BERT meets Cranfield: Uncovering the Properties of Full Ranking on Fully Labeled Data

Negin Ghasemi
Radboud University, Nijmegen
N.Ghasemi@cs.ru.nl

Djoerd Hiemstra
Radboud University, Nijmegen
Hiemstra@cs.ru.nl

Abstract

Recently, various information retrieval models have been proposed based on pre-trained BERT models, achieving outstanding performance. The majority of such models have been tested on data collections with partial relevance labels, where various potentially relevant documents have not been exposed to the annotators. Therefore, evaluating BERT-based rankers may lead to biased and unfair evaluation results, simply because a relevant document has not been exposed to the annotators while creating the collection. In our work, we aim to better understand a BERT-based ranker's strengths compared to a BERT-based re-ranker and the initial ranker. To this aim, we investigate BERT-based rankers performance on the Cranfield collection, which comes with full relevance judgment on all documents in the collection. Our results demonstrate the BERT-based full ranker's effectiveness, as opposed to the BERT-based re-ranker and BM25. Also, analysis shows that there are documents that the BERT-based full-ranker finds that were not found by the initial ranker.

1 Introduction

Transformers-based pre-trained language representations, such as ELMo (Peters et al., 2018), BERT (Devlin et al., 2019), and T5 (Raffel et al., 2020) have been counted as a promising approaches to various information retrieval tasks, such as document ranking (Nogueira and Cho, 2019) and question answering (Yang et al., 2019a).

Prior work argued that utilizing BERT for ranking can achieve state-of-the-art results on popular ad-hoc retrieval collections such as Robust04 (MacAvaney et al., 2019; Akkalyoncu Yilmaz et al., 2019; Yang et al., 2019b; Dai and Callan, 2019), ClueWeb09-B (Dai and Callan, 2019), and MS MARCO (Padigela et al., 2019; Nogueira and Cho, 2019; Nogueira et al., 2019). However, two major limitations make these collections unfair to BERT.

One of the limitations of these collections is that they have not been created to reflect BERT's superiority to its best (Yilmaz et al., 2020). As BERT-based models did not contribute to their assessment pool, testing a BERT-based ranking system with these collections can lead to unfair and biased results (Yilmaz et al., 2020). There may be relevant documents that traditional methods could not find, hence assumed irrelevant in the pooling process. Therefore, new collections are needed or collections with full relevant judgments.

The second limitation is that although reusable test collections play an important role in IR, conducting a lot of research on a single collection can direct the results to an outstanding value by chance (Carterette, 2015).

To avoid these two limitations, we use a previously unused collection with characteristics such as containing a relatively large number of queries in the form of short, full questions. Also, to address the first mentioned limitation, the collection contains a full set of judgments for each query.

The Cranfield[1] collection has all the mentioned features. Unlike other collections, Cranfield's main feature is a complete judgment, so documents uniquely found by a BERT-based ranker can be fairly assessed. This collection is built using abstracts of aerospace-related documents such as papers, research reports and articles from the collection of the College of Aeronautics, Cranfield, England. (Richmond, 1963). The documents' authors were asked to provide a set of related terms for their documents which were turned into natural language queries (Robertson, 2008). The collection contains 225 queries and 1400 documents. The collection has not been used for document ranking

[1] http://ir.dcs.gla.ac.uk/resources/test_collections/cran/

with BERT before and can also address the second limitation. Furthermore, queries and most of the documents are short, which seems to be a good fit for BERT due to the BERT's token limitation.

Using the Cranfield collection, we address the following research questions:

- **RQ1**: Can we replicate a BERT-based re-ranker on a previously unused collection?
- **RQ2**: Does BERT only learn how to re-rank, or can it learn to find relevant documents that are not found by a bag-of-words baseline?

Our work to answer these research questions includes two steps: first, we provide experimental results on the Cranfield collection for document re-ranking with BERT, following the BERT-based re-ranker of Nogueira and Cho (2019) using BM25 (Robertson et al., 1995) as the initial ranker. Second, we study BERT's behavior as a *full*-ranker. In the paper, We refer to a ranker without any initial filtering method as a full-ranker. The code to reproduce our results is available at `https://gitlab.science.ru.nl/nghasemi/bert-meets-cranfield`.

