# A Little Bit Is Worse Than None: Ranking with Limited Training Data

**Xinyu Zhang**,[1] **Andrew Yates**,[2] and **Jimmy Lin**[1]
[1] David R. Cheriton School of Computer Science, University of Waterloo
[2] Max Planck Institute for Informatics

## Abstract

Researchers have proposed simple yet effective techniques for the retrieval problem based on using BERT as a relevance classifier to rerank initial candidates from keyword search. In this work, we tackle the challenge of fine-tuning these models for specific domains in a data and computationally efficient manner. Typically, researchers fine-tune models using corpus-specific labeled data from sources such as TREC. We first answer the question: How much data of this type do we need? Recognizing that the most computationally efficient training is no training, we explore zero-shot ranking using BERT models that have already been fine-tuned with the large MS MARCO passage retrieval dataset. We arrive at the surprising and novel finding that "some" labeled in-domain data can be worse than none at all.

## 1 Introduction

Given a corpus $\mathcal{C}$ comprised of an arbitrary number of texts, the goal of the retrieval task is to generate a ranked list of $k$ results for a user query $q$ that maximizes some metric of interest. Texts can differ in length: if the corpus is comprised of paragraph-length segments, the task is referred to as *passage retrieval*. Otherwise, information retrieval (IR) researchers use the term *document retrieval*.

BERT (Devlin et al., 2019) has been successfully applied to the passage retrieval task by using it as a relevance classifier that reranks an initial list of candidate results (Nogueira and Cho, 2019), which are retrieved using bag-of-words queries and efficient exact-match scoring techniques such as BM25. As passages are usually shorter than the 512 token input length limit of BERT, this solution is straightforward. Even in cases where the candidate text exceeds this length limitation, Dai and Callan (2019) showed that simply taking the best-scoring passage from a longer document as a proxy for the document score is an effective technique. In keeping with the theme of this workshop, these are simple yet effective approaches to tackling the retrieval problem.

Building on these two previous innovations, our work tackles the problem of training ranking models for specific domains (corpora) in a data and computationally efficient manner. While the most straightforward solution would be to gather relevance judgments on the target corpus, this is a nontrivial task. Such IR test collections are usually produced via efforts like the Text Retrieval Conferences (TRECs) organized by the U.S. National Institute for Standards and Technology. These collections are the result of community-wide efforts and beyond the capabilities of individual research teams. We consider the question of how much training data are needed to fine-tune an effective ranking model. Does adapting a BERT-based ranker to a new domain require TREC-like levels of effort?

The most data and computationally efficient training procedure is, of course, no training at all—that is, zero-shot learning. Given the appeal of skipping the fine-tuning process altogether, we also explore how a zero-shot approach compares to fine-tuning on the target domain. There exists publicly available BERT models that have already been fine-tuned with existing labeled data, for example, in the "model zoo" of HuggingFace's Transformer library (Wolf et al., 2019). We explore using these directly on our target corpora, and arrive at the interesting finding that a bit of labeled in-domain data can be worse than having none at all. In other words, if we don't have sufficient in-domain training data, it's better to simply adopt a zero-shot ranking approach using an already fine-tuned model: for this task, "few shot" is worse than "zero shot"! The primary contribution of this paper is an explication of this surprising finding that, to our knowledge, has not been reported in the literature.

*Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing*, pages 107–112
Online, November 20, 2020. ©2020 Association for Computational Linguistics

## 2 Related Work

While BERT's pretraining has reduced the burden of applying the model to downstream tasks, BERT's maximum input length of 512 tokens presents a challenge for document retrieval. This length limitation prevents the straightforward application of BERT to documents in typical corpora used for retrieval tasks, which are frequently longer. The obvious solution is to split documents into smaller passages, but this immediately raises the question of how to construct "passage-level" relevance labels from document-level labels. Dai and Callan (2019) proposed the simple strategy of giving all passages the same label as the document (at training time) and aggregating passage scores (at inference time). Their most effective approach, BERT–MaxP, uses the maximum passage score as the document score at inference (ranking) time.

