QuadrupletBERT: An Efficient Model For Embedding-Based Large-Scale Retrieval

Peiyang Liu[1,2], **Sen Wang**[3], **Xi Wang**[2], **Wei Ye**[1,*] and **Shikun Zhang**[1]
[1] National Engineering Research Center for Software Engineering, Peking University, Beijing, China,
[2] School of Software and Microelectronics, Peking University, Beijing, China,
[3] PX Securities, Beijing, China,
wangsen_ai_px@yeah.net, {liupeiyang, wangxi5629, wye, zhangsk}@pku.edu.cn

Abstract

The embedding-based large-scale query-document retrieval problem is a hot topic in the information retrieval (IR) field. Considering that pre-trained language models like BERT have achieved great success in a wide variety of NLP tasks, we present a QuadrupletBERT model for effective and efficient retrieval in this paper. Unlike most existing BERT-style retrieval models, which only focus on the ranking phase in retrieval systems, our model makes considerable improvements to the retrieval phase and leverages the distances between simple negative and hard negative instances to obtaining better embeddings. Experimental results demonstrate that our QuadrupletBERT achieves state-of-the-art results in embedding-based large-scale retrieval tasks.

1 Introduction

Large-scale retrieval systems such as search engines have been a vital tool to help people access the massive amount of online information. Various techniques have been developed to improve retrieval quality in the last decades.

Due to the difficulty of computing search intent from the query text and accurately representing the semantic meaning of document requirements, most previous studies are based on classic term-weighting methods such as BM-25 (Robertson and Zaragoza, 2009) or TF-IDF (Spärck Jones, 1972, 2004) or simple context-free word embedding (Mikolov et al., 2013) that perform well for the cases that keyword matching can address. However, these models only accept sparse handcrafted features and cannot capture complex semantic features.

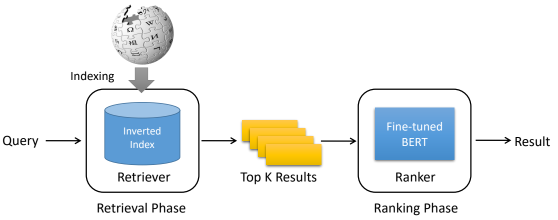


Figure 1: The architecture of large-scale retrieval systems.

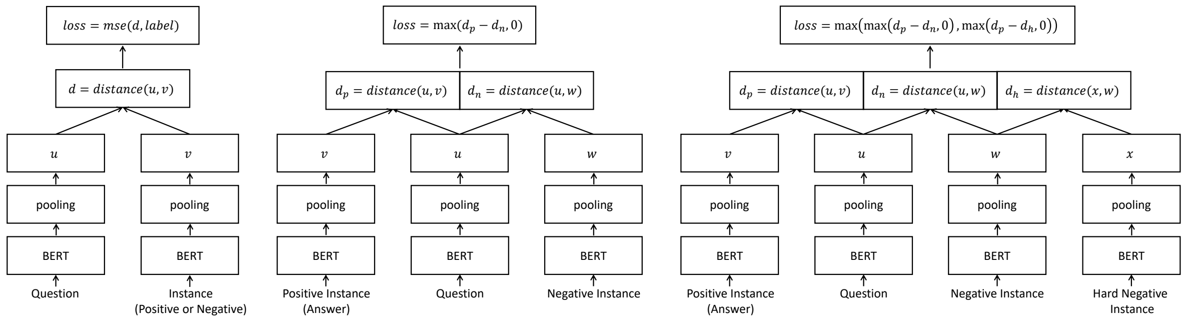
Considering that pre-trained language models like BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) have achieved great success in a wide variety of NLP tasks, more and more researchers propose BERT-style models to solve large-scale retrieval problems.

Some previous work has confirmed the effectiveness of BERT for enhancing retrieval systems. For example, Yilmaz et al. (2019) apply a BERT-style model to document retrieval via integration with the open-source anserini information retrieval toolkit to demonstrate end-to-end search over large document collections. Yang et al. (2019) build a BERT-based reader to identify answers from a large corpus of Wikipedia articles in an end-to-end fashion. Padaki et al. (2020) use query expansion to generate better queries for BERT-based *Ranker* in retrieval. Mass and Roitman (2020) describe a weakly-supervised method for training BERT-style models for *ad hoc* document retrieval.

