BIT.UA@TREC 2020 Deep Learning Track

Tiago Almeida[0000−0002−4258−3350] and Sérgio Matos[0000−0003−1941−3983]

DETI/IEETA, University of Aveiro, 3810-193 Aveiro, Portugal
{tiagomeloalmeida,aleixomatos}@ua.pt

Abstract. We describe a two-stage retrieval pipeline for the TREC Deep Learning 2020 track, where we used a lightweight neural model to rerank a baseline produced by an efficient traditional technique. In terms of overall performance, our results are slightly below the median, with a best score of 0.5283 nDCG@10. Our source code is available from https://github.com/T-Almeida/TREC-DL-2020.

Keywords: Neural Networks · Information Retrieval · Lightweight Neural Model.

1 Introduction

This work describes the participation, for the first time, of the Biomedical Informatics and Technologies (BIT) group from the University of Aveiro, Portugal, in the TREC Deep Learning track. More precisely, we submitted results to the document ranking task, that aimed to retrieve documents from the MSMarco [5] dataset for the given test topics.

Our approach was focused on an in-house lightweight shallow interaction-based neural network, with only 620 trainable parameters in its current configuration, that was used as a reranker in a two-stage pipeline. Our main objective was to gain intuition on the track and evaluate the model behavior on a large scale dataset, as well as comparing it against state-of-the-art models such as transform-based ones.

In the remainder of the paper we describe our methodology for the construction of the submitted runs, present the results that were obtained, and finish with a conclusion section.

2 Methodology

As already hinted, we explored a classic two-stage retrieval pipeline, where the first stage corresponds to a baseline originated from a traditional retrieval model, namely BM25 [7]. In the second stage we adopted our shallow interaction-based model to further score the previously retrieved documents. Through this section, we describe the most important steps that comprise our submissions.

2.1 Data preparation

In terms of data preparation, we kept a simple approach and built a regex based tokenizer and 200 dimension word embeddings for the produced vocabulary. The tokenizer consisted of filtering off non-alphanumeric characters with the exception of the hyphen character since this usually appears as part of compound words, that empirically seems to be important to keep together. We run this tokenizer over the MSMARCO documents and the development and test questions, resulting in a vocabulary with approximately 2 million tokens. Note this is a large number which hints that more attention should be given to this step.

We used the word2vec skip-gram algorithm from the Gensim [6] library to obtain the word embeddings. Specifically, we adopted the default configuration present on the Gensim library for training with word2vec skip-gram.

2.2 Neural reranking model

The adopted neural model was originally proposed in [2] and further improved in [1]. Figure 1 describes the overall architecture, where we employed an interaction network to learn and pool relevant signals and an aggregation network to weigh all the evidence found on different document passages.

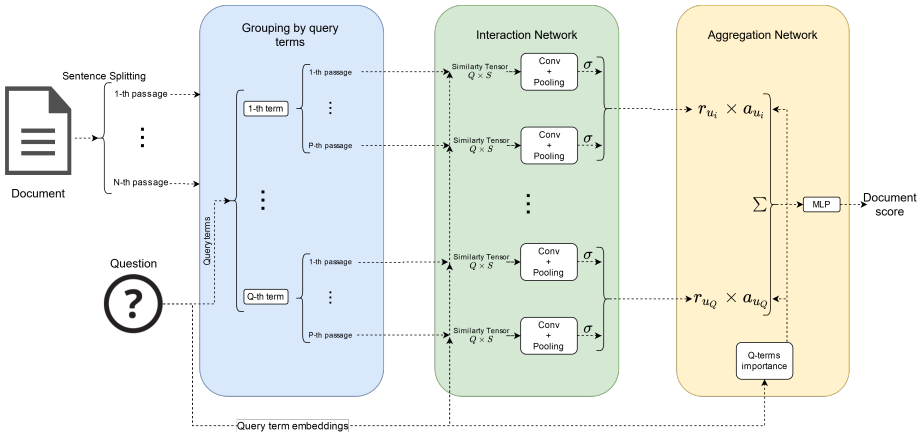

Fig. 1: Lightweight neural model data and operation flow.

