAR on the Natural Questions (NQ) dataset.

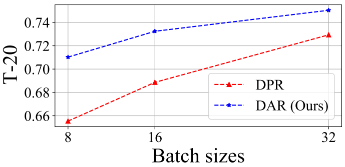

Figure 4: T-20 on the NQ dataset with varying batch sizes.

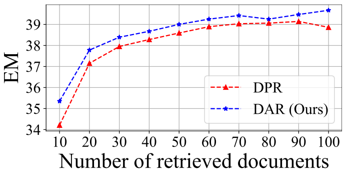

Figure 5: Exact Match (EM) scores for a reader on the NQ.

4 Experiments

4.1 Experimental Setups

Here, we describe datasets, models, and implementation details for experiments. For more experimental details, please see **Appendix** A.

Datasets For documents to retrieve, we use the Wikipedia, following Karpukhin et al. (2020), where the processed dataset contains 21,015,324 passages. To evaluate retrieval models, we use two open-domain QA datasets, following Karpukhin et al. (2020): 1) **Natural Questions (NQ)** is collected with Google search queries (Kwiatkowski et al., 2019); 2) **TriviaQA (TQA)** is a QA collection scraped from the Web (Joshi et al., 2017).

Retrieval Models **1) BM25** is a sparse term-based retrieval model based on TF-IDF (Robertson et al., 1994). **2) Dense Passage Retriever (DPR)** is a dense retrieval model with a dual-encoder of query-document pairs (Karpukhin et al., 2020). **3) DPR with Query Augmentation (DPR w/ QA)** augments pairs with query generation for the document, adopting (Liang et al., 2020; Mao et al., 2021a). **4) DPR with Document Augmentation (DPR w/ DA)** augments pairs by replacing words in the document (Ma, 2019). **5) DPR with Axiomatic Regularization (DPR w/ AR)** regularizes the retrieval model to satisfy certain axioms (Rosset et al., 2019). **6) DAR** is ours with interpolation and perturbation of document representations.

Implementation Details For all models, we set the training epoch as 25 and batch size as 32. We use in-batch negative sampling as our negative sampling strategy without hard negative samples. Also, we retrieve 100 passages per question.

4.2 Results

In this subsection, we show the overall performance of our DAR, and then give detailed analyses.

Overall Results As Table 1 shows, DAR outperforms dense retrieval baselines on all datasets on the DPR framework. Note that DAR contributes to more accurate retrieval performance, since the smaller K gives higher performance improvements. Furthermore, Figure 3 shows that, with our method, the retrieval performance on unlabeled documents – not seen during training – together with the labeled ones is improved, where performance gains on unlabeled are remarkable. To see the robustness of DAR on other retrievers, we further evaluate our model on the recent ANCE framework (See Appendix A for setups). As Table 2 shows, we observe that the performance improvement is more dominant on MRR when given a smaller number of training queries (low-resource settings), thus DAR effectively augments document representations.

Effectiveness of Interpolation & Perturbation Table 4 shows that each of the interpolation and stochastic perturbation positively contributes to the performance. In particular, when both of them are simultaneously applied, the performance is much improved, which demonstrates that these two techniques are in a complementary relationship.

Batch Size We test DAR with varying numbers of batch sizes. Figure 4 indicates that our DAR consistently improves the retrieval performance. Note that the smaller the batch size, the bigger the performance gap. Also, the batch size 16 of DAR outperforms the batch size 32 of the baseline, which highlights that DAR effectively augments document representations within a small batch.

Reader Performance To see whether accurately retrieved documents lead to better QA performance, we experiment with the same extractive reader from DPR without additional re-training. Figure 5 illustrates the effectiveness of our method on passage reading. Our reader performance can be further enhanced with advanced reading schemes (Mao et al., 2021a; Qu et al., 2021; Mao et al., 2021b).

5 Conclusion

We presented a novel method of augmenting document representations focusing on dense retrieval models, which require an extensive amount of labeled query-document pairs for training. Specifically, we augment documents by interpolating and perturbing their embeddings with mixup and dropout masks. The experimental results and analyses on two benchmark datasets demonstrate that DAR remarkably improves retrieval performances.

