Dimension Reduction for Efficient Dense Retrieval via Conditional Autoencoder

Zhenghao Liu[1], Han Zhang[1], Chenyan Xiong[2], Zhiyuan Liu[3], Yu Gu[1], and **Xiaohua Li[1]**
[1]Department of Computer Science and Technology, Northeastern University, China
[2]Microsoft Research, United States
[3]Department of Computer Science and Technology, Institute for AI, Tsinghua University, China
Beijing National Research Center for Information Science and Technology, China
{liuzhenghao,guyu,lixiaohua}@mail.neu.edu.cn
zhanghan@stumail.neu.edu.cn; chenyan.xiong@microsoft.com
liuzy@tsinghua.edu.cn

Abstract

Dense retrievers encode queries and documents and map them in an embedding space using pre-trained language models. These embeddings need to be high-dimensional to fit training signals and guarantee the retrieval effectiveness of dense retrievers. However, these high-dimensional embeddings lead to larger index storage and higher retrieval latency. To reduce the embedding dimensions of dense retrieval, this paper proposes a Conditional Autoencoder (ConAE) to compress the high-dimensional embeddings to maintain the same embedding distribution and better recover the ranking features. Our experiments show that ConAE is effective in compressing embeddings by achieving comparable ranking performance with its teacher model and making the retrieval system more efficient. Our further analyses show that ConAE can alleviate the redundancy of the embeddings of dense retrieval with only one linear layer. All codes of this work are available at https://github.com/NEUIR/ConAE.

1 Introduction

As the first stage of numerous multi-stage IR and NLP tasks (Nogueira et al., 2019; Chen et al., 2017; Thorne et al., 2018), dense retrievers (Xiong et al., 2021a) have shown lots of advances in conducting semantic searching and avoiding the vocabulary mismatch problem (Robertson and Zaragoza, 2009). Dense retrievers usually encode queries and documents as high-dimensional embeddings, which are necessary to guarantee retrieval effectiveness during training (Ma et al., 2021; Reimers and Gurevych, 2021). Nevertheless, high dimensional embeddings usually exhaust the memory to store the index and lead to longer retrieval latency (Indyk and Motwani, 1998; Meiser, 1993).

The research of building efficient dense retrieval systems has been stimulated recently (Min et al., 2021). To reduce the dimensions of document embeddings, existing work reserves the principle dimensions or compresses query and document embeddings for building more efficient retrievers (Yang and Seo, 2021; Ma et al., 2021).

There are two challenges in compressing embeddings of dense retrievers: The compressed embeddings should share a similar distribution with the original embeddings, making the low-dimensional embedding space uniform and the document embeddings distinguishable; All the compressed embeddings should have the ability to maintain the maximal information for matching related queries and documents during retrieval, which helps better align the related query-document pairs.

This paper proposes a Conditional Autoencoder (ConAE), which aims to build efficient dense retrieval systems by reducing the embedding dimensions of queries and documents. ConAE first encodes high-dimensional embeddings into a low-dimensional embedding space and then generates embeddings that can be aligned to related queries or documents in the original embedding space. In addition, ConAE designs a conditional loss to regulate the low-dimensional embedding space to mimic the embedding distribution of high-dimensional embeddings. Our experiments show that ConAE is effective to compress the high-dimensional embeddings and avoid redundant ranking features by achieving comparable retrieval performance with vanilla dense retrievers and better visualizing the embedding space with t-SNE.

2 Related Work

Dense retrievers use a bi-encoder architecture to encode queries and documents and map them in an embedding space for retrieval (Karpukhin et al., 2020; Xiong et al., 2021b,a; Lewis et al., 2020; Zhan et al., 2021; Li et al., 2021; Yu et al., 2021).

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 5692 - 5698
December 7-11, 2022

To learn an effective embedding space, dense retrievers are forced to maintain high-dimensional embeddings to fit training signals.

The most direct way to reduce the dimension of embeddings is that retaining parts of the dimensions of high-dimensional embeddings (Yang and Seo, 2021; Ma et al., 2021). Some work uses the first 128 dimensions to encode both queries and documents (Yang and Seo, 2021) or utilizes PCA to retain the primary dimensions to recover most information from the raw embeddings (Ma et al., 2021). Other work (Ma et al., 2021) proposes a supervised method, which uses neural networks to compress the high-dimensional embeddings as lower-dimensional ones. These supervised models provide a better dimension reduction way than unsupervised models by avoiding missing too much information. To optimize the encoders, some work (Ma et al., 2021) continuously trains dense retrievers with the contrastive training strategies (Karpukhin et al., 2020; Xiong et al., 2021a).