In BERT, the prediction function $f(query, doc)$ is a pre-trained deep bidirectional Transformer model (Vaswani et al., 2017). While the above BERT-style models are very successful, this approach cannot be directly applied to large-scale retrieval problems because predicting f for every possible document can be prohibitively expensive. Thus, the methods mentioned above first use a less powerful but more efficient retrieval algorithm (*Retriever*) such as an inverted index to reduce the solution space and then use the BERT-style model to re-rank the retrieved documents. As shown in figure 1, we refer to all such BERT-style retrieval models as *Ranker*.

[*] Corresponding author

Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 3734–3739
June 6–11, 2021. ©2021 Association for Computational Linguistics



(a) Architecture of two-tower BERT model where $label = 0$ when input pairs are negative $label = 1$ when input pairs are positive.

(b) Architecture of three-tower BERT model.

(c) Architecture of our QuadrupletBERT.

Figure 2: Architecture of BERT-Style *Retriever* in large-scale retrieval.

Unlike these *Ranker* which have recently seen significant advances, constructing a BERT-style *Retriever* is a new topic in the large-scale retrieval field, on which few studies have thus far focused. For example, Reimers and Gurevych (2019) present a modification of the pre-trained BERT network that uses siamese and triplet network structures to derive semantically meaningful sentence embeddings that can be compared using cosine similarity. Chang et al. (2020) build a two-tower Transformer model with more pre-training data, which can significantly outperform the widely used BM-25 algorithm. Lu et al. (2020) distill knowledge from BERT into a two-tower architecture network for efficient retrieval.

As shown in figure 2 (a) and (b), the existing BERT-style *Retriever* mentioned above simply builds a two- or three-tower network structure to compute distances between positive and negative instances, which ignores the fact there are not only simple negative instances in the dataset: some instances are seemingly positive but actually negative, which we call hard negative instances. As we all know, the *Retriever* should have high recall; otherwise, many positive instances will not even be considered in the ranking phase. However, due to hard negative instances being literally related, treating them as equal to simple negative instances may harm the embedding of positive instances and lead the model to identify positive instances as negative ones mistakenly.

The key to solving the problem mentioned above is incorporating the distances between hard negative and simple negative instances into the training step. Our intuition is that hard negative instances are negative compared to positive instances but should be considered positive compared to simple negative instances. Therefore, we explore a new way to incorporate distances between hard negative and simple negative instances into the training process and build a four-tower BERT-style model named QuadrupletBERT. We have evaluated our model on two Retrieval Question-Answering (ReQA) benchmarks. Experimental results show that our model registers huge improvements over existing BERT-style *Retriever* models and achieves state-of-the-art results.

Our main contributions are as follows:

1. We propose a new four-tower BERT-style model named QuadrupletBERT, which is very easy to use and improves hugely over existing BERT-style *Retriever* models.

2. We find that leveraging distances between hard negative and simple negative instances in the training process helps improve the *Retriever* model.

2 Task Description

Large-scale retrieval problems can be defined as: given a query, return the most relevant documents from a large corpus, where the corpus' size can be hundreds of thousands or more. The embedding-based retrieval model jointly embeds queries and documents in the same embedding space and uses an inner product or cosine distance to measure the similarity between queries and documents. Since embeddings of all candidate documents can be precomputed and indexed, the inference can be made efficiently with approximate nearest neighbor search algorithms in the embedding space (Shrivastava and Li, 2014; Guo et al., 2016). Let the query embedding model be $\phi(\cdot)$, and the document embedding model be $\psi(\cdot)$ The distance function can be defined as:

$$f(query, doc) = \langle \phi(query), \psi(doc) \rangle \quad (1)$$

In this paper, we are interested in parameterizing the encoders ϕ and ψ as a four-tower BERT which incorporates the distances between hard negative and simple negative instances into the training step.

3 QuadrupletBERT

As shown in figure 2 (c), the core of our model is a four-tower sentence-level BERT relevance encoder. Each tower of our retrieval model follows the architecture and hyper-parameters of the 12 layers BERT model[1]. Note that for all BERT baselines, we all pre-train them on the specific downstream datasets by **Masked LM** and **Next Sentence Prediction** tasks (Devlin et al., 2019). The embedding dimension is 768. The sequence length for the encoder is set to be 64. For all towers, taking the average of the encoding layer's hidden state on the time axis as the final embedding.