Alternatively, this obstacle can be entirely avoided with a zero-shot approach: the model is fine-tuned on a passage retrieval dataset and then directly applied to the target corpus (Yilmaz et al., 2019; Nogueira et al., 2020). For example, Yilmaz et al. (2019) found that when BERT was fine-tuned on a combination of (out-of-domain) datasets, the model exhibited state-of-the-art effectiveness (at the time) on Robust04. Nogueira et al. (2020) confirmed this finding and further improved zero-shot effectiveness on Robust04 by fine-tuning T5 (Raffel et al., 2020) on the MS MARCO passage dataset (Bajaj et al., 2018). Cohen et al. (2018) investigated the use of adversarial regularization to prevent pre-BERT neural models from learning representations closely tied to a specific domain. They found that training on a dataset fused from multiple domains was effective and can be further improved using adversarial regularization.

In a supervised setting with transformers, fine-tuning on a related "intermediate" dataset before fine-tuning on the target dataset can be beneficial. Phang et al. (2018) was the first to show this for natural language inference tasks, and the evidence is consistent for retrieval tasks. Dai and Callan (2019) showed that fine-tuning BERT on Bing search logs before fine-tuning on a TREC dataset improved BERT–MaxP's effectiveness. Similarly, Li et al. (2020) found that BERT–MaxP also benefits from intermediate fine-tuning on MS MARCO.

As expected, we encounter diminishing returns in effectiveness improvements as the amount of labeled training data increases; that is, increasing amounts of data are needed to obtain further improvements. Nogueira et al. (2020) demonstrated this on the MS MARCO passage ranking dataset. However, to the best of our knowledge, no previous work has investigated the effect of training data size for traditional TREC-style datasets on BERT-based models, which are smaller than MS MARCO by orders of magnitude.

Beyond ranking tasks, methods for tackling the limited labeled data issue using transfer learning have also been investigated. Rietzler et al. (2019) conducted supervised learning on the source dataset and unsupervised learning on the target dataset. Ma et al. (2019) employed adversarial learning to generate pseudo-labels for target datasets. Interestingly, both papers reported that directly transferring knowledge learned from a supervised out-of-domain dataset or unsupervised in-domain dataset to a target domain consistently underperforms supervised in-domain training without the intervention of special techniques (e.g., adversarial regularization).

## 3 Methodology

In order to analyze the impact of fine-tuning a BERT ranking model with limited training data, we sample standard benchmark datasets to simulate having less data available. Rather than proposing a new model, we use the BERT–MaxP model (Dai and Callan, 2019) due to its simplicity and demonstrated effectiveness on several datasets.

To simulate the impact of having limited data, we prepare six different datasets that comprise relevance judgments sampled from the full dataset at a sampling rate $r \in \{0.1, 0.3, 0.5, 0.7, 0.9, 1.0\}$. The setting $r = 1.0$ is equivalent to using the full dataset. Specifically, given a dataset with $N$ queries and $M$ relevance judgements, the $r$-sampled dataset contains roughly $r \times N$ queries and exactly $r \times M$ judgements. That is, queries (along with all their associated judgments) are dropped with a higher priority. This is accomplished by randomly dropping a query until doing so would result in fewer than $r \times M$ judgments. When this condition is reached, we loop over the remaining queries, randomly removing one judgment per query until there are exactly $r \times M$ judgments remaining. When we split our datasets into training, validation, and test folds for experiments, sampling is applied to only training and validation; we always calculate evaluation metrics using all available judgments.