3 Methodology

This section introduces our Conditional Autoencoder (ConAE). We first introduce the preliminaries of dense retrieval (Sec. 3.1), and then describe the architecture of ConAE (Sec. 3.2).

3.1 Preliminary of Dense Retrieval

Given a query q and a document collection $D = \{d_1, \ldots, d_j, \ldots, d_n\}$, dense retrievers (Xiong et al., 2021b,a; Karpukhin et al., 2020) employ pre-trained language models (Devlin et al., 2019; Liu et al., 2019) to encode q and d as K-dimensional embeddings, h_q and h_d.

Then we can calculate the retrieval score $f(q, d)$ of q and d with dot product $f(h_q, h_d) = h_q \cdot h_d$. Then we contrastively train query and document encoders by maximizing the retrieval probability $P(d^+|q, \{d^+\} \cup D^-)$ of the relevant document d^+ (Xiong et al., 2021b,a):

$$P(d^+|q, \{d^+\} \cup D^-) = \frac{e^{f(h_q, h_{d^+})}}{e^{f(h_q, h_{d^+})} + \sum\limits_{d^- \in D^-} e^{f(h_q, h_{d^-})}}, \tag{1}$$

where d^- is the document sampled from the irrelevant document set D^- (Karpukhin et al., 2020; Xiong et al., 2021a).

3.2 Dimension Compression with ConAE

In this subsection, we introduce ConAE to compress the K-dimensional embeddings h_q and h_d of both queries and documents to the L-dimensional embeddings h_q^e and h_d^e.

Encoder. We first get the initial representations h_q and h_d for query q and document d from existing dense retrievers, such as ANCE (Xiong et al., 2021a). Then these K-dimensional embeddings can be compressed to low dimensional ones with two different linear layers, Linear_q and Linear_d:

$$h_q^e = \text{Linear}_q(h_q); h_d^e = \text{Linear}_d(h_d). \tag{2}$$

h_q^e and h_d^e are L-dimensional embeddings. The dimension L can be 256, 128 or 64, which is much lower than the dimension K of h_q and h_d.

Then we use KL divergence to regulate encoded embeddings to mimic the initial embedding distributions of queries and documents:

$$L_{KL} = \sum_q \sum_{d \in D_{\text{top}}} P(d|q, D_{\text{top}}) \cdot \log \frac{P(d|q, D_{\text{top}})}{P_e(d|q, D_{\text{top}})}, \tag{3}$$

where $P_e(d|q, D_{\text{top}})$ is calculated with E.q. 1, using the encoded embeddings h_q^e and h_d^e. D_{top} consists of the top-ranked documents, which are searched by the teacher retriever–ANCE.

Decoder. The decoder module maps the encoded embeddings h_q^e and h_d^e into the original embedding space by aligning the compressed embeddings h_q^e and h_d^e with h_q and h_d. It aims at optimizing encoder modules to maximally maintain ranking features from the initial representations h_q and h_d of query and document.

Firstly, we use one linear layer to project h_q^e and h_d^e to K-dimensional embeddings, $\hat{h}_q$ and $\hat{h}_d$:

$$\hat{h}_q = \text{Linear}(h_q^e); \hat{h}_d = \text{Linear}(h_d^e). \tag{4}$$

Then we respectively train the decoded embeddings $\hat{h}_q$ and $\hat{h}_d$ to align with h_q and h_d in the original embedding space using two max margin losses L_q and L_d. The max margin loss is widely used in previous neural IR research to optimize the ranking scores (Xiong et al., 2017; Dai et al., 2018).

The first loss L_q is used to optimize the decoded query representation $\hat{h}_q$:

$$L_q = \sum_q 1 + \tanh f(\hat{h}_q, h_{d^-}) - \tanh f(\hat{h}_q, h_{d^+}), \tag{5}$$

and we can also optimize the decoded document representation $\hat{h}_d$ with the second loss function L_d:

$$L_d = \sum_q 1 + \tanh f(h_q, \hat{h}_{d^-}) - \tanh f(h_q, \hat{h}_{d^+}). \tag{6}$$

Dataset	#Doc	#Queries		
		Train	Dev	Test
MS MARCO	8,841,823	452,939	50,000	6,980
NQ	21,015,323	79,168	8,757	3,610
TREC DL	8,841,823	-	-	43
TREC-COVID	171,332	-	-	50

Table 1: Data Statistics.

