IELAB at TREC Deep Learning Track 2021

Shengyao Zhuang
s.zhuang@uq.edu.au
The University of Queensland
Brisbane, QLD, Australia

Hang Li
hang.li@uq.edu.au
The University of Queensland
Brisbane, QLD, Australia

Shuai Wang
shuai.wang2@uq.edu.au
The University of Queensland
Brisbane, QLD, Australia

Guido Zuccon
g.zuccon@uq.edu.au
The University of Queensland
Brisbane, QLD, Australia

ABSTRACT

We describe the methods used by the IELAB in the TREC Deep Learning Track (TREC DL) 2021. The IELAB investigated three BERT-based ranking models to improve both the retrieval and the re-ranking stages of the passage ranking system. As for the passage retrieval runs, the methods used were: a novel learned sparse index method called uniCOIL, a dense retriever method called ADORE, and the combination of the two (hybrid). For the passage re-ranking run, TILDEv2, a fast yet effective passage re-ranking method, was used. Both uniCOIL and TILDEv2 rely on passage expansion. Common practice is to use the docTquery-T5 for passage expansion – however this method does not scale well. In fact, performing the expansion for the newly released MS MARCOv2 passage collection, which is 15.6 times larger than the old v1 collection, was not possible within the timeframe of the TREC DL 2021 task. To address this issue, we adapt the TILDE model to serve as the passage expansion method: compared to docTquery-T5, TILDE reduces passage expansion time by 98%.

KEYWORDS

BERT-based ranker, Learned sparse index, fast passage expansion

ACM Reference Format:
Shengyao Zhuang, Hang Li, Shuai Wang, and Guido Zuccon. 2020. IELAB at TREC Deep Learning Track 2021. In *WWW '21: ACM International Conference on The Web Conference, April 19–23, 2021, Ljubljana, Slovenia.* ACM, New York, NY, USA, 3 pages. https://doi.org/XXXX/XXXX

1 INTRODUCTION

The methods we use in our TREC DL 2021 submissions leverage the contextualized representations produced by deep LMs, such as BERT [1] to estimate the semantic matching score between documents and queries. However, one of the limitations of using BERT to perform matching is that it is a computationally expensive operation, thus producing high query latency. This problem is further exacerbated for this year's TREC DL since the collection is much larger than the previous years. Hence, for TREC DL 2021 the IELAB focused on devising effective and efficient BERT-based retrieval and re-rank methods; specifically we used the learned sparse retrieval method uniCOIL [2], the dense retrieval method ADORE [8] and the fast passage re-ranking method TILDEv2.

2 METHODOLOGY

2.1 Models

Our submitted runs are based on the following BERT-based ranking models:

- **TILDEv2** [9] is a fast BERT-based re-ranking method; it uses BERT to precompute passage token weights at indexing time and uses the BERT tokenizer to process the query at query time. The relevance matching scores are computed based on exact term matching. It uses relevant judgments as positive training samples and randomly picks negatives from BM25 top 1,000.
- **uniCOIL** [2] is a BERT-based retrieval method that precomputes token scores in each passage and stores them into an inverted index. At query time, it requires a single BERT inference to get token scores in the query. The relevance matching scores are computed based on exact term matching. It uses relevant judgments as positive training samples and randomly picks negatives from BM25 top 1,000.
- **ADORE** [8] is a BERT-based dense retriever, which has been trained with advanced hard negative sampling training strategies. The relevance matching scores are computed based on the dense vector similarity search.

Instead of directly using the above models to generate run files, we follow the recent works that have shown that the simple interpolation of BERT scores with BM25 scores [6] or the hybrid dense retriever results with sparse retriever results [2, 3], can improve effectiveness over using the BERT-based retriever alone. Thus, we applied interpolation or hybrid on top of the BERT-based retrieval models to produce our submitted runs.

We directly use the model checkpoints provided by the original authors. Note that these are trained on MS MARCOv1 training data: this means we are testing the effectiveness of these models on the MS MARCOv2 dataset in a "zero-shot" way.

WWW '21, April 19–23, 2021, Ljubljana, Slovenia

ACM ISBN 978-1-4503-XXXX-X/20/XX…$15.00
https://doi.org/XXXX/XXXX

2.2 Passage expansion

Both TILDEv2 and uniCOIL are exact term matching methods. Thus, if no other matching mechanism is implemented, their effectiveness is limited by the vocabulary mismatch problem. In order to reduce the impact of this problem, these two methods require passages to be expanded in advance. Passage expansion appends semantically related and potentially relevant terms at the end of a passage, in a bid to increase the likelihood of retrieving the passage for queries containing those expanded terms and for which the passage is relevant.

One of the most popular passage expansion methods is docTquery-T5 [4]. docTquery-T5 is a T5-based [5] sequence-to-sequence generative language model, which can only generate one token at a time. Thus, multiple inferences from docTquery-T5 are needed to obtain several tokens for passage expansion. Provided that T5 is a large transformer model, passage expansion with docTquery-T5 requires large computational resources. According to the statistics provided by the docTquery-T5 authors [4], sampling 40 queries per passage for each of the 8.8 million passages in the MS MARCOv1 collection requires 320 hours on a single TPU. Thus, ≈5,000 hours are required for expanding the MS MARCO v2's 138.3 million passages, which is infeasible for our participation in the TREC DL 2021 task. Thus, following Zhuang and Zuccon [9], we adapt TILDE [10] to perform passage expansion. TILDE is trained with a term-independent query likelihood loss; thus, it can predict important query terms for a passage with only a single step of BERT inference. As a result, we can expand the whole MS MARCOv2 passage collection in around 20 hours with 6 Tesla v100 16G GPUs.

