L3S at the TREC 2021 Deep Learning Track

Jurek Leonhardt[1], Koustav Rudra[2⋆], and Avishek Anand[1]

[1] L3S Research Center, Hannover, Germany
{leonhardt,anand}@L3S.de
[2] Indian Institute of Technology (ISM) Dhanbad, India
koustav@iitism.ac.in

Abstract. In this paper we describe the approach we used for the passage and document ranking task in the TREC 2021 deep learning track. Our approach aims for efficient retrieval and re-ranking by making use of fast look-up-based forward indexes for dense dual-encoder models. The score of a query-document pair is computed as a linear interpolation of the corresponding lexical (BM25) and semantic (re-ranking) scores. This is akin to performing the re-ranking step "implicitly" together with the retrieval step. We improve efficiency by avoiding forward passes of expensive re-ranking models without compromising performance.

Keywords: retrieval · dense · sparse · hybrid · ranking · interpolation

1 Introduction

The previous TREC deep learning tracks have been dominated by transformer-based models [2, 3]. Specifically, contextual cross-attention models based on BERT [5] are popular choices for re-ranking in both the passage- and document-level tasks.

Recently, contextual models have also been applied in the retrieval step [8, 13, 11]. These dense retrievers compute independent, low-dimensional query and document representation vectors and rank documents by computing the similarity of their representations to the query. Because of the independent query and document encoders, these approaches are also referred to as *two-tower* or *dual-encoder* models. Dense retrieval is typically performed using (approximated) nearest neighbor search [7] or maximum inner product search (MIPS). However, this is more resource intensive (requiring GPU acceleration) and generally slower than sparse retrieval.

Dense retrievers are somewhat complementary to sparse models (such as BM25); due to their pre-training, the learnt representations allow for *semantic matching*, which often helps capturing related documents otherwise missed. However, as they do not perform any direct term matching, the recall tends to suffer [9]. For that reason, *hybrid* retrieval approaches have been studied [6], where documents retrieved by a sparse and dense retriever, respectively, are

⋆ Research was primarily conducted while affiliated to L3S Research Center.

combined. This takes advantage of both sparse and dense retrieval, but is still bottle-necked by the dense retrieval step.

Our approach goes one step further and obviates the dense retrieval step altogether. Instead, we perform sparse retrieval and simply compute the corresponding dense score for each retrieved document. The two scores are then interpolated linearly to obtain the final document score. We refer to this process as *score completion*. As we use dual-encoder models, all document representations can be pre-computed and stored in an efficient hash map. Consequently, our approach consists of an inexpensive sparse retrieval step and a series of look-ups and only requires a single forward pass to encode the query.

Our experiments show that interpolation of sparse and dense scores improves retrieval performance. Further, using our look-up technique, we are able gain speed improvements compared to MIPS-based retrieval.

2 Approach

In this section we briefly introduce our look-up-based hybrid retrieval approach.

2.1 Interpolation-based Re-ranking

The retrieval pipeline for a query q consists of two stages. In the first stage, a set of documents, K_S^q, is retrieved using a (typically sparse, term-based) retrieval model. Afterwards, for each document $d \in K_S^q$, another (typically dense or semantic) model is employed to compute a corresponding re-ranking score w.r.t. q. Let the sparse and dense scores be denoted by $\phi_S(q, d)$ and $\phi_D(q, d)$, respectively. Linear interpolation of both scores to obtain the final ranking has been shown to improve overall performance [1]:

$$\phi(q, d) = \alpha \cdot \phi_S(q, d) + (1 - \alpha) \cdot \phi_D(q, d) \tag{1}$$

We denote the number of documents retrieved from the sparse index as $k_S = |K_S^q|$. Note that dual-encoder-based dense models, such as ANCE [13] or TCT-ColBERT [11], often split documents into passages (*maxP* [4]). The score of a document is then computed as follows:

$$\phi_D(q, d) = \max_{p_i \in d} \phi_D(q, p_i) \tag{2}$$

2.2 Forward Indexes

The computation of the (dense) re-ranking scores is usually very computationally expensive. In this work, we focus on dual-encoders or two-tower models. These models compute the relevance score for a query-document pair as

$$\phi_D(q, d) = \zeta(q) \cdot \eta(d), \tag{3}$$

where ζ and η are the query and document encoders, each outputting a representation vector. As the two encoders are independent, the document representations $\eta(d)$ can be pre-computed in advance for each document d in the corpus.

This allows for storing the document representations in look-up based hash map, which can be accessed in constant time. After the index is created, the score of a query-document pair can be computed as

$$\phi_D^L(q, d) = \zeta(q) \cdot \eta^L(d), \tag{4}$$

where the superscript L indicates a pre-computed document representation that is looked up. For maxP indexes the score is computed as follows:

$$\phi_D^L(q, d) = \max_{p_i \in d} \left(\zeta(q) \cdot \eta^L(p_i)\right) \tag{5}$$

Thus, for each query q, only a single forward pass of the encoder model, $\zeta(q)$, is required. This is in contrast to traditional (interpolation-based) re-ranking, e.g. using a BERT model [12], which requires k expensive forward passes to re-rank the top-k retrieved documents.

3 Experiments

In this section we present our results on the passage and document ranking tasks.

3.1 Setup

We use the Pyserini [10] for our retrieval experiments. The per-query latency is measured as the sum of scoring, interpolation and sorting cost (pre-processing and tokenization is ignored) using an Intel Xeon Silver 4210 CPU with 40 cores and 256GB of RAM. The dense model (TCT-ColBERT) uses a batch size of 256 and outputs 768-dimensional representations. We use BM25 as the sparse retriever for all experiments.