Training Loss. Finally, we train our conditional autoencoder model with the following loss L:

$$L = L_{KL} + \lambda L_q + \lambda L_d, \tag{7}$$

where λ is a hyper-parameter to weight the autoencoder losses.

4 Experimental Methodology

This section describes the datasets, evaluation metrics, baselines and implementation details of our experiments.

Dataset. Four datasets are used to evaluate the retrieval effectiveness of different dimension reduction models, including MS MARCO (Passage Ranking) (Nguyen et al., 2016), NQ (Kwiatkowski et al., 2019), TREC DL (Craswell et al., 2020) and TREC-COVID (Roberts et al., 2020). In our experiments, we randomly sample 50,000 queries from the raw training set of MS MARCO as the development set and use MS MARCO (Dev) as the testing set. The dimension reduction models that are trained on MS MARCO are also evaluated on two benchmarks, TREC DL and TREC-COVID, aiming to evaluate their generalization ability. All data statistics are shown in Table 1.

Evaluation Metrics. NDCG@10 is used as the evaluation metric on three benchmarks, MS MARCO, TREC DL and TREC-COVID. MS MARCO also uses MRR@10 as the primary evaluation metric (Nguyen et al., 2016). For the NQ dataset, the hit accuracy on Top20 and Top100 is used as the evaluation metric, which is the same as previous work (Karpukhin et al., 2020).

Baselines. In our experiments, we compare ConAE with two baselines from previous work (Ma et al., 2021), Principle Component Analysis (PCA) and CE. PCA reduces the embedding dimension by retaining the principle dimensions that can keep most of the variance within the original representation. CE model uses two linear layers W_q and W_d without biases to transform dense representations of queries and documents into lower embeddings (Ma et al., 2021). We also start from CE models and continuously train the whole model to implement our ANCE models to generate query and document embeddings of different dimensions.

Implementation Details. The rest describes our implementation details. All embedding dimension reduction models base on one of the best dense retrievers ANCE (Xiong et al., 2021a) and build document index with exact matching (flat index), which is implemented by FAISS (Johnson et al., 2019). During training ConAE, we set the hyper-parameter λ as 0.1 and search Top100 documents using vanilla ANCE to construct the D_{top} collection for each query. For our CE and ANCE models, we sample 7 negative documents for each query to contrastively train these models and sample 1 negative document to train ConAE. In our experiments, we set the batch size to 2 and accumulate step to 8 for ANCE. The batch size and accumulate step are 128 and 1 for other models. All models are implemented with PyTorch and tuned with Adam optimizer. The learning rates of ANCE and other models are set to $2e-6$ and 0.001, respectively.

5 Evaluation Result

Four experiments are conducted in this section to study the effectiveness of ConAE in reducing embedding dimensions for dense retrieval.

5.1 Overall Performance

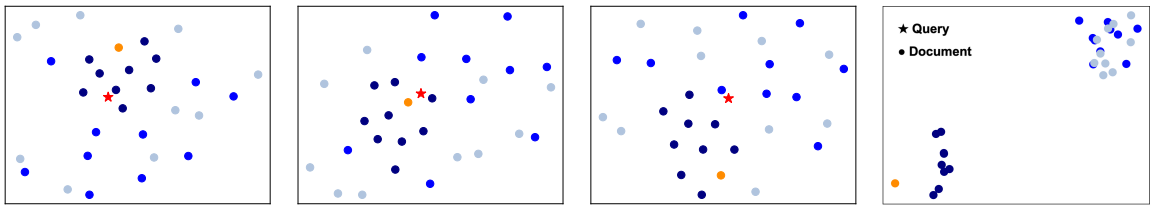
The performance of different dimension reduction models is shown in Table 2. PCA, CE and ConAE are based on ANCE (Teacher), which freezes the teacher model and only optimizes the dimension projection layers. ANCE starts from CE and continuously tunes all parameters in the model.

Compared with PCA and CE (Ma et al., 2021), ConAE achieves the best performance on almost of datasets, which shows its effectiveness in compressing dense retrieval embeddings. ConAE can achieve comparable performance with ANCE (Teacher) using 128-dimensional embeddings to build the document index on MS MARCO, which reduces the retrieval latency (from 17.152 ms to 3.942 ms per query) and saves the index storage (from 26.0G to 4.3G) significantly. It demonstrates that ConAE is effective to alleviate the redundancy of the embeddings learned by dense retrievers.