In a more detailed way, each document is split into sentences that are further combined with the query to build interaction matrices. Then, taking these matrices in the interaction network, 3-by-3 convolutions are adopted to learn n-gram patterns that are then extracted by pooling operations, lastly, the resulting feature vector is linearly combined with a trainable vector to compute a sentence relevant signal, with 0 for irrelevant and 1 for relevant. Next, the job of the aggregation network is to weigh the importance of each sentence in order

to produce the final document score. For that, we follow the heuristic to first weigh each sentence by the importance of each query term, as suggested in [4], where this importance is learned by taking into consideration the embedding representation of the query term.

More importantly, this model inner-working follows some of the best-reported ideas from shallow interaction-based models resulting in a completely transform-free architecture. As intuition, it was designed to weigh the importance of the document sentences by taking into consideration the context where the exact match with the query terms occurs. In other words, this model produces a more refined judgment of the previously exact match signal considered in the first stage of the pipeline.

2.3 Training and hardware

Regarding the neural model, it was trained using a pairwise cross-entropy loss over the entire training collection. After each training epoch we also store the current model weights and measure the performance in the validation and 2019 test data. The model architecture parameters were the same as used in [1], so we redirect the reader for further information or details.

Additionally, our experiments ran on a machine with 2x Intel(R) Xeon(R) CPU E5-2630 v4 @ 2.20GHz and 128GB of RAM, highlighting that the neural model only ran on the CPU not requiring a GPU.

2.4 Runs identification

The TREC Deep Learning track allowed three submissions that we utilized in the following way:

- **BIT-run1**: We adopted as the first stage the baseline provided by the organizers and applied our neural ranking model to score these documents producing a final ranking order.
- **BIT-run2**: Consisted of an ensemble submission, using the reciprocal rank fusion [3], over four runs similar to the **BIT-run1** using different training checkpoints for the neural ranking model. Furthermore, we chose the checkpoints that maximized some evaluation metrics on the validation or 2019 test data.
- **BIT-run3**: This run also used the previous ensemble strategy. However, it corresponds to a full rank submission because we utilized the BM25 for the first stage retrieval instead of the TREC baseline. Furthermore, we indexed the full MSMARCO dataset using the ElasticSearch and finetuned the BM25 hyperparameters on the TREC 2019 test data.

3 Results and discussion

In this section, we present the results of our runs and compare to the median measures per topic, as summarized in Table 1.

Table 1: Summary of our runs results comparatively to the TREC average of the median.

Submissions	nDCG@10	nDCG@100	Reciprocal Rank	AP
BIT-run1	0.5239	0.5430	0.8389	0.3466
BIT-run2	0.5283	0.5447	0.8611	0.3466
BIT-run3	0.5063	0.5365	0.8296	0.3267
Median	0.5733	0.5859	0.9444	0.3902

In general, our system under-performed comparatively to the median scores. Moreover, the BIT-run2 achieved our best scores confirming the improvement, although only slight in this case, that is usually achieved when a combination of multiple runs is adopted. Our full ranking approach, BIT-run3, was our weakest submission, which indicates that our BM25 baseline does not offer a better starting point compared to the TREC baseline. We speculate that this behavior may be related to overfitting of BM25 to the TREC 2019 data, which is further aggravated by the fact that we retrieved 250 documents per query instead of 100, which means that the reranker has more unrelated documents to score.

As mentioned, our architecture was the same as used in [1], which may not be the best suitable for this challenge, given the high availability of training data and a broader question domain. It would be interesting to test with a larger architecture and see its behavior. Another detail is the low percentage of relevant documents per query in the training data, which may require a different training setup from what we currently follow.