We have three runs each for the passage and document ranking task, varying the interpolation parameter α (cf. Eq. (1)). The dense models we use are available on the HuggingFace Hub. We use `castorini/tct_colbert-msmarco` for the 2019 test set and `castorini/tct_colbert-v2-msmarco` for the 2021 test set.[3] For the TCT-ColBERT baseline, we use dense indexes provided by Pyserini as `msmarco-doc-tct_colbert-bf` and `msmarco-passage-tct_colbert-bf`. As the deep learning 2021 track uses the `MS MARCO v2` corpus, we exploit the new passage-document mapping to use the passage corpus for both the passage and document ranking task, i.e. the document scores are computed using the maxP approach (cf. Eq. (2)).

Table 1. Passage retrieval performance. Latency is reported per query on the 2019 test set. For the interpolation-based retrievers, we set the sparse retrieval depth to $k_S = 5000$.

			2019		2021	
	time [ms]	α	MAP@100	MRR@100	MAP@100	MRR@100
BM25	-	-	0.248	0.704	0.136	0.506
TCT-ColBERT	307	-	0.348	0.823	-	-
BM25 + TCT-ColBERT	114	0.2	0.400	0.902	0.200	0.649
		0.3	0.389	0.886	0.193	0.640
		0.5	0.338	0.813	0.173	0.603

Table 2. Document retrieval performance (maxP). Latency is reported per query on the 2019 test set. For the interpolation-based retrievers, we set the sparse retrieval depth to $k_S = 5000$.

			2019		2021	
	time [ms]	α	MAP@100	nDCG@20	MAP@100	nDCG@20
BM25	-	-	0.244	0.483	0.214	0.498
TCT-ColBERT	582	-	0.225	0.570	-	-
BM25 + TCT-ColBERT	253	0.2	0.311	0.653	0.277	0.608
		0.5	0.301	0.609	0.273	0.603
		0.7	0.281	0.572	0.253	0.571

3.2 Results

We illustrate our results on the passage and document retrieval task for the 2019 and 2021 tracks in Tables 1 and 2, respectively. We further compare the results to a sparse (BM25) and dense (TCT-ColBERT) retrieval baseline.

In most cases, the dense retriever performs better than the sparse retriever, however, the lack of term matching shows in the MAP score for document retrieval. Further, the per-query latency is much higher due to the nearest-neighbor search that is required.

On the other hand, interpolation-based retrieval beats both sparse and dense retrievers consistently, considerably boosting the performance in all cases. Due to the look-up technique, the latency is also much lower than for dense retrieval.

In general, it is important to note that the dense retriever (TCT-ColBERT) is a zero-shot model, as it was trained on an old version of the corpus. Thus, we expect that the performance is degraded to some extent because of this.

[3] Note that these models are trained on the `MS MARCO v1` corpus.

4 Conclusion

We have shown that the interpolation of lexical (sparse) and semantic (dense) scores boosts ranking performance. Further, complementing the scores of a sparse retriever using a dense dual-encoder model allows for fast and efficient "implicit" re-ranking during the retrieval stage and does not require GPU acceleration.

Acknowledgements Funding for this work was in part provided by EU Horizon 2020 grant no. 871042 (*SoBigData++*) and 832921 (*MIRROR*).

References

1. Akkalyoncu Yilmaz, Z., Yang, W., Zhang, H., Lin, J.: 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). pp. 3481–3487 (Nov 2019)
2. Craswell, N., Mitra, B., Yilmaz, E., Campos, D.: Overview of the trec 2020 deep learning track (2021)
3. Craswell, N., Mitra, B., Yilmaz, E., Campos, D., Voorhees, E.M.: Overview of the trec 2019 deep learning track (2020)
4. Dai, Z., Callan, J.: Deeper text understanding for ir with contextual neural language modeling. In: ACM SIGIR'19. pp. 985–988 (2019)
5. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: 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). pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019). https://doi.org/10.18653/v1/N19-1423, https://aclanthology.org/N19-1423
6. Gao, L., Dai, Z., Chen, T., Fan, Z., Van Durme, B., Callan, J.: Complement lexical retrieval model with semantic residual embeddings. In: Hiemstra, D., Moens, M.F., Mothe, J., Perego, R., Potthast, M., Sebastiani, F. (eds.) Advances in Information Retrieval. pp. 146–160. Springer International Publishing, Cham (2021)
7. Johnson, J., Douze, M., Jégou, H.: Billion-scale similarity search with gpus. IEEE Transactions on Big Data (2019)
8. Karpukhin, V., Oğuz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., tau Yih, W.: Dense passage retrieval for open-domain question answering (2020)
9. Leonhardt, J., Rudra, K., Khosla, M., Anand, A., Anand, A.: Fast forward indexes for efficient document ranking (2021)
10. Lin, J., Ma, X., Lin, S.C., Yang, J.H., Pradeep, R., Nogueira, R.: Pyserini: A Python Toolkit for Reproducible Information Retrieval Research with Sparse and Dense Representations, p. 2356–2362. Association for Computing Machinery, New York, NY, USA (2021), https://doi.org/10.1145/3404835.3463238
11. Lin, S.C., Yang, J.H., Lin, J.: 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). pp. 163–173. Association for Computational Linguistics, Online (Aug 2021). https://doi.org/10.18653/v1/2021.repl4nlp-1.17, https://aclanthology.org/2021.repl4nlp-1.17
12. Nogueira, R., Cho, K.: Passage re-ranking with BERT. CoRR **abs/1901.04085** (2019), http://arxiv.org/abs/1901.04085
13. Xiong, L., Xiong, C., Li, Y., Tang, K.F., Liu, J., Bennett, P.N., Ahmed, J., Overwijk, A.: Approximate nearest neighbor negative contrastive learning for dense text retrieval. In: International Conference on Learning Representations (2021), https://openreview.net/forum?id=zeFrfgyZln