Among all baselines, PCA shows significantly worse ranking performance on MS MARCO, indicating that embedding dimensions of dense retrievers are usually nonorthogonal. ConAE-128 achieves more than 11% improvements than

Method	MS MARCO				NQ		TREC DL	TREC-COVID
	Latency(ms)	MRR@10	NDCG@10	Rec@1000	Top20	Top100	NDCG@10	NDCG@10
Teacher-768	17.152	0.3302	0.3877	0.9584	0.8224	0.8787	0.6489	0.6529
ANCE-256	6.159	0.3145	0.3709	0.9545	**0.8188**	**0.8765**	**0.6455**	0.5722
PCA-256	6.296	0.2440	0.2940	0.9257	0.8042	0.8715	0.5118	0.2601
CE-256	7.344	0.2959	0.3472	0.9333	0.8066	0.8726	0.5916	0.4110
ConAE-256	7.158	**0.3294**	**0.3864**	**0.9560**	0.8053	0.8723	0.6438	**0.6405**
ANCE-128	3.419	0.3092	0.3667	**0.9527**	**0.8069**	**0.8709**	**0.6514**	0.5612
PCA-128	3.525	0.2348	0.2838	0.9170	0.7875	0.8620	0.4795	0.2523
CE-128	4.530	0.2917	0.3438	0.9345	0.7934	0.8668	0.6170	0.4692
ConAE-128	3.942	**0.3245**	**0.3816**	0.9523	0.8064	0.8687	0.6380	**0.6381**
ANCE-64	3.041	0.2773	0.3295	0.9217	**0.7687**	**0.8474**	**0.6003**	0.4731
PCA-64	2.627	0.1855	0.2259	0.8540	0.6698	0.7928	0.3788	0.2174
CE-64	3.046	0.2551	0.3036	0.9042	0.7404	0.8341	0.5561	0.3968
ConAE-64	3.087	**0.2862**	**0.3376**	**0.9222**	0.7604	0.8460	0.5877	**0.5006**

Table 2: Performance of Different Dimension Reduction Models. We start from ANCE (Teacher), reduce the embedding dimension and evaluate their retrieval effectiveness. The document indices are built with flat index and the sizes of MS MARCO indices are 26.0G, 8.5G, 4.3G and 2.2G for 768, 256, 128 and 64 dimensional embeddings.



(a) Teacher (t-SNE). (b) ConAE-128 (t-SNE). (c) ConAE-64 (t-SNE). (d) Teacher (ConAE).

Figure 1: Embedding Visualization of Different Dense Retrievers. Figure 1(a), 1(b) and 1(c) are plotted with t-SNE with 768, 128 and 64 dimensional embeddings. In Figure 1(d), we directly use ConAE w/o Decoder to visualize the document embedding space of ANCE. The "•" in "dark orange" color denotes the golden document that ranked 2nd by ConAE-64 and 1st by other models. For other documents, darker blue ones are more relevant to the query.

Method	MS MARCO MRR@10	TREC-COVID NDCG@10	TREC DL NDCG@10
ConAE-256	**0.3294**	0.6405	0.6438
w/o Decoder	0.3271	**0.6546**	0.6377
w/o KL	0.3276	0.6218	**0.6491**
ConAE-128	**0.3245**	0.6381	**0.6380**
w/o Decoder	0.3203	**0.6525**	0.6266
w/o KL	0.3234	0.6365	0.6367
ConAE-64	**0.2862**	**0.5006**	0.5877
w/o Decoder	0.2846	0.4703	**0.5951**
w/o KL	0.2822	0.4658	0.5759

Table 3: Retrieval Performance of Different Ablation Models. ConAE w/o Decoder and ConAE w/o KL use L_{KL} and $L_q + L_d$ to train the distillation models.

CE and performs much better on TREC-COVID, demonstrating its ranking effectiveness and generalization ability. ANCE can further improve the retrieval performance of CE by continuously training the query and document encoders, which adapts the teacher model to the low-dimensional version.

5.2 Ablation Study

This subsection conducts ablation studies in Table 3 to investigate the effectiveness of different modules in our ConAE model.