2.3 Submitted runs

Passage ranking task runs:

- TILDEv2: This is a two-stage run. First, BM25 retrieves the top 1000 passages, and re-rank is done with the TILDEv2 model.
- BM25-uniCOIL: BM25 interpolated with uniCOIL. This is a single-stage retrieval run in which we interpolate BM25 top 1,000 passage scores with uniCOIL top 1,000 scores. Scores are normalised before interpolation.
- ADORE-uniCOIL: ADORE interpolated with uniCOIL. This is a single-stage retrieval run in which we interpolate ADORE top 1,000 passage scores with uniCOIL top 1,000 scores. Scores are normalised before interpolation.

3 RESULTS

The evaluation results obtained by runs are reported in Table 1. All our submitted runs have significantly higher effectiveness than the baseline BM25 across all the official evaluation metrics. ADORE-uniCOIL achieves the highest effectiveness, highlighting the benefit of hybrid advanced learned sparse retrieval with dense retrieval. Interestingly, the run BM25-uniCOIL which interpolates BM25 and uniCOIL – hybird of two sparse retrieval methods, also achieves strong effectiveness. We note BM25-uniCOIL cannot solve the vocabulary mismatch problem because all retrieved passages need to contain at least one query term. On the other hand, ADORE-uniCOIL has the ability to solve the vocabulary mismatch problem as it exploits the ADORE component to perform dense vector matching.

Table 1: The results for the baseline BM25 run and the results from all our submitted passage runs. The best results are marked with bold.

Model	MAP	RR	nDCG@10	nDCG@1000
BM25	0.1357	0.5060	0.4458	0.3497
TILDEv2	0.2112	0.6926	0.5825	0.4256
BM25-uniCOIL	0.2745	0.7975	0.6420	0.4890
ADORE-uniCOIL	**0.2842**	**0.8045**	**0.6714**	**0.4944**

Finally, the re-ranking method TILDEv2 achieves the lowest effectiveness among our three submitted runs. However, it enjoys low query latency and no GPU is needed. At query time, TILDEv2 uses a BERT tokenizer to process each query which takes only 0.1 ms per test query on CPU. This is followed by searching in our custom index (implemented using Hashtable) to compute the final scores and re-rank the 1,000 passages retrieved by BM25, which takes around 11 ms per query on CPU. Thus, the overall query latency for TILDEv2 is given by the BM25 retrieval time (around 45 ms [1]) plus the re-ranking time (11 ms). Compared to the uniCOIL's retrieval which takes around 230 ms per query on CPU and ADORE's retrieval which takes around 85 ms per query on GPU (very high latency on CPU), TILDEv2 achieves the lowest query latency with limited computational resources.

4 CONCLUSION

The IELAB's submissions for TREC Deep Learning Track 2021 focused on three efficient BERT-based retrieval and re-ranking methods. The methods were chosen by taking into account the high query latency challenge posed by BERT-based ranking methods on the MS MARCOv2 dataset. In addition, we replaced the expensive docTquery-T5 passage expansion method with TILDE to efficiently expand the passage collection used in the task. Our empirical results show that the submitted methods achieve both high effectiveness and efficiency.

REFERENCES

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT (1)*.
[2] Jimmy Lin and Xueguang Ma. 2021. A Few Brief Notes on DeepImpact, COIL, and a Conceptual Framework for Information Retrieval Techniques. *arXiv preprint arXiv:2106.14807* (2021).
[3] Sheng-Chieh Lin, Jheng-Hong Yang, and Jimmy Lin. 2021. In-batch negatives for knowledge distillation with tightly-coupled teachers for dense retrieval. In *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*. 163–173.
[4] Rodrigo Nogueira and Jimmy Lin. 2019. From doc2query to docTTTTTquery. *Online preprint* (2019).
[5] 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 (2020), 1–67.

[1] We use Anserini implementation of BM25 [7]

[6] Shuai Wang, Shengyao Zhuang, and Guido Zuccon. 2021. BERT-based Dense Retrievers Require Interpolation with BM25 for Effective Passage Retrieval. In *Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval*. 317–324.

[7] Peilin Yang, Hui Fang, and Jimmy Lin. 2017. Anserini: Enabling the use of lucene for information retrieval research. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1253–1256.

[8] Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2021. Optimizing Dense Retrieval Model Training with Hard Negatives. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

[9] Shengyao Zhuang and Guido Zuccon. 2021. Fast passage re-ranking with contextualized exact term matching and efficient passage expansion. *arXiv preprint arXiv:2108.08513* (2021).

[10] Shengyao Zhuang and Guido Zuccon. 2021. TILDE: Term Independent Likelihood MoDEl for Passage Re-Ranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21)*. 1483–1492. https://doi.org/10.1145/3404835.3462922