H_2oloo at TREC 2019: Combining Sentence and Document Evidence in the Deep Learning Track

Zeynep Akkalyoncu Yilmaz, Shengjin Wang, and Jimmy Lin
David R. Cheriton School of Computer Science
University of Waterloo

1 INTRODUCTION

The $h_2$oloo team at the University of Waterloo participated in the TREC 2019 Deep Learning Track for both the document and passage (full) ranking tasks. Our primary goal was to validate BERT-based retrieval techniques that we have been working on [2, 12] in the context of a two-stage retrieval architecture [3] comprising a candidate document generation stage (driven by "bag of words" techniques) followed by a reranking stage built around BERT [4], the popular deep transformer model that takes advantage of massive pretraining using language modeling tasks.

All of our experiments used the open-source Anserini IR toolkit [9, 10], which is based on the popular open-source Lucene search library, for initial retrieval that feeds the BERT-based rerankers. We used a pre-0.6.0 release of Anserini based on Lucene version 8.0. For passage reranking and document reranking of the candidate lists retrieved with Anserini, we used different codebases, which will be described in detailed below.

In addition to our own submissions, at the invitation of the track coordinators, we also prepared a number of baseline runs (to be more specific, runs that did not take advantage of deep learning) to enrich the judgment pool. These runs used a variety of query expansion techniques.

2 PASSAGE RANKING

2.1 Baselines

Our first four runs used BM25 with Anserini's default parameters, alone and in conjunction with three different query expansion approaches based on pseudo-relevance feedback:

- `baseline/bm25base_p`: BM25 baseline using Anserini's default parameters ($k_1 = 0.9$, $b = 0.4$).
- `baseline/bm25base_ax_p`: BM25 baseline using Anserini's default parameters with axiomatic semantic term matching [11].
- `baseline/bm25base_prf_p`: BM25 baseline using Anserini's default parameters with probabilistic relevance feedback [13].
- `baseline/bm25base_rm3_p`: BM25 baseline using Anserini's default parameters with RM3 query expansion [5].

For all query expansion approaches, we used Anserini's default parameters; these exact values are specified as constants in the class `io.anserini.search.SearchArgs`.

The next set of four runs were based on BM25 using parameters tuned with the MS MARCO passage dataset. Since we did not have sufficient time or resources to tune on the *entire* dev set, we performed parameter tuning on sampled subsets. Specifically, tuning was based on five different sets of 10k samples (extracted using the Linux `shuf` command). We tuned on each individual set (grid search on k_1 and b in tenth increments) and then averaged the optimal parameter values across all five sets, which has the effect of regularization. We optimized recall@1000 since Anserini output serves as input to later stage rerankers, and we wanted to maximize the number of relevant documents the rerankers have to work with. The tuned parameters using this method are $k_1 = 0.82$, $b = 0.68$. This led to the following four runs:

- `baseline/bm25tuned_p`: BM25 baseline using tuned parameters ($k_1 = 0.82$, $b = 0.68$).
- `baseline/bm25tuned_ax_p`: BM25 baseline using tuned parameters with axiomatic semantic term matching.
- `baseline/bm25tuned_prf_p`: BM25 baseline using tuned parameters with probabilistic relevance feedback.
- `baseline/bm25tuned_rm3_p`: BM25 baseline using tuned parameters with RM3 query expansion.

For all query expansion approaches, we also used Anserini's default parameters. Note this likely led to sub-optimal effectiveness, since it is likely that BM25 parameters need to be tuned in conjunction with query expansion parameters.