| $r$ | No. of judgements | No. of queries | No. avg. docs per query |
|---|---:|---:|---:|
| (a) Robust04 | | | |
| 0.1 | 31,141 | 25 | 1,245 |
| 0.3 | 93,423 | 79 | 1,182 |
| 0.5 | 155,705 | 125 | 1,245 |
| 0.7 | 217,987 | 175 | 1,245 |
| 0.9 | 280,269 | 229 | 1,223 |
| 1.0 | 311,410 | 249 | 1,250 |
| (b) GOV2 | | | |
| 0.1 | 13,535 | 20 | 676 |
| 0.3 | 40,605 | 51 | 796 |
| 0.5 | 67,676 | 83 | 815 |
| 0.7 | 94,746 | 114 | 831 |
| 0.9 | 121,816 | 140 | 870 |
| 1.0 | 135,352 | 149 | 908 |

Table 1: Robust04 (1a) and GOV2 (1b) statistics, where $r$ is the sampling rate and $r = 1.0$ corresponds to the full dataset.

## 4 Experiments

### 4.1 Datasets

We conduct experiments on two standard TREC benchmarks from different domains, namely the Robust04[1] and GOV2[2] collections. Robust04 is a collection of newswire documents, whereas GOV2 contains crawled websites under the `.gov` domain. Summary statistics are shown in Table 1. On both datasets, we consider only keyword queries.

On Robust04, we use 5-fold cross-validation with three folds for training, one for validation, and the other for evaluation, matching the splits in (Yang et al., 2019). On GOV2, we randomly split the queries into three groups and run 3-fold cross-validation with one fold for training, one for validation, and the final for evaluation.

### 4.2 Experimental setup

Following previous work, we initialize BERT–MaxP with the BERT-Base model (Dai and Callan, 2019; Li et al., 2020). For experiments involving MS MARCO fine-tuning prior to fine-tuning on the target domain (i.e., using MS MARCO as an intermediate dataset), we initialize BERT–MaxP with the BERT-Base checkpoint released by Nogueira and Cho (2019).[3] To obtain candidate documents to rerank, we use Anerini's implementation of BM25 with its default parameters ($k_1 = 0.9$, $b = 0.4$) as the first-stage ranker (Yang et al., 2017). BERT–MaxP reranks the top 100 candidate documents at test time and uses the top 1000 during training.

For both datasets, we split documents into a maximum of 30 overlapping passages. Each passage contains 150 tokens and we use a stride of 75 tokens. Following the original work,[4] passages after the first are randomly selected with probability 0.1 during training.

All experiments use pairwise hinge loss over 36 epochs, where one epoch contains 256 batches. Each batch consists of 16 training pairs. We use the Adam optimizer (Kingma and Ba, 2014) with $lr = 10^{-3}$ for non-BERT parameters and $lr = 10^{-5}$ for BERT parameters; other parameters are $\beta_1 = 0.9, \beta_2 = 0.999$, and $\epsilon = 10^{-7}$. The validation set is used to determine the best model for evaluation. We implement our experiments in Capreolus (Yates et al., 2020), a toolkit for ad hoc retrieval with neural models. Our models are trained with Tensorflow 2.3. We perform the Robust04 and GOV2 experiments on TPU v2-8 and NVIDIA Quadro RTX 8000, respectively.

We only report effectiveness in terms of nDCG@20 due to space limitations, but we observed similar trends for mAP and P@20. When experimenting with different sampling rates, we run each model configuration five times and report the median nDCG@20. Our code and experimental outputs are available on GitHub.[5]

## 5 Results and Discussion

In this section, we investigate BERT–MaxP's effectiveness on the Robust04 and GOV2 datasets when trained with limited data. Figure 1 shows the model's effectiveness as the amount of training data increases both with and without first fine-tuning on the intermediate MS MARCO dataset. We also include BM25 (our first-stage retrieval), the unsupervised BM25+RM3 query expansion approach, and a zero-shot model in which BERT–MaxP is fine-tuned only on MS MARCO. These correspond