3.1 Per topic analysis

We now present two visualizations to look with more detail at the individual query performance of our submitted runs. In both visualizations, we use a sequential identifier for the topics and show the conversion to the original TREC topic identifier in the Appendix.

In Figure 2 we show the performance in terms of nDCG@10 per each topic for all the submitted runs comparatively to the official TREC median. It is observable that for a great majority of topics our submissions were able to match the official median, which is a bit counter-intuitive when comparing with the results presented in Table 1. Moreover, our submissions only severely underperformed for topics 32 (1116380), 40 (1131069), and 22 (1030303), especially in the last case due to the first stage baseline failing to retrieve any relevant document, which explains the missing values in the figure.

We also compare, in Figure 3, our best run with the official median and best values, in terms of nDCG@10, reinforcing the idea that our system was able to achieve close to median results and in some cases being close to top results.

In Appendix B we also present the same visualization for the other available evaluation metrics, namely nDCG@100 and reciprocal rank.

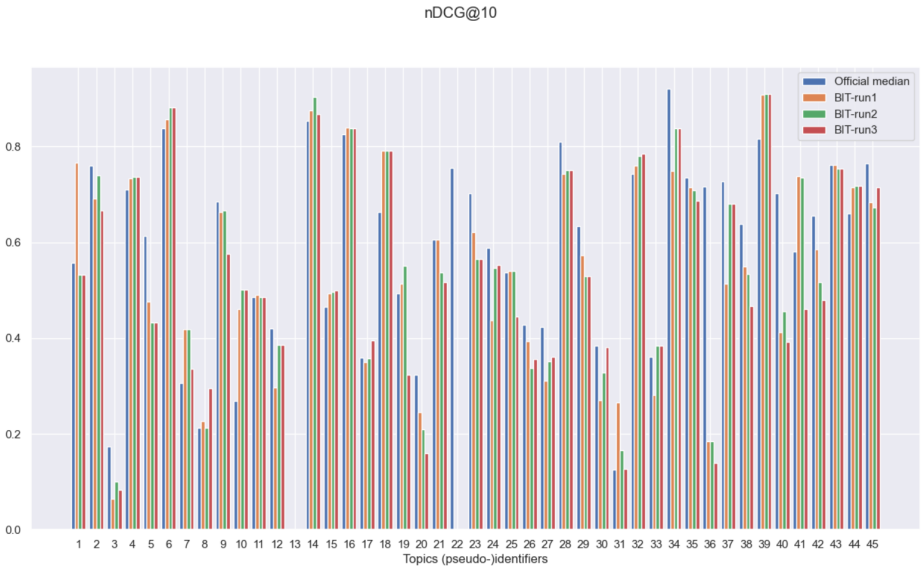

Fig. 2: nDCG@10 of all the submitted runs in comparison to the official median.

4 Conclusion

Despite the relatively weaker results, we gained fundamental insights on the model behavior, given this large training regime, making it an useful effort and an important stepping stone for future enhancements. We believe that more attention can be given to improve the quality of the first stage retrieval while also correcting and finetuning the neural model for this larger training regime.

Acknowledgments

This work has received support from the EU/EFPIA Innovative Medicines Initiative 2 Joint Undertaking under grant agreement No 806968 and from National Funds through the FCT - Foundation for Science and Technology, in the context of the project UIDB/00127/2020.

References

1. Almeida, T., Matos, S.: BIT.UA at bioasq 8: Lightweight neural document ranking with zero-shot snippet retrieval. In: Cappellato, L., Eickhoff, C., Ferro, N., Névéol, A. (eds.) Working Notes of CLEF 2020 - Conference and Labs of the Evaluation Forum, Thessaloniki, Greece, September 22-25, 2020. CEUR Workshop Proceedings, vol. 2696. CEUR-WS.org (2020), http://ceur-ws.org/Vol-2696/paper_161.pdf
2. Almeida, T., Matos, S.: Calling attention to passages for biomedical question answering. In: Jose, J.M., Yilmaz, E., Magalhães, J., Castells, P., Ferro, N., Silva,