The contributions of this work are as follows:

- We show that document re-ranking with BERT significantly improves the BM25 baseline on an unseen collection, although different hyperparameter settings may be needed.
- Our Analysis of the BERT-based full-ranker reveals better results than the re-ranker. Moreover, the BERT-based full-ranker retrieved documents are quite different from BM25, demonstrating BERT's ability to find relevant documents not found by a bag-of-words approach.

The rest of the paper is structured as follows: Section 2 reviews the related works on BERT and its usage in ranking problems. In Section 3, we describe the model used for document ranking with BERT in more detail. Experimental results and analysis are presented in Section 4. The final Section 5 concludes the paper and discusses potential research questions for future work.

2 Related Work

2.1 BERT

BERT's architecture is mostly designed based on several transformer blocks introduced by Vaswani et al. (2017); however, these blocks only consist of encoders. The authors introduce two differently sized BERT models. BERT-base and BERT-large consist of 12 and 24 transformer blocks, respectively. BERT is trained on two different unsupervised tasks. Train loss is the sum of the mean of both tasks' likelihood. The first task is the Masked Language Model, which trains to predict all the masked words. The second task is Next Sentence Prediction: This task aims to find if the second half of the input is the following sentence of the first half, or a random sentence.

BERT is not limited to the discussed pre-trained tasks. The self-attention mechanism in the transformers block makes BERT capable of modeling many tasks as long as the task inputs are appropriately processed with BERT's desired setups, namely using proper special tokens ($[CLS]$, $[SEP]$) and BERT's specialized tokenizer. Fine-tuning the BERT pre-trained model helps to gain better performance on a specific task. Adding one additional output layer to a BERT pre-trained model is suggested in the fine-tuning phase to minimize the number of learning parameters. We refer readers to Devlin et al. (2019) for more details.

2.2 Collections and BERT-based Rankers

Many valuable collections in the information retrieval community, such as MS MARCO, Robust, and Clueweb, are standard collections for ranking tasks. Robust and Clueweb are popular TREC collections. TREC extensively uses the pooling method to create each collection. They assume that each participant system's top-k ranked items are likely to cover most of the collection's relevant documents; therefore, judging this pool would make a decent collection. The k number for Robust04 is 100 (Voorhees, 2004) and for Clueweb12 is 20 or less (Collins-Thompson et al., 2014). The MS MARCO dataset is formed based on Bing's query samples and related human-generated answers. The corpus was initially formed by retrieving the top-10 passages from the Bing search engine. On average, each query has one relevant passage. However, there are queries with no relevant passages as well (Nguyen et al., 2016).

A recent work proposed by Yilmaz et al. (2020) analyzes the reusability of the collections for information retrieval ranking tasks when a deep neural approach is being used, especially when the collection is created solely using traditional methods.

