Université Grenoble Alpes / LIG at TREC Deep Learning Track 2022

Petra Galuščáková, Lucas Alberede, Gabriela Nicole Gonzalez Saez, Aidan Mannion, Philippe Mulhem, Georges Quénot

Univ. Grenoble Alpes, CNRS, Inria, Grenoble INP[**], LIG, 38000 Grenoble, France

Abstract. This paper describes the submissions of the MRIM research Group/Laboratoire d'Informatique de Grenoble from the Université Grenoble Alpes at the TREC Deep Learning Track 2022. We participated in both fullranking and reranking sub-tasks in the passages ranking task. We proposed several ways to combine neural and non-neural runs using reciprocal-rank based fusion.

1 Introduction

This report describes the submissions of the Modeling and Retrieval of Multimedia Information research Group at Université Grenoble Alpes team at the TREC Deep Learning Track 2022. We have participated in the passages ranking task, in both fullranking and reranking sub-tasks.

For the **fullranking sub-task**, we used several ranking models for the first stage retrieval and an index of the full collection. We subsequently applied reranking on top of the outputs of this ranking. Last, we combined the outputs of multiple rankers and re-rankers into a single output. We comprehensively experimented with a range of first stage retrieval models, rerankers, and architectures and selected 3 runs which we submitted and which were used in the pooling and 4 additional runs, which we submitted but which were not used in the pooling.

In the **reranking sub-task**, we applied three reranking approaches to the provided list of passages and we either combined them or we chained them on the top of each other.

In the submissions selection process, we used the test collection used at TREC Deep Learning 2021 Passage ranking task, which in terms of collection statistics, should well correspond to this year track's collection. This collection contains 477 queries, for 53 out of which there are provided the relevance judgements. As this number is relatively small, in the selection we aimed for both good performance and robustness of the submitted system.

In the following section, we cover the basic blocks which we used in the submissions. We review the fullranking models used for getting the top relevant passages from the full passage collection and the models for reranking which we applied to the passages retrieved by these fullranking models. We then review our fullranking submissions in the Section 3 and the reranking submissions in Section 4. All the scripts used in our submissions are freely available[1].

[**] Institute of Engineering Univ. Grenoble Alpes

2 Basic Building Blocks

We use full ranking models for ranking all the passages available in the collection as the first stage of the retrieval. We experimented with a range of full ranking models from which we selected 5 well performing and diverse models, as such systems might be expected to perform well in a combination. We eventually use following full ranking models in all our submissions:

- **BM25:** The standard BM25 retrieval model with default parameters, using the index and implementation provided by Anserini [11]. In addition to the index of passages, we also use the index of documents, which are further used in the reranking using the Graph Model (see below). If not stated otherwise, we further refer to the retrieval of passages in this report.
- **Expanded BM25:** BM25 with document expansion using docT5query [6] and RM3 blind relevance feedback. We used the index and implementation provided by Anserini.
- **DPH:** DPH [4] is a well performing and a parameter free scoring technique derived from the Divergence from Randomness model. In this approach, we indexed and retrieved the segments using Terrier v5.0 framework [8]. DPH was used with stemming and with blind relevance feedback[2], as this is expected to increase a diversity of the performance of the models.
- **Colbert:** A BERT-based passage retrieval model [2]. Specifically, we used the TCT-ColBERTv2 model fine-tuned on MS MARCO v2 provided by Anserini.
- **uniCOIL:** A model that integrates BERT embeddings into a sparse representation [3]. Specifically, we used Castorini unicoil-msmarco-passage model pre-trained on MS MARCO v2 Passage Corpus, along with on-the-fly docT5query query inference. We used an index provided by Anserini.

Passages returned by these models which were ranked at the top positions are in some cases further reranked. Reranking is done by calculating the similarity between the query and retrieved passage and reordering the passages based on this similarity. Specifically, we use following approaches:

- **BERT:** We used the monoBERT model [7] from Castorini, available in Pygaggle[3], which was finetuned on MS MARCO v1.
- **T5:** We used the monoT5 model [5] from Castorini available in Pygaggle which was finetuned on MS MARCO v1.

---

[1] `https://github.com/galuscakova/trec-dl-2022`

[2] As recommended in `http://terrier.org/docs/current/javadoc/org/terrier/matching/models/DPH.html`

[3] `https://github.com/castorini/pygaggle`

- **MiniLM:** Cross-encoders are transformer models [10] fine-tuned for sentence similarity calculation, which are widely used in semantic search as they compute attention weights across the query and the document together. We used the MiniLM architecture provided by the `sentence-transformers` library[4], which have been pre-trained on the MS MARCO v2 passage reranking dataset.
- **Graph Model:** We use Heterogeneous Graph Attention Networks (HGATs). Document retrieval (BM25) is applied as a first stage retrieval returning the top 1000 documents. Every passage from these documents is then reranked using HGATs to leverage its relations with other passages in a graph-based document representation [1]. Passage and query embeddings are computed using a multiple-representation embedding encoder similar to the one in the Colbert architecture [2].

Finally, for the combination of the ranked and reranked runs, we used reciprocal rank-based (RR) fusion implemented in TrecTools[5].

3 Fullranking Submissions

All our submissions consisted of a combination of fullranking and reranking models. We submitted three runs which were further used in the pooling: hierarchical combination, flat combination and double reranking. A graphical view of the submissions is displayed in Figure 1.