2.2 BERT Runs

The starting point of our deep learning work is the BERT-based passage reranking model of Nogueira and Cho [7], which has served as a competitive baseline for the MS MARCO passage ranking leaderboard since early 2019. In this approach, BERT is used as a relevance classifier trained on query–passage pairs. To this, we added doc2query [8], a document expansion technique that takes advantage of a neural sequence-to-sequence model (also trained on the query–passage pairs). These two techniques were combined into the following submitted runs:

- `h2oloo/p_bert`: We retrieved the top 1000 passages for each query with tuned BM25 and then reranked the passages with the BERT-based relevance classifier [7].
- `h2oloo/p_exp_bert`: Prior to indexing, we expanded each passage using doc2query [8]. To generate diverse expansions, we used two different models: the first is a from-scratch trained model on MS MARCO passage data and the second is the TREC CAR model made available by Nogueira et al., which is exactly the same model used in their paper. We used top 10 sampling for decoding (as recommended by the authors) and simply took the union of the expansions generated by both models. The expansions were appended to the original passages to form an expanded collection. This expanded collection was then indexed with Anserini, which we queried using tuned BM25, followed by reranking with the BERT-based classifier [7].
- `h2oloo/p_exp_rm3_bert`: After expanding and indexing the collection as described for `h2oloo/p_exp_bert`, we retrieved the top 1000 passages with tuned BM25 and RM3 query expansion,

Run	AP	nDCG@10
baseline/bm25base_p	0.3013	0.5058
baseline/bm25base_ax_p	0.3745	0.5511
baseline/bm25base_prf_p	0.3561	0.5372
baseline/bm25base_rm3_p	0.3390	0.5180
baseline/bm25tuned_p	0.2903	0.4973
baseline/bm25tuned_ax_p	0.3632	0.5461
baseline/bm25tuned_prf_p	0.3684	0.5536
baseline/bm25tuned_rm3_p	0.3377	0.5231
h2oloo/p_bert	0.4677	0.7380
h2oloo/p_exp_bert	0.4749	0.7336
h2oloo/p_exp_rm3_bert	0.5049	0.7422
IDST/idst_bert_p1	0.5030	0.7645

Table 1: Passage ranking results.

the results of which are then reranked with the BERT-based relevance classifier.

2.3 Results

Results of our passage ranking runs are shown in Table 1. Note that NIST judgments were provided on a four-point scale: (3) perfectly relevant, (2) highly relevant, (1) related, and (0) irrelevant. For the purposes of computing nDCG, all grades were used, but for computing AP, grade (1) related judgments were *not* considered relevant.[1] For reference, we show results from the best submitted run in terms of nDCG@10 (idst_bert_p1). On a per-team basis in terms of nDCG@10, we ranked number two among all participants, based on our best run (p_exp_rm3_bert). However, in terms of AP, p_exp_rm3_bert was the best submitted run in the evaluation.

Interestingly, we note that the effectiveness of our tuned runs is not much different (and in some cases lower) than the corresponding runs with default BM25 parameters. The reason appears to be that we performed parameter tuning using the sparse MS MARCO judgments, whereas the official evaluation used the dense judgments from NIST assessors. This suggests that parameters may not generalize across different "styles" of relevance judgments.

Our main runs nicely show an increase in effectiveness with growing sophistication of the applied technique. Starting from BERT-based reranking, adding document expansion improves AP (but not nDCG@10), and further introducing query expansion yields even more improvements (in both metrics).

3 DOCUMENT RANKING

3.1 Baselines

Our first four runs used BM25 with default parameters ($k_1 = 0.9, b = 0.4$), alone and in conjunction with three different query expansion approaches based on pseudo-relevance feedback (all using Anserini default parameters):

- baseline/bm25base: BM25 baseline using default parameters ($k_1 = 0.9, b = 0.4$).
- baseline/bm25base_ax: BM25 baseline using default parameters with axiomatic semantic term matching.
- baseline/bm25base_prf: BM25 baseline using default parameters with probabilistic relevance feedback.
- baseline/bm25base_rm3: BM25 baseline using default parameters with RM3 query expansion.

These four runs mirror the four baselines in the passage retrieval condition described in the previous section.

In addition, we submitted four runs using BM25 parameters tuned with the MS MARCO document data. Similar to the passage condition, tuning was performed on five different sets of 10k samples from the training queries (grid search on parameters in tenth increments). The final setting was the average of the optimal parameters across all five sets. We optimized for average precision.