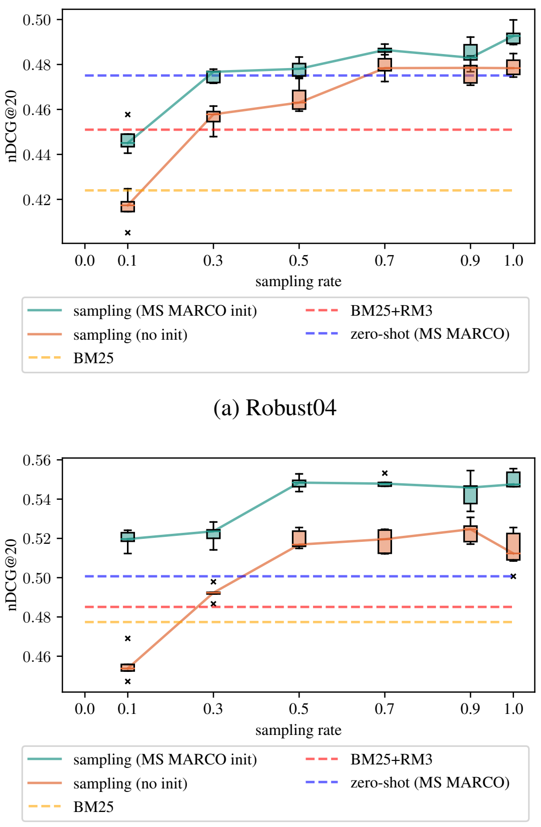
[1] `https://trec.nist.gov/data/robust/04.guidelines.html`

[2] `http://ir.dcs.gla.ac.uk/test_collections/gov2-summary.htm`

[3] `https://github.com/nyu-dl/dl4marco-bert`

[4] `https://github.com/AdeDZY/SIGIR19-BERT-IR/blob/master/run_qe_classifier.py#L468-L471`

[5] `https://github.com/crystina-z/a-little-bit-is-worse-than-none`



(a) Robust04

(b) GOV2

Figure 1: nDCG@20 on Robust04 (1a) and GOV2 (1b) fine-tuned on $r$-sampled judgements. The box plot shows the distribution of the five runs while the lines connect the median values for each condition. The × symbol indicates outliers.

to horizontal dotted lines in the plots. Corresponding numerical values can be found in Table 2.

**Impact of dataset size.** As expected, we observe in Figure 1 that effectiveness generally increases as the amount of training data grows (i.e., as the sampling rate increases) on both collections regardless of whether the model is first fine-tuned on MS MARCO. The increase in effectiveness is most obvious when the amount of data is small, e.g., from $r = 0.1$ to 0.3.

Effectiveness appears to plateau at $r = 0.7$, which is surprising given that both corpora contain relatively small numbers of examples when compared to the sizes of datasets typically used in deep learning today. Effectiveness even drops slightly when the full GOV2 dataset is used (when not using MS MARCO). This suggests that the amount of relevance judgments available in both corpora is sufficient for fine-tuning the BERT–MaxP model.

| | **Robust04** | | **GOV2** | |
|---|---|---|---|---|
| BM25 | 0.4240 | | 0.4774 | |
| BM25RM3 | 0.4510 | | 0.4851 | |
| Zero shot | 0.4751 | | 0.5007 | |
| | w/ MS | w/o MS | w/ MS | w/o MS |
| Dai and Callan (2019) | – | 0.469 | – | – |
| Li et al. (2020) | 0.4931 | – | 0.560 | – |
| $r = 0.1$ | 0.4451 | 0.4173 | 0.5197 | 0.4538 |
| $r = 0.3$ | 0.4767 | 0.4578 | 0.5236 | 0.4923 |
| $r = 0.5$ | 0.4781 | 0.4630 | 0.5484 | 0.5168 |
| $r = 0.7$ | 0.4865 | 0.4784 | 0.5479 | 0.5196 |
| $r = 0.9$ | 0.4830 | 0.4785 | 0.5459 | 0.5247 |
| $r = 1.0$ | 0.4927 | 0.4929 | 0.5475 | 0.5123 |

Table 2: Tabular presentation of median nDCG@20 scores from Figure 1, compared to previously reported scores. Columns "w/ MS" and "w/o MS" indicate training with and without intermediate MS MARCO fine-tuning, respectively.