The different modules in ConAE play different roles. Compared with ConAE w/o Decoder, ConAE w/o KL usually shows better retrieval effectiveness on the two benchmarks MS MARCO and TREC DL, which ask the model to retrieve candidates from the same data source. It demonstrates that our autoencoder architecture can reserve more ranking features to fit the training supervision of MS MARCO. On the other hand, ConAE w/o Decoder shows stronger generalization ability by outperforming ConAE w/o KL on TREC-COVID, which belongs to a different domain. The source of the generalization ability of ConAE w/o Decoder may come from finer-grained training signals from our teacher model. The annotated training signals usually face the hole rate problem (Xiong et al., 2020) and using neural IR models to denoise the training signals has shown strong effectiveness in training neural IR models (Qu et al., 2021).

ConAE combines both KL and autoencoder architectures to fully use training signals and regulate the distribution of compressed embedding, which usually achieves better retrieval performance.

5.3 Embedding Visualization with ConAE

We randomly sample one case from MS MARCO and visualize the embedding space of query and retrieved documents in Figure 1.

We first employ t-SNE (van der Maaten and Hinton, 2008) to visualize the embedding spaces of ANCE (Teacher) and ConAE. As shown in Figure 1(b), ConAE-128 conducts a more meaningful visualization results: the related query-document pair is closer and the other documents are distributed around the golden document according to their relevance to the query. The visualization of ANCE (Teacher) is slightly distorted and different from our expectations, which is mainly due to its redundancy. The redundant features usually mislead t-SNE to overfit these ranking features, thus reducing the embedding dimension of dense retrievers to 128 provides a possible way to alleviate redundant features and better visualize the embedding space of dense retrievers using t-SNE. Besides, ConAE-64 shows decreased retrieval performance than ConAE-128 (Sec. 5.1). As shown in Figure 1(c), it mainly derives from that ConAE-64 loses some ranking features with the limited embedding dimensions.

The other way to visualize the embedding space is using ConAE w/o Decoder to project the embedding to a 2-dimensional coordinate. It uses KL divergence to optimize the 2-dimensional embeddings to mimic the relevance score distribution of teacher models. As shown in Figure 1(d), the distributions of documents are distinguishable, which provides an intuitive way to analyze the ranking-oriented document distribution. In addition, the query is usually far away from the documents. The main reason lies that the relevance scores are calculated by dot product and the embedding norms are meaningful to distinguish the relevant documents.

5.4 Retrieval Performance with HNSW

Besides exact searching, we also show retrieval results of different dimension reduction methods in Table 4, which are implemented by the approximate nearest neighbor (ANN) search, Hierarchical Navigable Small World (HNSW). Using HNSW, the retrieval efficiency can be further improved, especially for high-dimensional embeddings. ConAE keeps its advanced retrieval performance again with less than 1ms retrieval latency.

Dim.	Method	Latency (ms)	MS MARCO MRR@10	MS MARCO NDCG@10
768	ANCE	2.056	0.3295	0.3869
256	PCA	**1.016**	0.2427	0.2924
	CE	1.646	0.2948	0.3461
	ConAE	1.478	**0.3257**	**0.3818**
128	PCA	**0.702**	0.2340	0.2829
	CE	1.047	0.2906	0.3424
	ConAE	0.978	**0.3209**	**0.3770**
64	PCA	**0.616**	0.1844	0.2246
	CE	0.860	0.2545	0.3030
	ConAE	0.983	**0.2837**	**0.3344**

Table 4: ANN Retrieval Effectiveness of Different Models. The ANN index is built with HNSW.

6 Conclusion

This paper presents ConAE, which reduces the embedding dimension of dense retrievers. Our experiments show that ConAE can achieve comparable retrieval performance with the teacher model, significantly reduce the index storage and accelerate the searching process. Our further analyses show that the high-dimensional embeddings of dense retrievers are usually redundant and ConAE helps to alleviate such redundancy and visualize the embedding space more intuitively and effectively.

Limitations

In this paper, we mainly focus on compressing the embeddings of dense retrievers in an additional stage between query/document encoding and index building. As a result, we fix query and document embeddings of dense retrievers and project high-dimensional embeddings to low-dimensional ones using only one linear layer. Thus, the effectiveness of ConAE is limited by the number of learnable parameters. Even though ConAE shows comparable performance with ANCE (Teacher), joint modeling the query/document encoder, dimension reduction module and index building still show strong potential to achieve better retrieval performance.