References

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1870–1879. Association for Computational Linguistics.

Jiaao Chen, Zichao Yang, and Diyi Yang. 2020a. MixText: Linguistically-informed interpolation of hidden space for semi-supervised text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2147–2157, Online. Association for Computational Linguistics.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020b. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR.

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.

Michael A. Hedderich, Lukas Lange, Heike Adel, Jannik Strötgen, and Dietrich Klakow. 2021. A survey on recent approaches for natural language processing in low-resource scenarios. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2545–2568, Online. Association for Computational Linguistics.

Soyeong Jeong, Jinheon Baek, ChaeHun Park, and Jong Park. 2021. Unsupervised document expansion for information retrieval with stochastic text generation. In *Proceedings of the Second Workshop on Scholarly Document Processing*, pages 7–17, Online. Association for Computational Linguistics.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *ACL*.

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.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

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

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096, Florence, Italy. Association for Computational Linguistics.

Seanie Lee, Minki Kang, Juho Lee, and Sung Ju Hwang. 2021. Learning to perturb word embeddings for out-of-distribution qa. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Davis Liang, Peng Xu, Siamak Shakeri, Cicero Nogueira dos Santos, Ramesh Nallapati, Zhiheng Huang, and Bing Xiang. 2020. Embedding-based zero-shot retrieval through query generation. *arXiv preprint arXiv:2009.10270*.

Edward Ma. 2019. Nlp augmentation. https://github.com/makcedward/nlpaug.

Ji Ma, Ivan Korotkov, Yinfei Yang, Keith Hall, and Ryan McDonald. 2021. Zero-shot neural passage retrieval via domain-targeted synthetic question generation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1075–1088, Online. Association for Computational Linguistics.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*.

Yuning Mao, Pengcheng He, Xiaodong Liu, Yelong Shen, Jianfeng Gao, Jiawei Han, and Weizhu Chen. 2021a. Generation-augmented retrieval for open-domain question answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4089–4100, Online. Association for Computational Linguistics.

Yuning Mao, Pengcheng He, Xiaodong Liu, Yelong Shen, Jianfeng Gao, Jiawei Han, and Weizhu Chen. 2021b. Reader-guided passage reranking for open-domain question answering. In *Findings of the*

Association for Computational Linguistics: ACL-IJCNLP 2021, pages 344–350, Online. Association for Computational Linguistics.

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.

Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. 1994. Okapi at TREC-3. In *Proceedings of The Third Text REtrieval Conference, TREC 1994, Gaithersburg, Maryland, USA, November 2-4, 1994*, volume 500-225 of *NIST Special Publication*, pages 109–126. National Institute of Standards and Technology (NIST).

Stephen E. Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: BM25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Corby Rosset, Bhaskar Mitra, Chenyan Xiong, Nick Craswell, Xia Song, and Saurabh Tiwary. 2019. An axiomatic approach to regularizing neural ranking models. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*, pages 981–984. ACM.

Connor Shorten and Taghi M. Khoshgoftaar. 2019. A survey on image data augmentation for deep learning. *J. Big Data*, 6:60.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958.

Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. 2019. Manifold mixup: Better representations by interpolating hidden states. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 6438–6447. PMLR.

Ellen M. Voorhees. 1999. The TREC-8 question answering track report. In *Proceedings of The Eighth Text REtrieval Conference, TREC 1999, Gaithersburg, Maryland, USA, November 17-19, 1999*, volume 500-246 of *NIST Special Publication*. National Institute of Standards and Technology (NIST).

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. 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 6383–6389, Hong Kong, China. Association for Computational Linguistics.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *International Conference on Learning Representations*.

Wenpeng Yin, Huan Wang, Jin Qu, and Caiming Xiong. 2021. BatchMixup: Improving training by interpolating hidden states of the entire mini-batch. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4908–4912, Online. Association for Computational Linguistics.

Yang You, Jing Li, Sashank J. Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. 2020. Large batch optimization for deep learning: Training BERT in 76 minutes. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. 2018. mixup: Beyond empirical risk minimization. *International Conference on Learning Representations*.