- baseline/bm25tuned: BM25 baseline using tuned parameters ($k_1 = 3.44, b = 0.87$).
- baseline/bm25tuned_ax: BM25 baseline using tuned parameters with axiomatic semantic term matching.
- baseline/bm25tuned_prf: BM25 baseline using tuned parameters with probabilistic relevance feedback.
- baseline/bm25tuned_rm3: BM25 baseline using tuned parameters with RM3 query expansion.

3.2 BERT Runs

Our work on BERT for ranking documents [2, 12] was motivated by the inability of the relevance classifier of Nogueira and Cho [7] to handle long spans of text, as BERT has a 512 token limit. We have discovered a surprisingly simple solution [2]. Given an initial ranked list of documents, we segment each into sentences, and then apply inference (with the relevance classifier) over *each sentence* separately, after which sentence-level scores are aggregated to yield a final score for ranking documents.

Specifically, we combine the top n sentence scores with the original document score as follows:

$$S_f = \alpha \cdot S_{\text{doc}} + (1 - \alpha) \cdot \sum_{i=1}^{n} w_i \cdot S_i \tag{1}$$

where S_{doc} is the original document score and S_i is the i-th top scoring sentence according to BERT. In other words, the relevance score of a document comes from the combination of a document-level term-matching score and evidence contributions from the top sentences in the document as determined by the BERT model. Typically, we find that $n = 3$ achieves optimal effectiveness (and in some cases, $n = 1$). The parameters α and w_i's are learned.

The sentence-level relevance classifier can be trained with the MS MARCO passage data, although we have found that BERT is able to transfer models of relevance across domains [2]. For example, data from the TREC Microblog Tracks [6] are useful for ranking newswire documents (a completely different domain). Since those data comprise (query, tweet, relevance judgment) triples and tweets are quite short, they can be used to directly fine-tune BERT as well. In more detail, we began with a BERT model that has already been fine-tuned on the MS MARCO passage data, and then further fine-tune with the microblog data.

[1] In trec_eval, this is specified using the -l 2 option.

Run	AP	nDCG@10
baseline/bm25base	0.2443	0.5190
baseline/bm25base_ax	0.2452	0.4730
baseline/bm25base_prf	0.2542	0.5106
baseline/bm25base_rm3	0.2772	0.5169
baseline/bm25tuned	0.2318	0.5140
baseline/bm25tuned_ax	0.2816	0.5245
baseline/bm25tuned_prf	0.2759	0.5281
baseline/bm25tuned_rm3	0.2700	0.5485
h2oloo/bm25_marcomb	0.3229	0.6403
h2oloo/bm25exp_marco	0.3030	0.6399
h2oloo/bm25exp_marcomb	0.3190	0.6456
IDST/idst_bert_v3	0.3137	0.7257

Table 2: Document ranking results.

This technique is implemented in our new open-source retrieval toolkit called Birch [1]. With Birch and Anserini, we submitted the following runs:

- h2oloo/bm25_marcomb: We first retrieved the top 1000 documents using tuned BM25 with RM3 query expansion. These are then reranked using the approach described above, with the relevance classifier fine-tuned on both MS MARCO passage data and microblog data. Weights for the top three sentences ($n = 3$, selected based on our previous experiences [2]) are learned from the dev set of the MS MARCO document data.
- h2oloo/bm25exp_marcomb: Prior to indexing, we expanded each document using doc2query [8]. To generate diverse expansion terms, we used exactly the same approach as in the passage task (i.e., union of two different models). The expansion of a document was defined as the union of the expansion of each passage in the document. We retrieved and reranked documents from the index with the expanded documents in the same way as h2oloo/bm25_marcomb.
- h2oloo/bm25exp_marco: This condition differs from the previous condition only in the relevance classifier used to rerank the documents: instead of a model fine-tuned on both MS MARCO and microblog data, we only fine-tuned on MS MARCO data.