Acknowledgments

This work is mainly supported by Beijing Academy of Artificial Intelligence (BAAI) as well as supported in part by the Natural Science Foundation of China under Grant No. 62206042 and No. 62006129, the Fundamental Research Funds for the Central Universities under Grant No. N2216013, China Postdoctoral Science Foundation under Grant No. 2022M710022 and National Science and Technology Major Project (J2019-IV-0002-0069).

References

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *Proceedings of ACL*, pages 1870–1879.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. 2020. Overview of the trec 2020 deep learning track.

Zhuyun Dai, Chenyan Xiong, Jamie Callan, and Zhiyuan Liu. 2018. Convolutional neural networks for soft-matching n-grams in ad-hoc search. In *Proceedings of WSDM*, pages 126–134.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Piotr Indyk and Rajeev Motwani. 1998. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 604–613.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of EMNLP*, pages 6769–6781.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Mike Lewis, Marjan Ghazvininejad, Gargi Ghosh, Armen Aghajanyan, Sida Wang, and Luke Zettlemoyer. 2020. Pre-training via paraphrasing. In *Proceedings of NeurIPS*.

Yizhi Li, Zhenghao Liu, Chenyan Xiong, and Zhiyuan Liu. 2021. More robust dense retrieval with contrastive dual learning. In *Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval*, pages 287–296.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach.

Xueguang Ma, Minghan Li, Kai Sun, Ji Xin, and Jimmy Lin. 2021. Simple and effective unsupervised redundancy elimination to compress dense vectors for passage retrieval. In *Proceedings of EMNLP*, pages 2854–2859.

Stefan Meiser. 1993. Point location in arrangements of hyperplanes. *Information and Computation*, 106(2):286–303.

Sewon Min, Jordan Boyd-Graber, Chris Alberti, Danqi Chen, Eunsol Choi, Michael Collins, Kelvin Guu, Hannaneh Hajishirzi, Kenton Lee, Jennimaria Palomaki, et al. 2021. Neurips 2020 efficientqa competition: Systems, analyses and lessons learned. In *Proceedings of the NeurIPS 2020 Competition and Demonstration Track*, pages 86–111.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. In *Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, December 9, 2016*, volume 1773.

Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. 2019. Document expansion by query prediction.

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 NAACL-HLT*, pages 5835–5847.

Nils Reimers and Iryna Gurevych. 2021. The curse of dense low-dimensional information retrieval for large index sizes. In *Proceedings of ACL*, pages 605–611.

Kirk Roberts, Tasmeer Alam, Steven Bedrick, Dina Demner-Fushman, Kyle Lo, Ian Soboroff, Ellen Voorhees, Lucy Lu Wang, and William R Hersh. 2020. Trec-covid: rationale and structure of an information retrieval shared task for covid-19. *Journal of the American Medical Informatics Association*, 27(9):1431–1436.

Stephen Robertson and Hugo Zaragoza. 2009. *The probabilistic relevance framework: BM25 and beyond*. Now Publishers Inc.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2018. The fact extraction and VERification (FEVER) shared task. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 1–9.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605.

Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power. 2017. End-to-end neural ad-hoc ranking with kernel pooling. In *Proceedings of SIGIR*, pages 55–64.

Chenyan Xiong, Zhenghao Liu, Si Sun, Zhuyun Dai, Kaitao Zhang, Shi Yu, Zhiyuan Liu, Hoifung Poon, Jianfeng Gao, and Paul Bennett. 2020. Cmt in trec-covid round 2: mitigating the generalization gaps from web to special domain search.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021a. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *Proceedings of ICLR*.

Wenhan Xiong, Xiang Lorraine Li, Srinivasan Iyer, Jingfei Du, Patrick S. H. Lewis, William Yang Wang, Yashar Mehdad, Wen-tau Yih, Sebastian Riedel, Douwe Kiela, and Barlas Oguz. 2021b. Answering complex open-domain questions with multi-hop dense retrieval. In *Proceedings of ICLR*.

Sohee Yang and Minjoon Seo. 2021. Designing a minimal retrieve-and-read system for open-domain question answering. In *Proceedings of NAACL-HLT*, pages 5856–5865.

Shi Yu, Zhenghao Liu, Chenyan Xiong, Tao Feng, and Zhiyuan Liu. 2021. Few-shot conversational dense retrieval. In *Proceedings of SIGIR*.

Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. 2021. Optimizing dense retrieval model training with hard negatives. In *Proceedings of SIGIR*.