3.1 Training

One unique advantage of the multi-tower retrieval model compared with classic IR algorithms is the ability to train it for specific tasks. In this paper, our training data x can be defined as quaternion query-document pairs:

$$\tau = \{(q_i, p_i, n_i, hn_i)\}_{i=1}^{|\tau|} \tag{2}$$

where q, p, n, and hn are representing query, positive document, negative document, hard negative document separately. We estimate the model parameters by minimizing the following loss function:

$$\begin{aligned}
loss &= \sum_{i=1}^{|\tau|} max(loss_i^h, loss_i^n) \\
loss_i^h &= max(d_i^p - d_i^h + m, 0) \\
loss_i^n &= max(d_i^p - d_i^n + m, 0) \\
d_i^p &= f(q_i, p_i) \\
d_i^n &= f(q_i, n_i) \\
d_i^h &= f(hn_i, n_i)
\end{aligned} \tag{3}$$

where d_i^p is the distance between q_i and p_i, d_i^n is the distance between q_i and n_i, and d_i^h is the distance between hn_i and n_i. This loss function constructed by two parts, where both $loss_i^h$ and $loss_i^n$ aim to minimize d_i^p. Besides, $loss_i^h$ aims to maximize d_i^h, and $loss_i^n$ aims to maximize d_i^n. m is the margin enforced between positive, negative, and hard negative documents. This loss function's intuition is to cluster the query and positive documents and separate the positive and hard negative documents from the negative documents by a distance margin. The distance function f we select is cosine distance, which can be defined as follows:

$$f(\mathbf{X}, \mathbf{Y}) = 1 - \frac{\mathbf{X} \cdot \mathbf{Y}}{||\mathbf{X}|| \times ||\mathbf{Y}||} \tag{4}$$

3.2 Inference

First, we pre-compute all the document embeddings. Then, given an unseen query q, we only need to rank the document based on its cosine distance with the query embedding. To make our QuadrupletBERT can be applied in resource-restricted and time-sensitive systems such as query understanding in search engines (Nakamura et al., 2019), we deployed an inverted index based ANN (approximate near neighbor) search algorithms to our model. We employed Faiss library (Johnson et al., 2017) to quantize the vectors and then implemented the efficient embedding search in our model.

4 Experiments and Results

4.1 Datasets and Baselines

We consider the Retrieval Question-Answering (ReQA) benchmark proposed by Ahmad et al. (2019). The two QA datasets we consider are SQuAD and Natural Questions. Note that each entry of QA datasets is a tuple (q, a, e), where q is the question, a is the answer span, and e is the evidence passage containing a. Following Ahmad et al. (2019), we split a passage into sentences $e = s_1 s_2 ... s_n$ and transform the original entry to a new tuple (q, s_i).

Different from the ranking phase of large-scale retrieval. The retrieval phase is that given a question q, retrieve the correct sentence s from all candidates. For each evidence passage e we create a set of candidate sentences s_i, and the retrieval candidate set is built by combining such sentences for all passages.

To construct our training quaternion pairs (q_i, p_i, n_i, hn_i). For a specific question q_i, we define the gold sentence containing a_i as p_i, and randomly select a sentence not containing a_i as n_i. We firstly train our model with $loss_i^h = 0$ until the loss is converged. Then we use the trained model

[1] https://github.com/google-research/bert

Train/Test	Model	R@1	R@10	R@50	R@100
5%/95%	Three-T Emb	1.02	3.41	7.05	9.34
	Three-T BERT	1.13	5.28	12.14	17.08
	QuadrupletBERT	**6.28**	**9.59**	**16.41**	**21.62**
80%/20%	Three-T Emb	18.25	41.08	61.39	68.41
	Three-T BERT	21.04	43.29	64.17	71.79
	QuadrupletBERT	**28.15**	**59.64**	**75.39**	**81.11**
5%/95%	Three-T Emb	0.26	1.04	1.99	2.53
	Three-T BERT	0.39	1.92	2.98	3.08
	QuadrupletBERT	**3.11**	**5.76**	**7.84**	**9.19**
80%/20%	Three-T Emb	9.59	33.94	50.21	55.18
	Three-T BERT	16.88	41.27	59.28	65.56
	QuadrupletBERT	**19.84**	**50.33**	**68.82**	**74.83**

Table 1: Recall@k on two datasets, where three-T Emb represents the three-tower word embedding retrieval method (Huang et al., 2020) and Three-T BERT represents the three-tower Sentence-BERT (Reimers and Gurevych, 2019). The top half of the table are results of SQuAD; bottom half are results of Natural Questions. Numbers are in percentage (%).

to retrieve a candidate set $\mathcal{C}_i$ for q_i. We randomly select a sentence in $\mathcal{C}_i$ as hn_i.