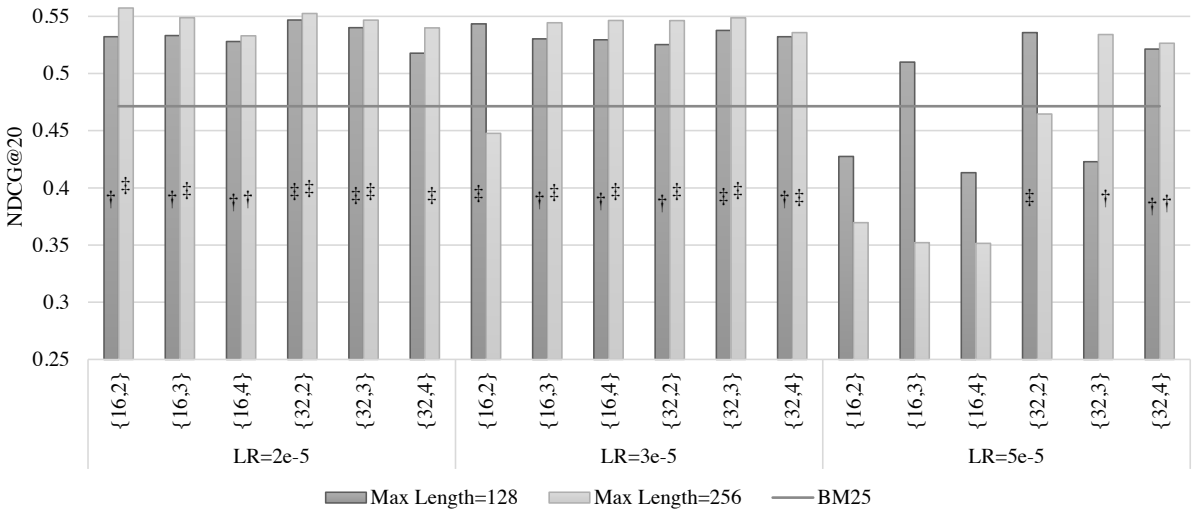

Figure 1: BERT re-ranker results for different hyperparameters. LR shows the value of the Learning Rate. Different batch sizes and epochs are shown as {Batch Size, Epoch}. Max Length shows the BERT token limitation.

The analysis argues that collections created without having neural rankers in their assessment pool should be used cautiously. They may lead to biased results for neural models compared to traditional methods. This work inspired us to use Cranfield, which contains a full set of relevant judgments.

Nogueira and Cho (2019) propose a common re-ranking approach to enhance the top results retrieved by BM25. In their method, a pre-trained BERT model is tuned to find the representation of a concatenated sequence, including query and document tokens.

Experiments by Padigela et al. (2019) show that a BERT-based re-ranker generally achieves higher performance as BM25 is more biased towards higher query term frequency than BERT. Also, the BERT-based re-ranker retrieves documents with more novel words.

Although BERT has shown to be very successful in all the described re-ranking models and many approaches have been proposed to improve the re-ranker, investigating a BERT-based full-ranker performance is still an open question. Of course, BERT-based models use re-ranking for efficiency reasons; however, if the full-ranker outperforms the re-ranker, it can be considered for offline systems and, more importantly, as an essential baseline to create high-quality search collections.

3 Method

Although BERT pre-trained models were trained on large corpora, fine-tuning is necessary for using BERT effectively. Inspired by the work proposed by Nogueira and Cho (2019), we fine-tuned the BERT model and used it to find the representation of the query and document pairs. Due to limited resources, we use the BERT-base, uncased, pre-trained model. This model produces 768-dimensional representation vectors.

Our input vector to BERT is similar to the next sentence prediction task's input vector used in pre-training BERT. Following the BERT's standard input format (Devlin et al., 2019), we converted each query Q and document D to a pair $[CLS] + Q + [SEP] + D + [SEP]$. The query always remains unchanged, but as BERT has a limitation on the number of tokens, we truncate documents that were longer than the model's maximum length.

We use a pre-trained BERT model with an added single linear classification layer on top of the $[CLS]$ output vector to suit the ad-hoc retrieval task. In this case, Each input's $[CLS]$ output vector would be fed to the classification layer, and both the pre-trained BERT model and the additional untrained classification layer is tuned on training samples of queries and documents from the Cranfield collection.

As mentioned in Section 1, the Cranfield collection contains 225 queries and 1400 documents. All queries in the Cranfield collection have full, graded relevance judgments. There are five different relevancy grades. Grades one to four are known to be relevant, and five shows irrelevant documents.