	Train	Val	Test
Natural Question (NQ)	58,880	6,515	3,610
TriviaQA (TQA)	60,413	6,760	11,313
MS MARCO, # Query: 10K	6,591	6,980	-
MS MARCO, # Query: 50K	32,927	6,980	-

Table 5: Statistics for training, validation, and test sets on the NQ, TQA, and randomly sampled MS MARCO datasets. Note that, for MS MARCO, we only sample the number of training query-document pairs except for the validation set.

A Experimental Setups

Datasets To evaluate the performance of retrieval models, we need two types of datasets: 1) a set of documents to retrieve, and 2) pairs of a query and a relevant document, having an answer for the query. We first explain the dataset that we used for the DPR framework (Karpukhin et al., 2020), and then describe the dataset for the ANCE framework (Xiong et al., 2021).

For documents to retrieve, we use the Wikipedia snapshot from December 20, 2018, which contains 21,015,324 passages consisting of 100 tokens, and follow the dataset processing procedure of Karpukhin et al. (2020) for the DPR framework. For open-domain QA datasets, we use Natural Question (NQ) (Kwiatkowski et al., 2019) and Trivia QA (TQA) (Joshi et al., 2017), following the dataset processing procedure of Karpukhin et al. (2020). We report the statistics of the training, validation, and test sets on NQ and TQA in Table 5.

To see the performance gain of our DAR on other dense retrieval models, we evaluate DAR on the ANCE framework (Xiong et al., 2021), which is one of the recent dense retrieval models. ANCE is evaluated on the MS MARCO dataset, thus we use MS MARCO for training and testing our model. Note that training ANCE with the full MS MARCO dataset requires 225 GPU hours even after excluding the excessive BM25 pre-training and inference steps. Thus we randomly sample the MS MARCO dataset to train the model under the academic budgets. Specifically, the subset of our MS MARCO passage dataset contains 500,000 passages. Also, we randomly divide the training queries into two subsets: one for 10,000 training queries and the other for 50,000 training queries. Then we align the sampled training queries to the query-document pairs in the MS MARCO dataset. On the other hand, we do not modify the validation set (dev set) of query-document pairs for testing. We summarize the statistics of the dataset in Table 5. Note that since the test set of MS MARCO is not publicly open, we evaluate the dense retrievers with the validation set, following Xiong et al. (2021).

Metrics Here, we explain the evaluation metrics for retrievers in detail. Specifically, given an input query, we measure the ranks of the correctly retrieved documents for the DPR framework with the following metrics:

1) Top-K Accuracy (T-K): It measures whether an answer of the given query is included in the retrieved top-k documents.

2) Mean Reciprocal Rank (MRR): It computes the rank of the first correct document for the given query among the top-100 retrieved documents, and then computes the average of the reciprocal ranks for all queries.

3) Mean Average Precision (MAP): It computes the mean of the average precision scores for all queries, where precision scores are calculated by the ranks of the correctly retrieved documents among top-100 ranked documents.

Next, we explain the evaluation metric for the reader, which identifies the answer from retrieved documents.

1) Exact Match (EM): It measures whether the reader exactly predicts one of the reference answers for each question.

Note that, for the ANCE framework, we follow the evaluation metrics, namely MRR@10 and Recall@1k, in the original paper (Xiong et al., 2021).

Experimental Implementation Details For dense retrieval models based on the DPR framework, we follow the dual-encoder structure of query and document by using the publicly available code from DPR[1] (Karpukhin et al., 2020). For all experiments, we set the batch size as 32, and train models on a single GeForce RTX 3090 GPU having 24GB memory. Note that, in contrast to the best reported setting of DPR which requires industrial-level resources of 8 V100 GPUs (8 × 32GB = 256GB) for training with a batch size of 128, we use a batch size of 32 to train the model under academic budgets. We optimize the model parameters of all dense retrieval models with the Adam optimizer (Kingma and Ba, 2015) having a learning rate of 2e-05. We train the models for 25 epochs, following the analysis[2] that the training phases converge after 25 epochs.