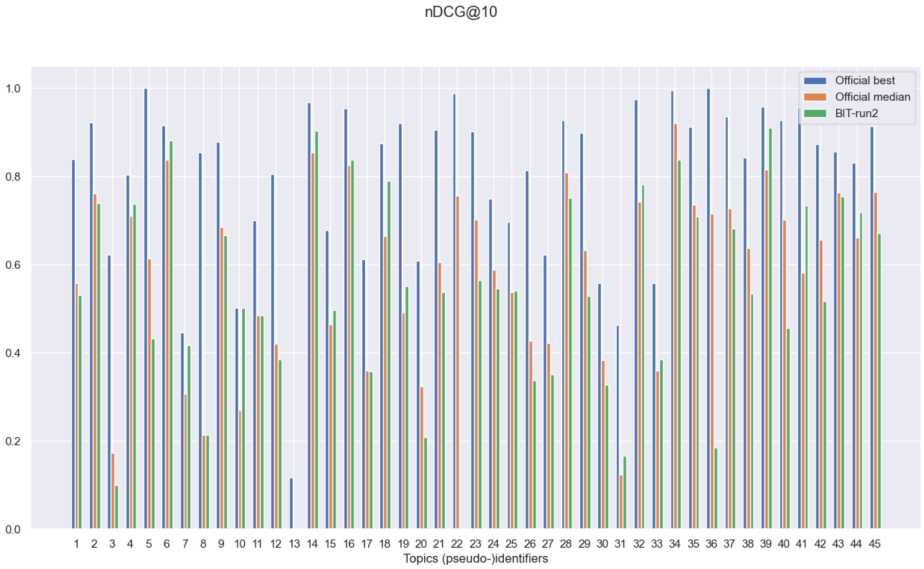

Fig. 3: nDCG@10 of our best run against the official median and best values for each topic.

M.J., Martins, F. (eds.) Advances in Information Retrieval. pp. 69–77. Springer International Publishing, Cham (2020). https://doi.org/10.1007/978-3-030-45442-5_9
3. Cormack, G.V., Clarke, C.L.A., Buettcher, S.: Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In: Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval. p. 758–759. SIGIR '09, Association for Computing Machinery, New York, NY, USA (2009). https://doi.org/10.1145/1571941.1572114, https://doi.org/10.1145/1571941.1572114
4. Guo, J., Fan, Y., Ai, Q., Croft, W.B.: A deep relevance matching model for ad-hoc retrieval. Proceedings of the 25th ACM International on Conference on Information and Knowledge Management (Oct 2016). https://doi.org/10.1145/2983323.2983769
5. Nguyen, T., Rosenberg, M., Song, X., Gao, J., Tiwary, S., Majumder, R., Deng, L.: Ms marco: A human-generated machine reading comprehension dataset (2016)
6. Řehůřek, R., Sojka, P.: Software Framework for Topic Modelling with Large Corpora. In: Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks. pp. 45–50. ELRA, Valletta, Malta (May 2010), http://is.muni.cz/publication/884893/en
7. Robertson, S., Zaragoza, H.: The probabilistic relevance framework: Bm25 and beyond. Found. Trends Inf. Retr. **3**(4), 333–389 (Apr 2009). https://doi.org/10.1561/1500000019

A Topic identifiers

Table A1 shows the mapping between our sequential identifiers and the original TREC topic identifiers to facilitate analysing the per topic visualization.

Table A1: Translation table between sequential indetifiers and the TREC topic identifiers.