**Zero-shot effectiveness.** Surprisingly, fine-tuning BERT with in-domain data is sometimes *worse* than zero shot, i.e., worse than not using in-domain data. From Figure 1, we see that this occurs up to $r = 0.3$ on both Robust04 and GOV2 when no intermediate dataset is used. On Robust04, this also occurs at $r = 0.1$ even when the model is initialized with the MS MARCO checkpoint. At a sampling rate of $r = 0.3$, fine-tuning on in-domain data directly (without MS MARCO) barely beats the traditional BM25+RM3 approach, which does not involve any neural network. When using intermediate data, however, the models are able to beat this non-neural baseline more easily.

Note that zero-shot BERT is more effective than BM25+RM3, which confirms that all these observed effectiveness differences are the result of different fine-tuning procedures using in-domain data (i.e., Robust04 or GOV2). Interestingly, even using all available data for Robust04, the model only modestly improves over the zero-shot baseline. This finding is consistent with the recent work of Nogueira et al. (2020), who eschew in-domain training data completely in the context of ranking with sequence-to-sequence models.

**Computational efficiency.** Although competitive (and in some cases, even better) effectiveness results can be obtained without using all available judgments (i.e., with $r < 1.0$), these settings do not appear to be more computationally efficient; the total training time remains roughly the same. In other words, it is not the case that we regularly reach peak validation effectiveness earlier when fine-tuning with fewer judgments.

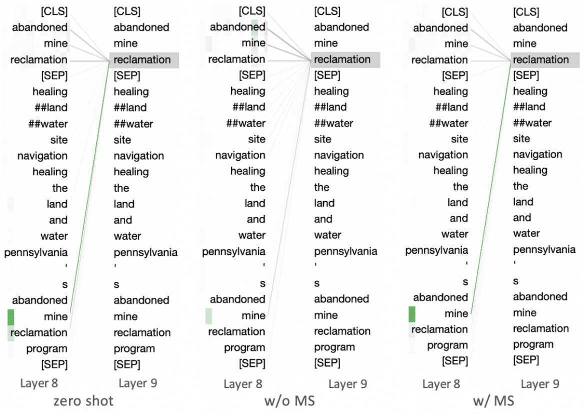


Figure 2: Attention visualization from three BERT models, where "w/o MS" and "w/ MS" indicate without and with MS MARCO fine-tuning, respectively. Colors refer to different attention heads. Deeper colors indicate stronger attention. All attention connections are from layer 8 to layer 9.

**Attention visualizations.** To investigate how MS MARCO increases effectiveness in other domains, we visualize attention from three models using the BertViz toolkit (Vig, 2019): a zero-shot model, a model fine-tuned on GOV2 directly, and a model fine-tuned on GOV2 after first fine-tuning on MS MARCO. Figure 2 compares the attention that the query term "reclamation" received in each model when predicting the relevance between the query *"abandoned mine reclamation"* and the text *"healinglandwater site navigation healing the land and water pennsylvania's abandoned mine reclamation program"*. Both the query and the text snippet are from GOV2. The attention map visualizes interactions from layer 8 to layer 9 in the models.

From Figure 2, it can be observed that the model with prior MS MARCO fine-tuning (right) captures a similar pattern as the zero-shot model (left), where "reclamation" receives the strongest signal from "mine", its bigram complement. However, this particular relation is more weakly captured by the model without prior fine-tuning (middle). This suggests that one way in which fine-tuning on a large intermediate dataset could help is by providing a more accurate basis for determining the relationships between terms for retrieval tasks, which might be hard to learn with only a small amount of (target domain) training data. While this particular attention analysis is anecdotal, we do observe many similar instances. Nevertheless, how to precisely determine what a BERT model learns from fine-tuning remains an open question.