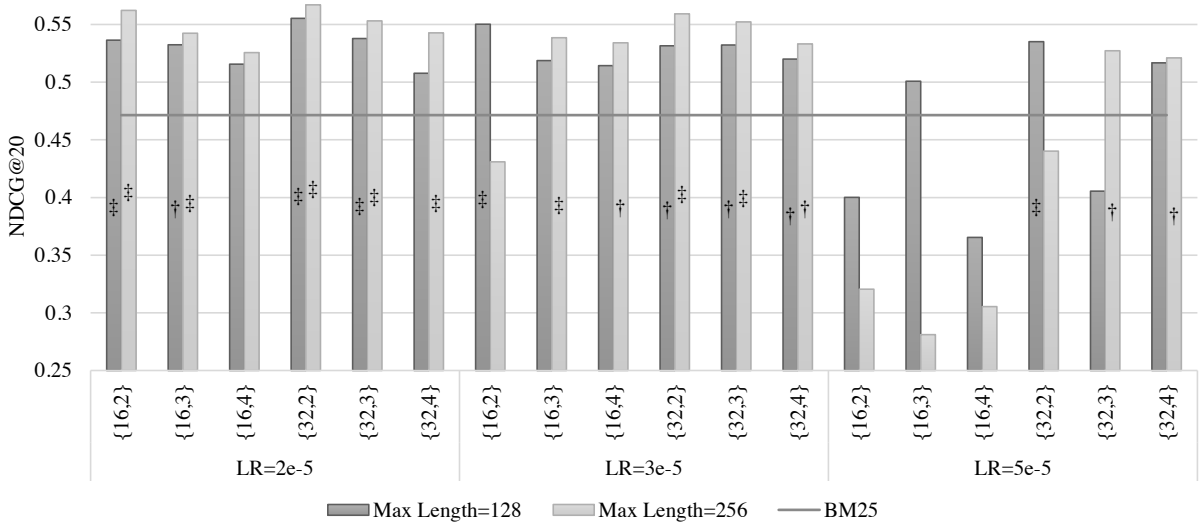

Figure 2: BERT full-ranker results for different hyperparameters. LR shows the value of the Learning Rate. Different batch sizes and epochs are shown as {Batch Size, Epoch}. Max Length shows the BERT token limitation.

To use judgments in classifier tuning, we turn them to binary labels despite its relevance grade. We assumed all the documents with grades one to four to be relevant.

We follow the many methods mentioned in Section 2 that fine-tune BERT on a re-ranking task using the top-k BM25 results. Tuning and testing process is performed using 5-fold cross-validation and the top-100 selection of BM25 results for each query. Folds are split by queries and are available in the code repository.

We test a BERT-based re-ranker and a BERT-based full-ranker, using the same fine-tuning method and hyperparameters. After tuning the BERT model, we use the top-100 BM25 results to do re-ranking, but we use all documents for full-ranking. We evaluate the results using MAP@100, and NDCG@20 (Järvelin and Kekäläinen, 2002) for both models.

4 Results

4.1 Re-ranker Analysis

For our experiments, we use BM25 and a BERT-based re-ranker to address **RQ1**, and we use a BERT-based full-ranker to investigate **RQ2**. We train both models following the hyperparameter value ranges recommended by Devlin et al. (2019).

For the initial ranker, BM25 parameters are as follows: $k1 = 1.5$ and $b = 0.75$. To fine-tune BERT, we use the ADAM optimizer with three different learning rate values of $2e-5$, $3e-5$, and $5e-5$ without the recommended learning rate warmup. Also, we use two batch size values of 16 and 32 and three epoch values of 2, 3, and 4. As the Cranfield collection does not include large documents (the average document length is 118 tokens), we limit the input token length with two different values of 128 and 256. We did not test 512 tokens because we had limited resources available. Results for the BERT-based re-ranker and full-ranker on NDCG@20 are shown in Figure 1 and Figure 2 respectively.

In all demonstrated and tabulated results, † marks a statistically significant difference between the proposed model and the BM25 baseline at $p < 0.05$ based on a two-tailed paired t-test, and ‡ shows highly statistical significance at $p < 0.01$ based on a two-tailed paired t-test.

Experiments show that a lower learning rate is more effective in fine-tuning for the Cranfield collection. The same applies to the number of epochs. We perform our analysis on the model with the learning rate of $2e-5$, batch size of 32, epoch numbers of 2, and the maximum length of input tokens limited to 256. We only report NDCG@20 due to space limitations; however, similar trends were observed with MAP@100.