TREC topic identifier	**pseudo-identifier**	**TREC topic identifier**	**pseudo-identifier**
42255	1	1043135	24
47210	2	1049519	25
67316	3	1051399	26
135802	4	1056416	27
156498	5	1064670	28
169208	6	1071750	29
174463	7	1103153	30
258062	8	1105792	31
324585	9	1108729	32
330975	10	1109707	33
332593	11	1113256	34
336901	12	1115210	35
673670	13	1116380	36
701453	14	1119543	37
730539	15	1122767	38
768208	16	1127540	39
877809	17	1131069	40
911232	18	1132532	41
938400	19	1136043	42
940547	20	1136047	43
997622	21	1136769	44
1030303	22	1136962	45
1037496	23		

B Remaining visualisation

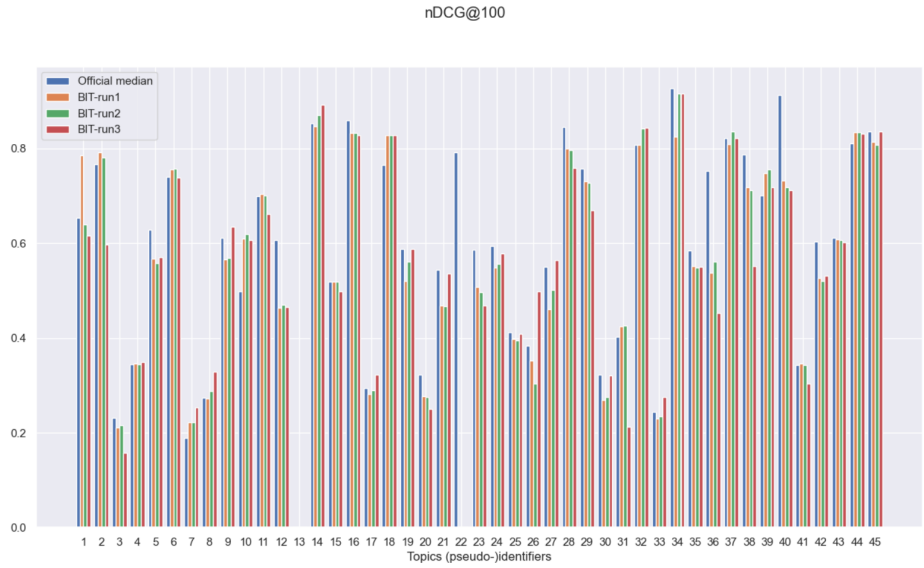

Fig. B1: nDCG@100 of all the submitted runs comparable to the official median.

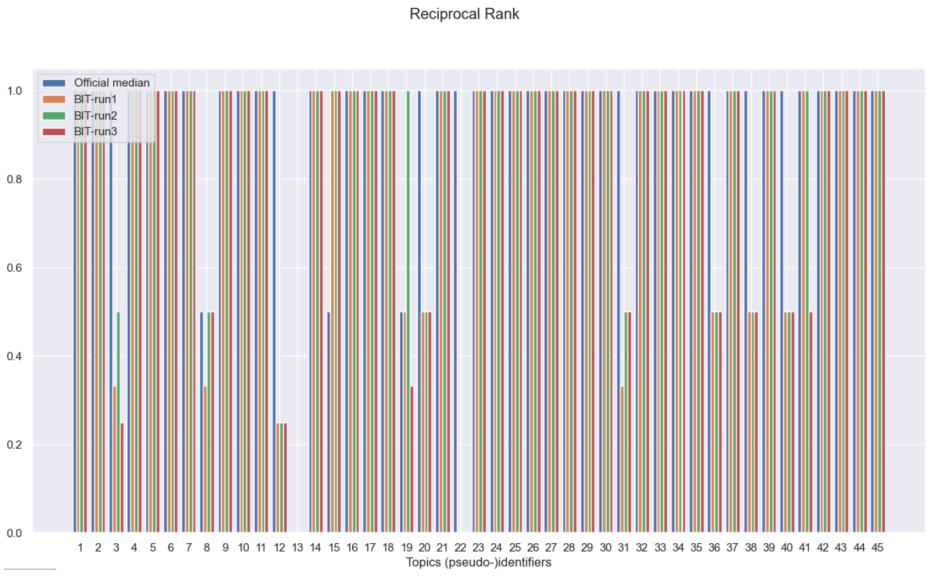

Fig. B2: Reciprocal rank of all the submitted runs comparable to the official median.