## 6 Conclusions

This paper shows that, on two TREC collections from different domains, the effectiveness of a fine-tuned BERT–MaxP model plateaus as the amount of available judgements increases. This suggests that the current sizes of TREC test collections are sufficient for training with current BERT architectures for document retrieval. We find that performing zero-shot learning by adapting a model trained on a different domain provides a strong baseline and can even be a substitute for in-domain fine-tuning under data-poor conditions. Whether these results are due to limitations with existing datasets (e.g., their annotation schemes), ranking models (e.g., their inability to extract more signal), or the training procedure (e.g., hyperparameter settings to properly mix out-of-domain and in-domain data) remains an interesting open-research question.

Our findings present interesting guidance to practitioners: before embarking on any annotation effort in a document ranking task, it would be wise to first carefully plan out the amount of resources that are available. Our experimental results show that a bit of data can be worse than none!

## Acknowledgments

This research was supported in part by the Canada First Research Excellence Fund and the Natural Sciences and Engineering Research Council (NSERC) of Canada. In addition, we would like to thank Google Cloud and TensorFlow Research Cloud for credits to support this work.

## References

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2018. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. *arXiv preprint arXiv:1611.09268v3*.

Daniel Cohen, Bhaskar Mitra, Katja Hofmann, and W. Bruce Croft. 2018. Cross domain regularization for neural ranking models using adversarial learning. In *Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1025–1028.

Zhuyun Dai and Jamie Callan. 2019. Deeper text understanding for IR with contextual neural language modeling. In *Proceedings of the 42nd Annual International ACM SIGIR Conference on Research and*

*Development in Information Retrieval (SIGIR 2019)*, pages 985–988.

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.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Canjia Li, Andrew Yates, Sean MacAvaney, Ben He, and Yingfei Sun. 2020. PARADE: Passage representation aggregation for document reranking. *arXiv preprint arXiv:2008.09093*.

Xiaofei Ma, Peng Xu, Zhiguo Wang, Ramesh Nallapati, and Bing Xiang. 2019. Domain adaptation with bert-based domain classification and data selection. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 76–83.

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with BERT. *arXiv preprint arXiv:1901.04085*.

Rodrigo Nogueira, Zhiying Jiang, Ronak Pradeep, and Jimmy Lin. 2020. Document ranking with a pre-trained sequence-to-sequence model. In *Findings of EMNLP*.

Jason Phang, Thibault Févry, and Samuel R. Bowman. 2018. Sentence encoders on STILTs: Supplementary Training on Intermediate Labeled-data Tasks. *arXiv preprint arXiv:1811.01088*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Alexander Rietzler, Sebastian Stabinger, Paul Opitz, and Stefan Engl. 2019. Adapt or get left behind: Domain adaptation through BERT language model finetuning for aspect-target sentiment classification. *arXiv preprint arXiv:1908.11860*.

Jesse Vig. 2019. A multiscale visualization of attention in the transformer model. *arXiv preprint arXiv:1906.05714*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2019. HuggingFace's Transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Peilin Yang, Hui Fang, and Jimmy Lin. 2017. Anserini: enabling the use of Lucene for information retrieval research. In *Proceedings of the 40th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2017)*, pages 1253–1256.

Wei Yang, Kuang Lu, Peilin Yang, and Jimmy Lin. 2019. Critically examining the neural hype weak baselines and the additivity of effectiveness gains from neural ranking models. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1129–1132.

Andrew Yates, Kevin Martin Jose, Xinyu Zhang, and Jimmy Lin. 2020. Flexible IR pipelines with Capreolus. In *Proceedings of the 29th International Conference on Information and Knowledge Management (CIKM 2020)*.

Zeynep Akkalyoncu Yilmaz, Wei Yang, Haotian Zhang, and Jimmy Lin. 2019. Cross-domain modeling of sentence-level evidence for document retrieval. 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 3481–3487.