4.2 Full-ranker Analysis

Table 1 shows the results of two BERT-based ranker models, namely re-ranker and full-ranker. Comparing the full-ranker with the same re-ranker results shows more improvement over the BM25 baseline. In this section, we provide more detail comparing

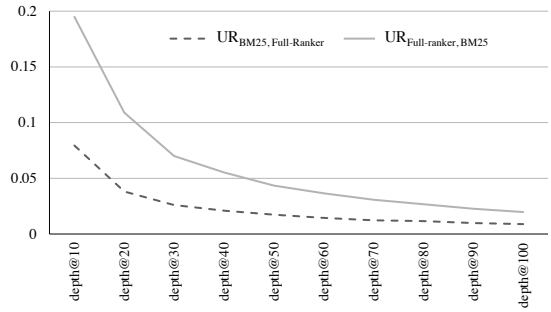

Figure 3: Unique Relevant (UR) percent of documents found by BM25 and the Full-ranker at different depths

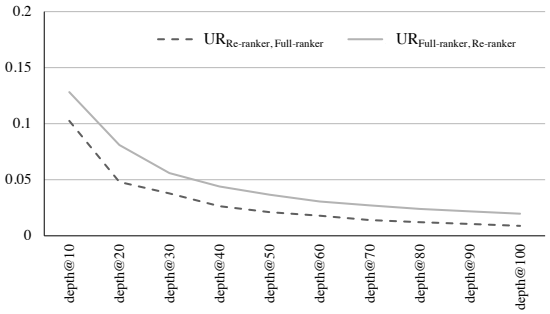

Figure 4: Unique Relevant (UR) percent of documents found by the Re-ranker and the Full-ranker at different depths

the BERT-based full-ranker with the BM25 initial ranker to address **RQ2**.

We observe performance improvement using full-ranking. We believe this behavior is rooted in two possible reasons: (1) The number of unique, relevant documents, which BERT-based ranker finds, that BM25 does not consider; (2) the BERT-based full-ranker can find more highly-ranked new documents.

Method	**MAP@100**	**NDCG@20**
BM25	0.3274	0.4714
Re-ranker	0.4198‡	0.5525‡
Full-ranker	0.4404‡	0.5670‡

Table 1: Results for Learning Rate=$2e-5$, Epoch=2, Batch Size=32, Max Length=256

To investigate these possible reasons, we went through the unique documents each model retrieves. Table 2 presents the results for the percentage of retrieved relevant documents that BM25 found that were not found by the full-ranker, and vice versa, for different depths. We are interested in comparing two different types of retrieved relevant documents: $UR_{BM25,Full-ranker}$ and $UR_{Full-ranker,BM25}$ which are defined as follows:

- $UR_{BM25,Full-ranker}$ (Unique Relevant found by BM25): the relevant documents retrieved by BM25 but not retrieved by the Full-ranker
- $UR_{Full-ranker,BM25}$ (Unique Relevant found by Full-ranker): the relevant documents retrieved by the Full-ranker but not retrieved by the BM25

Figure 3 demonstrates these results in more detail, showing the full-ranker's power to find more relevant documents than BM25, especially in the top results. Besides, We investigated another types of retrieved relevant documents: $UR_{Re-ranker,Full-ranker}$ and $UR_{Full-ranker,Re-ranker}$ which are defined as follows:

- $UR_{Re-ranker,Full-ranker}$ (Unique Relevant found by Re-ranker): the relevant documents retrieved by the Re-ranker but not retrieved by the Full-ranker
- $UR_{Full-ranker,Re-ranker}$ (Unique Relevant found by Full-ranker): the relevant documents retrieved by the Full-ranker but not retrieved by the Re-ranker

Table 3, and Figure 4 show the full-ranker's power in comparison with the re-ranker. It is worth mentioning that the re-ranker and the BM25 retrieved documents are the same in depth@100, but they have different rankings as the re-ranker changes the documents' ranking scores. Although there is a lower difference rate as the re-ranker itself outperforms BM25, the full-ranker still finds more unique, relevant documents than the re-ranker.

As shown in the results, (1) is a correct assumption. The BERT-based full-ranker can find more new relevant documents compared to BM25 and re-ranker, which confirms the **RQ2** hypothesis. It is also worth mentioning that there are 67.6% new documents in the BERT-based full-ranker model in total (both relevant and irrelevant) at top-100 results, which makes it a substantially different ranker than BM25.