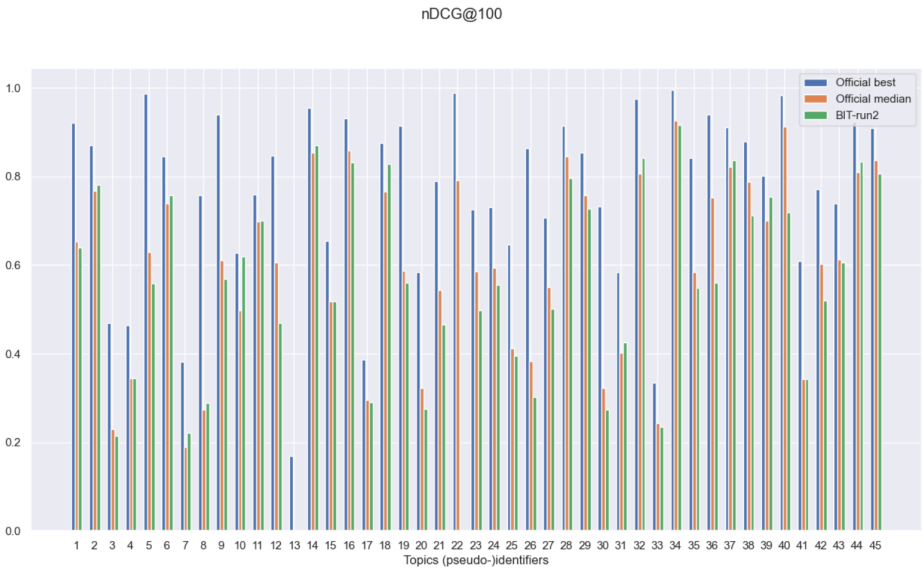

Fig. B3: nDCG@100 of our best run against the official median and best values for each topic.

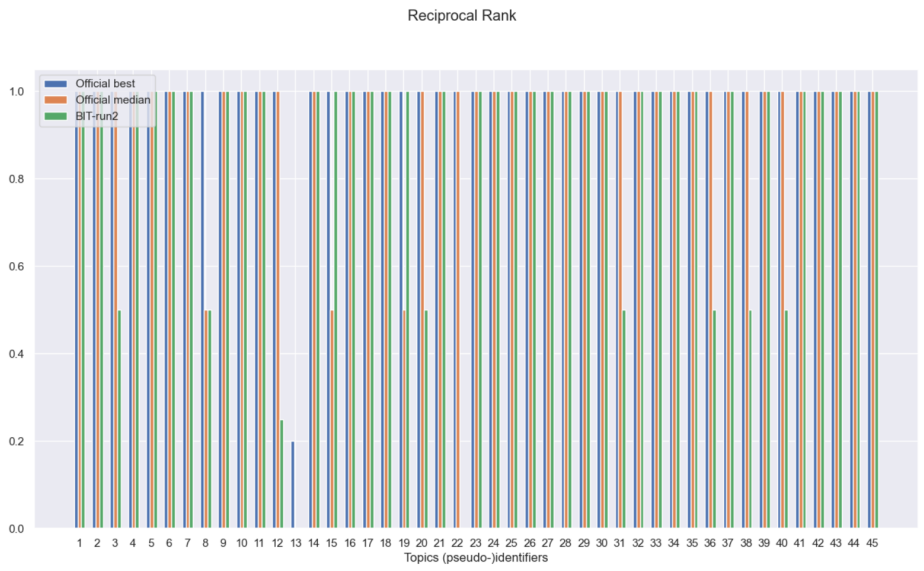

Fig. B4: Reciprocal rank of our best run against the official median and best values for each topic.