For the retrievers based on the ANCE framework, we refer to the implementation from

[1] https://github.com/facebookresearch/DPR

[2] See footnote 1.

ANCE[3] (Xiong et al., 2021). In order to directly measure the performance gain of the dense retrieval models based on ANCE from using our DAR, we use the pre-trained RoBERTa without warming up with the BM25 negatives. We train all the dense retrieval models for 50,000 steps with a single GeForce RTX 3090 GPU having 24GB memory, and simultaneously generate the ANN index with another GeForce RTX 3090 GPU, following Xiong et al. (2021). Following the standard implementation setting, we set the training batch size as 8, and optimize the model with the LAMB optimizer (You et al., 2020) with a learning rate of 1e-6.

Architectural Implementation Details For our augmentation methods, we use both interpolation and perturbation schemes of document representations obtained from the document encoder E_D in equation 1. Specifically, given a positive query-document pair $(\boldsymbol{q}, \boldsymbol{d}^+)$, we first perturb the document representation $\boldsymbol{d}^+$ with dropout masks sampled from a Bernoulli distribution, which generates n number of perturbed document representations $\{\boldsymbol{d}_i^+\}_{i=1}^{i=n}$. Then, we augment them to generate n number of positive query-document pairs $\{(\boldsymbol{q}, \boldsymbol{d}_i^+)\}_{i=1}^{i=n}$, which we use in equation 2. We search the number of perturbations n in the range from 3 to 9, and set the probability of the Bernoulli distribution as 0.1.

Instead of only using positive or negative pairs, we further augment query-document pairs having intermediate similarities with mixup. Specifically, we interpolate the representations between the perturbed-positive document $\boldsymbol{d}_i^+$ and the negative document $\boldsymbol{d}^-$ for the given query $\boldsymbol{q}$, with $\lambda \in [0, 1]$ in equation 3 sampled from a uniform distribution. Note that, given a positive pair of a query and a document, we consider the other documents except for the positive document in the batch as the negative documents. In other words, if we set the batch size as 32, then we could generate 31 interpolated document representations from 1 positive pair and 31 negative pairs. To jointly train the interpolation scheme with the original objective, we add the loss obtained from interpolation to the loss in equation 2.

	MRR	MAP	T-100	T-1
DPR+HN	53.40	33.38	84.82	43.21
DAR+HN (Ours)	**54.18**	**33.71**	**85.35**	**44.18**

Table 6: Retrieval results with hard negatives (HN) from BM25 on the NQ dataset for the DPR framework.

B Additional Experimental Results

B.1 Negative Sampling

One of the recent progresses in dense retrieval methods is to effectively sample negative query-document pairs over the whole combinations of queries and documents in a corpus. The most simple yet efficient negative sampling method is to sample negative pairs within the batch, where the documents in the batch except for the positive document for a given query are considered as the negative samples for the query. We use this in-batch negative sampling scheme (Chen et al., 2020b; Karpukhin et al., 2020) in equation 2. However, we could further use the sophisticated negative sampling schemes with the expense of computing resources. To mention a few, Karpukhin et al. (2020) proposed to use BM25 (Robertson and Zaragoza, 2009) to select the hard negative samples that have the highest similarity score for the positive document but cannot answer the given query. Also, Xiong et al. (2021) proposed to use learnable dense retrieval models to construct global negative samples from the entire corpus.

In our augmentation methods, we do not modify the negative log-likelihood loss for negative samples in equation 2. Therefore, our method could sample the negative pairs from any particular sampling schemes: $(q, d^-) \in \tau^-$. Thus, the negative sampling approaches are orthogonal to our DAR, from which the performance of DAR could be further improved by advanced sampling techniques. To see the effectiveness of our DAR coupled with a sophisticated negative sampling scheme, we compare DAR with the hard negative sampling strategy from BM25 (Karpukhin et al., 2020) against the baseline DPR with the same sampling strategy in Table 6. The results in Table 6 show that DAR with hard negative sampling outperforms the baseline method. The results demonstrate that the performance of dense retrieval models could be further strengthened with a combination of our augmentation methods and advanced negative sampling techniques. Also, in all our experiments of the ANCE framework, we already use the strategy of negative sampling in Xiong et al. (2021), where we observe the clear performance improvement of our

[3] https://github.com/microsoft/ANCE

	Memory (MiB)	Time (Min.)	Relative Time
DPR	22,071	19	1.00
DPR w/ QA	22,071	41	2.16
DPR w/ DA	22,071	38	2.00
DPR w/ AR	38,986	29	1.53
DAR (Ours)	22,071	21	1.11

Table 7: Maximum memory usage and time for training a DPR model per epoch.