Table 4 investigates the relevance degree (RD) of the unique, relevant documents found by each model at their top-100 results. The collection has four different grades indicating the relevancy of

Method	depth@10	depth@20	depth@100
$UR_{BM25,Full-ranker}$	0.08	0.04	0.009
$UR_{Full-ranker,BM25}$	0.19	0.11	0.02

Table 2: Unique Relevant (UR) percent of documents found by BM25 and the full-ranker at different depths

Method	depth@10	depth@20	depth@100
$UR_{Re-ranker,Full-ranker}$	0.1	0.05	0.009
$UR_{Full-ranker,Re-ranker}$	0.13	0.08	0.02

Table 3: Unique Relevant (UR) percent of documents found by the re-ranker and the full-ranker at different depths

Method	RD1	RD2	RD3	RD4
BM25/Re-ranker	0.26	0.45	0.22	0.07
Full-ranker	0.22	0.4	0.23	0.15

Table 4: Percentage of relevant documents per Relevance Degree (RD). RD4 indicates the highest relevance degree.

the document to a query. In this collection, RD4 indicates the highest relevance degree. The observations confirm (2). Results show that the BERT-based full-ranker is more likely to retrieve highly relevant documents.

5 Conclusion

This paper investigates BERT for document ranking. In our experiments, we explore the Cranfield collection, which has not been used on BERT-based ranking approaches and gives new insights because of its characteristics, such as full relevance judgments and a large number of queries. In addition to the document re-ranking with BERT, we considered using a full-ranker under the same experimental settings. The results show that the re-ranker and the full-ranker improve a BM25 baseline significantly. Furthermore, the BERT-based full-ranker outperforms the BERT-based re-ranker. Based on our studies, the BERT-based full-ranker is a different model than the BM25 ranker as it retrieves a notable number of new documents that were not found by BM25. This is especially true for highly relevant documents.

References

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 3490–3496, Hong Kong, China. Association for Computational Linguistics.

Ben Carterette. 2015. The best published result is random: Sequential testing and its effect on reported effectiveness. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, page 747–750, New York, NY, USA. Association for Computing Machinery.

Kevyn Collins-Thompson, Paul Bennett, Fernando Diaz, Charles L. A. Clarke, and Ellen M. Voorhees. 2014. Trec 2013 web track overview. In *Proceedings of the 22nd Text REtrieval Conference (TREC 2013)*.

Zhuyun Dai and Jamie Callan. 2019. Deeper text understanding for ir with contextual neural language modeling. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, page 985–988, New York, NY, USA. Association for Computing Machinery.

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.

Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446.

Sean MacAvaney, Andrew Yates, Arman Cohan, and Nazli Goharian. 2019. Cedr: Contextualized embeddings for document ranking. In *SIGIR*.

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. *arXiv preprint arXiv:1611.09268 (2016)*.

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

Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. 2019. Multi-stage document ranking with bert. *arXiv preprint arXiv:1910.14424*.

Harshith Padigela, Hamed Zamani, and W. Bruce Croft. 2019. Investigating the successes and failures of bert for passage re-ranking. *ArXiv*.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

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

Phyllis A. Richmond. 1963. Review of the cranfield project. *American Documentation*, 14(4):307–311.

Stephen Robertson. 2008. On the history of evaluation in ir. *Journal of Information Science*, 34:439–456.

Stephen Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. 1995. Okapi at trec-3. In *Overview of the Third Text REtrieval Conference (TREC-3)*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, undefinedukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010. Curran Associates Inc.

Ellen M. Voorhees. 2004. Overview of the trec 2004 robust retrieval track. In *Proceedings of the 13nd Text REtrieval Conference (TREC 2004)*.

Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. 2019a. End-to-end open-domain question answering with bertserini. In *NAACL-HLT*.

Wei Yang, Haotian Zhang, and Jimmy Lin. 2019b. Simple applications of bert for ad hoc document retrieval. *ArXiv*, abs/1903.10972.

Emine Yilmaz, Nick Craswell, Bhaskar Mitra, and Daniel Campos. 2020. On the reliability of test collections for evaluating systems of different types. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, page 2101–2104, New York, NY, USA. Association for Computing Machinery.