3.3 Results

Document ranking results are shown in Table 2. Note that NIST judgments were provided on a four-point scale: (3) perfectly relevant, (2) highly relevant, (1) relevant, and (0) irrelevant. The scale was defined in a slightly different way from the passage ranking task, and thus the metrics were computed differently as well. For the purposes of computing nDCG, all grades were used, and for computing AP, grade (1) relevant judgments were also considered relevant (unlike in the passage case). However, for fair comparison between the "full ranking" and "reranking" conditions, all submitted runs were truncated to 100 hits per query.[2] For reference, we show results from the best submitted run in terms of nDCG@10 (idst_bert_v3). On a per-team basis in terms of nDCG@10, we ranked number two among all participants. There appears to be a big gap between runs from the IDST group and our group. However, in terms of AP, bm25_marcomb was the best submitted run in the evaluation. Since nDCG@10 is an early precision metric, this rank swap is perhaps not surprising.

Looking at the baselines: as with the passage retrieval runs, tuned BM25 performed *worse* than BM25 with default parameters. However, with query expansion, tuning generally improved both metrics, although AP for RM3 appears to be an outlier.

Our BERT-based runs reveal two interesting findings: First, runs bm25_marcomb and bm25exp_marcomb form a contrastive pair, and shows that document expansion (at least with our implementation) does *not* appear to improve results (in fact, AP degrades slightly). Second, runs bm25exp_marco and bm25exp_marcomb form another contrastive pair, and shows that (additionally) fine-tuning the BERT relevance classifier on microblog data improves effectiveness. This confirms our previous finding [2] that BERT is able to effectively perform *cross-domain* relevance transfer.

[2]In trec_eval, this is specified using the -M 100 option.

ACKNOWLEDGMENTS

This research was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada, with computational resources provided by Compute Ontario and Compute Canada.

REFERENCES

[1] 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): System Demonstrations*. Hong Kong, China, 19–24.

[2] 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)*. Hong Kong, China, 3490–3496.

[3] Nima Asadi and Jimmy Lin. 2013. Effectiveness/Efficiency Tradeoffs for Candidate Generation in Multi-Stage Retrieval Architectures. In *Proceedings of the 36th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2013)*. Dublin, Ireland, 997–1000.

[4] 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)*. Minneapolis, Minnesota, 4171–4186.

[5] Jimmy Lin. 2018. The Neural Hype and Comparisons Against Weak Baselines. *SIGIR Forum* 52, 2 (2018), 40–51.

[6] Jimmy Lin, Miles Efron, Yulu Wang, and Garrick Sherman. 2014. Overview of the TREC-2014 Microblog Track. In *Proceedings of the Twenty-Third Text REtrieval Conference (TREC 2014)*. Gaithersburg, Maryland.

[7] Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. In *arXiv:1901.04085*.

[8] Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. 2019. Document Expansion by Query Prediction. In *arXiv:1904.08375*.

[9] 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)*. Tokyo, Japan, 1253–1256.

[10] Peilin Yang, Hui Fang, and Jimmy Lin. 2018. Anserini: Reproducible Ranking Baselines Using Lucene. *Journal of Data and Information Quality* 10, 4 (2018), Article 16.

[11] Peilin Yang and Jimmy Lin. 2019. Reproducing and Generalizing Semantic Term Matching in Axiomatic Information Retrieval. In *Proceedings of the 41th European Conference on Information Retrieval, Part I*. Cologne, Germany, 369–381.

[12] Wei Yang, Haotian Zhang, and Jimmy Lin. 2019. Simple Applications of BERT for Ad Hoc Document Retrieval. In *arXiv:1903.10972*.

[13] Zhaohao Zeng and Tetsuya Sakai. 2019. BM25 Pseudo Relevance Feedback Using Anserini at Waseda University. In *Proceedings of the Open-Source IR Replicability Challenge (OSIRRC 2019): CEUR Workshop Proceedings Vol-2409*. Paris, France, 62–63.