DAR based on ANCE in Table 2.

B.2 Efficiency

As described in the Efficiency paragraph of Section 3, compared to the existing query augmentation methods (Liang et al., 2020; Ma et al., 2021; Qu et al., 2021), document augmentation method (Ma, 2019), and word replacement method for regularization (Rosset et al., 2019), our method of augmenting document representations with interpolation and perturbation in a dense representation space is highly efficient. This is because, unlike the baselines above, we do not explicitly generate or replace a query or document text; but rather we only manipulate the representations of documents. This scheme greatly saves the time for training, since additional forwarding of the generated or replaced query-document pairs into the language model is not required for our data augmentation methods.

To empirically validate the efficiency of our methods against the baselines, we report the memory usage and time for training a retrieval model per epoch in Table 7. In case of memory efficiency, all the compared dense retrieval models using data augmentation methods, including ours, use the same amount of maximum GPU memory. This shows that the overhead of memory usage comes from operations in the large-size language model, such as BERT (Devlin et al., 2019), not from manipulating the obtained document representations to augment the query-document pairs. Technically speaking, there are no additional parameters to augment document representations; thus our interpolation and perturbation methods do not increase the memory usage. On the other hand, DPR w/ AR excessively increases the memory usage, since it requires an extra forwarding process to the language model to represent the additional word-replaced sentences for regularization, instead of using the already obtained dense representations like ours.

We also report the training time for dense retrieval models in Table 7. Note that, for the explicit augmentation method based models, such as DPR w/ QA and DPR w/ DA, we exclude the extra time for training a generation model and generating a query or document for the given text. Also, we additionally generate the same number of query-document pairs in the training set, where the total amount of training data-points for DPR w/ QA and DPR w/ DA baselines are twice larger than the original dataset. Unlike these explicit query or document generation baselines, we perturb the document n times, but also interpolate the representations of positive and negative documents. As shown in Table 7, our DAR is about doubly more efficient than the explicit text augmentation methods, since DPR w/ QA and DPR w/ DA explicitly augment query-document pairs instead of using the obtained dense representations like ours. Also, our DAR takes a little more time to augment document representations than the base DPR model, while significantly improving retrieval performances as shown in Table 1. Even compared to the term replacement based regularization model (DPR w/ AR), our DAR shows noticeable efficiency, since an additional embedding process of the document after the word replacement on it requires another forwarding step besides the original forwarding step.

	T-5	T-20	T-100
DPR (Karpukhin et al., 2020)	52.1	70.8	82.1
DPR (Ours)	**53.2**	**71.6**	**82.7**

Table 8: Comparison of the DPR models' top-k accuracy between the reported and reproduced scores. Best performance is highlighted in bold.

B.3 Reproduction of DPR

We strictly set the batch size as 32 for training all the dense retrievers using the DPR framework; therefore the retrieval performances are different from the originally reported ones in Karpukhin et al. (2020) that use a batch size of 128. However, while we use the available code from the DPR paper, one may wonder if our reproduction result is accurate. Therefore, since Karpukhin et al. (2020) provided the retrieval performances of the DPR with different batch sizes (e.g., a batch size of 32), evaluated on the development (validation) set of the NQ dataset, we compare the Top-K accuracy between the reported scores and our reproduced scores. Table 8 shows that our reproduced Top-K accuracy scores with three different Ks (e.g., Top-5, Top-20, and Top-100) are indeed similar to the reported ones, with ours even higher, thus showing that our reproductions are accurate.