For each dataset, we consider different training/test split of the data (5%/95% and 80%/20%) in the fine-tuning stage, and the 10% of the training set is held out as the validation set for hyperparameter tuning. The split is created assuming a cold-start retrieval scenario where the queries in the test (query, document) pairs are not seen in training.

We compare our method against two famous embedding-based large-scale retrieval baselines: (1) Recent three-tower word embedding retrieval method proposed by Facebook Search (Huang et al., 2020). (2) The state-of-the-art three-tower Sentence-BERT proposed by Reimers and Gurevych (2019).

4.2 Evaluation Metric

Since the goal of *Retriever* is to capture the positives in the top-k results, we select **Recall@k** as the evaluation metric. The following equation computes Recall@k:

$$Recall@k = \frac{1}{|D|} \sum_{x_i \in D} \frac{\sum_{y_i \in R_k} l_{<x_i, y_i>}}{\sum_{y_i \in D} l_{<x_i, y_i>}} \quad (5)$$

where R_k is the top k results recalled by our model. D is the dataset. x_i and y_i are the i-th question and i-th answer separately.

Train/Test	Model	R@1	R@10	R@50	R@100
5%/95%	$m = 0$	4.06	6.24	13.83	18.01
	$m = 0.1$	**6.28**	**9.59**	**16.41**	**21.62**
	$m = 0.2$	5.01	8.24	14.45	20.08
	$m = 1$	4.28	7.91	13.81	18.63
	$m = 1.5$	3.11	4.94	12.15	17.19
	$m = 2$	2.09	3.12	8.27	14.33
80%/20%	$m = 0$	21.67	37.03	57.18	73.02
	$m = 0.1$	**28.15**	**59.64**	**75.39**	**81.11**
	$m = 0.2$	22.08	37.91	57.95	74.77
	$m = 1$	19.89	31.02	46.18	66.59
	$m = 1.5$	17.23	28.11	42.03	60.09
	$m = 2$	15.25	24.23	37.17	55.68
5%/95%	$m = 0$	1.05	4.13	6.28	7.91
	$m = 0.1$	2.78	4.91	6.63	8.28
	$m = 0.2$	**3.11**	**5.76**	**7.84**	**9.19**
	$m = 1$	2.04	4.48	7.37	8.12
	$m = 1.5$	1.81	3.81	5.71	6.92
	$m = 2$	1.72	2.21	3.53	4.09
80%/20%	$m = 0$	17.18	46.93	65.14	71.03
	$m = 0.1$	18.43	47.89	66.14	72.27
	$m = 0.2$	**19.84**	**50.33**	**68.82**	**74.83**
	$m = 1$	16.07	42.11	64.03	69.71
	$m = 1.5$	14.02	40.49	60.55	64.13
	$m = 2$	12.44	38.39	57.13	60.34

Table 2: Experimental results of finetuning m. The top half of the table are results of SQuAD; bottom half are results of Natural Questions. Numbers are in percentage (%).

4.3 Overall Results

The experimental results[2] are shown in the table 1. We can see that:

1. Results of both Sentence-BERT and our QuadrupletBERT overpass the results of three tower word embedding, which confirms the effectiveness of BERT-style retrieval model.

2. Our four-tower QuadrupletBERT models gain improvements over the three-tower BERT. It is worth noting that the only difference between them is that our model leverages distances between hard negative and simple negative instances in the training process by an extra tower, which verifies our assumption.

3. Our QuadrupletBERT models surpass all the baseline models in all tasks, which verifies our method's effectiveness again. Especially the results on cold-start retrieval (5%/95% training/test split) tasks demonstrate our models keep improvements even on data-lacking scenarios.

[2] The experiment results in this paper are statistically significant with $p < 0.05$.

5 Hyper-Parameter Finetuning

As a key hyper-parameter of our QuadrupletBERT model, m denotes the margin enforced between positive and hard negative and negative instances. We further investigated the influence of m on our model.

With the SQuAD and Natural Questions datasets, we train models with m is set to 0, 0.1, 0.2, 1, 1.5, and 2, respectively. The experimental results are shown in Table 2. We found that tuning margin value is important – the optimal margin value varies a lot across different training tasks, and different margin values result in $5-10\%$ recall variance.

6 Related Work

We have covered research on embedding based large-scale retrieval in Section 1, related work that inspires our technical design is mainly introduced in the following:

Reimers and Gurevych (2019) present a modification of the pre-trained BERT network that uses multi-tower network structures to derive semantically meaningful sentence embeddings that can be compared using cosine similarity.

Huang et al. (2020) present a multi-tower word embedding retrieval method successfully applied in the Facebook online search. Besides, they mentioned that shuffling hard negative and simple negative instances in the training sets may help model learning, which inspired us to further investigate the effectiveness of hard negative instances.

7 Conclusion

We have presented our four-tower Quadruplet-BERT model and demonstrated its usage and effect on large-scale retrieval. Unlike many widely-used BERT-style *Ranker* models of large-scale retrieval tasks, our model focus on the retrieval phase. The multi-tower architecture making it extremely easy to be applied in retrieval systems. Moreover, incorporating distances between hard negative and simple negative instances into the training process shows significant superiority in improving *Retriever* model performance.

We hope our work can inspire more sophisticated techniques of leveraging BERT-style models in large-scale retrieval. Leveraging hard negative instances for other natural language processing tasks such as text generation and information extraction is also worth investigating.

Acknowledgments

This research was supported by the National Key Research And Development Program of China (No.2019YFB1405802), the 2019 Industrial Internet Innovation and Development Project No. TC190H46G/1, and the central government guides local science and technology development fund projects (science and technology innovation base projects) No. 206Z0302G.

References

Amin Ahmad, Noah Constant, Yinfei Yang, and Daniel Cer. 2019. Reqa: An evaluation for end-to-end answer retrieval models. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering, MRQA@EMNLP 2019, Hong Kong, China, November 4, 2019*, pages 137–146. Association for Computational Linguistics.

Wei-Cheng Chang, Felix X. Yu, Yin-Wen Chang, Yiming Yang, and Sanjiv Kumar. 2020. Pre-training tasks for embedding-based large-scale retrieval. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Ruiqi Guo, Sanjiv Kumar, Krzysztof Choromanski, and David Simcha. 2016. Quantization based fast inner product search. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, AISTATS 2016, Cadiz, Spain, May 9-11, 2016*, volume 51 of *JMLR Workshop and Conference Proceedings*, pages 482–490. JMLR.org.

Jui-Ting Huang, Ashish Sharma, Shuying Sun, Li Xia, David Zhang, Philip Pronin, Janani Padmanabhan, Giuseppe Ottaviano, and Linjun Yang. 2020. Embedding-based retrieval in facebook search. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pages 2553–2561. ACM.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2017. Billion-scale similarity search with gpus. *CoRR*, abs/1702.08734.

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. *CoRR*, abs/1907.11692.

Wenhao Lu, Jian Jiao, and Ruofei Zhang. 2020. Twinbert: Distilling knowledge to twin-structured compressed BERT models for large-scale retrieval. In *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*, pages 2645–2652. ACM.

Yosi Mass and Haggai Roitman. 2020. Ad-hoc document retrieval using weak-supervision with BERT and GPT2. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4191–4197, Online. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 3111–3119.

Thiago Akio Nakamura, Pedro H Calais, Davi de Castro Reis, and André Paim Lemos. 2019. An anatomy for neural search engines. *Information Sciences*, 480:339–353.

Ramith Padaki, Zhuyun Dai, and Jamie Callan. 2020. Rethinking query expansion for BERT reranking. In *Advances in Information Retrieval - 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14-17, 2020, Proceedings, Part II*, volume 12036 of *Lecture Notes in Computer Science*, pages 297–304. Springer.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 3980–3990. Association for Computational Linguistics.

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

Anshumali Shrivastava and Ping Li. 2014. Asymmetric LSH (ALSH) for sublinear time maximum inner product search (MIPS). In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2321–2329.

Karen Spärck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21.

Karen Spärck Jones. 2004. Idf term weighting and ir research lessons. *Journal of documentation*, 60(5):521–523.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5998–6008.

Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. 2019. End-to-end open-domain question answering with bertserini. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Demonstrations*, pages 72–77. Association for Computational Linguistics.

Zeynep Akkalyoncu Yilmaz, Shengjin Wang, Wei Yang, Haotian Zhang, and Jimmy Lin. 2019. Applying BERT to document retrieval with birch. 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 2019, Hong Kong, China, November 3-7, 2019 - System Demonstrations*, pages 19–24. Association for Computational Linguistics.