Hierarchical Combination: First, Colbert, UniCOIL and Expanded BM25 rankers were combined and then reranked by BERT [top 10 documents]. Then, this combination was further combined with UniCOIL reranked by dense MiniLM model [top 100 documents] and with BM25 reranked by the Graph Model [top 100 document]. This system achieved the highest performance on our training data in terms of nDCG@10, nDCG@1000 and MAP. The official run name used in the submissions is *hierarchical combination.*

Flat Combination: This combination consists of 6 rankers, rerankers and combinations which all together form the final combination. Due to the diversity of the combined models, this system is also supposed to be robust. Following models are combined:

- UniCOIL
- UniCOIL reranked by a MiniLM model [top 100 documents][6]
- Colbert reranked by T5 [top 100 documents]
- Expanded BM25 reranked by T5 [top 100 documents]

---

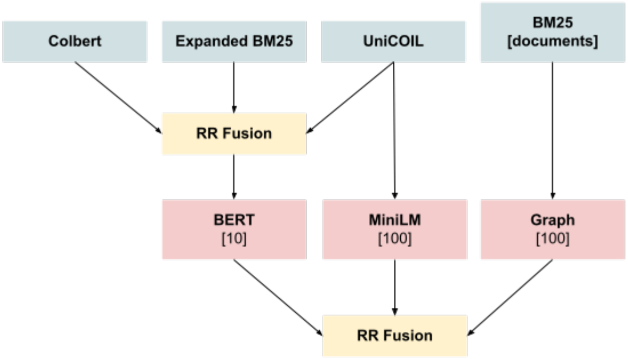
[4] `https://www.sbert.net/docs/pretrained-models/ce-msmarco.html`

[5] `https://github.com/joaopalotti/trectools`

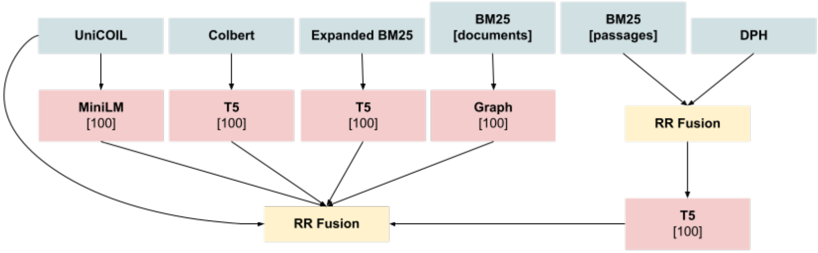
[6] The number of reranked documents is given in the brackets.

- BM25 reranked by the Graph Model [top 100 documents]
- BM25 and DPH first combined, then reranked by T5 [top 100 documents]

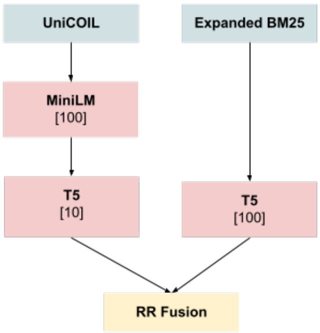
The official run name used in the submissions is *6systems*.


Colbert
Expanded BM25
UniCOIL
BM25 [documents]
RR Fusion
BERT [10]
MiniLM [100]
Graph [100]
RR Fusion

(a) Hierarchical Combination


UniCOIL
Colbert
Expanded BM25
BM25 [documents]
BM25 [passages]
DPH
MiniLM [100]
T5 [100]
T5 [100]
Graph [100]
RR Fusion
RR Fusion
T5 [100]

(b) Flat Combination (6systems)


UniCOIL
Expanded BM25
MiniLM [100]
T5 [10]
T5 [100]
RR Fusion

(c) Double Reranking (2systems)

Fig. 1: Full ranking architectures. Ranking models are in blue, reranking models in rose, and fusions in yellow. The number of reranked passages/documents is in the brackets.

Double Reranking: This submission follows the architecture of duoT5 model [9] with two different rerankers chained. The first stage ranker is UniCOIL model. Top 100 returned passages are first reranked using the MiniLM model. Subsequently, the top 10 reranked documents are further reranked by T5 model. Last, this double reranked model is further combined with the expanded BM25 model reranked by T5 model [top 100 documents]. The official run name used in the submissions is *2systems*.

In addition to these runs, we submitted other four runs which were officially evaluated by the task organizers, but were not included in the evaluation pool.

Graph: This system is simply the Graph Model described in Section 2 which uses a BM25 ranking of the documents followed by reranking done using the Graph Attention Networks. The official run name used in the submissions is *graph colbert*.

Reranked Unicoil: This system also does not use any reciprocal rank-based combination. Instead, it only uses double reranking of the UniCOIL system, first by MiniLM model [top 100 documents] and then by T5 model [top 10 documents]. The system is thus the same as the Double Reranking system which was submitted for the pooling, but without the expanded BM25 model applied. The official run name used in the submissions is *unicoil reranked*.

4 Systems with Graph: This combination contains a subset of 4 systems which are included in the Flat Combination:

- UniCOIL
- UniCOIL reranked by a MiniLM model [top 100 passages]
- Expanded BM25 reranked by T5 [top 100 passages]
- BM25 reranked by the Graph Model [top 100 passages]

The official run name used in the submissions is *4systems*.

4 Systems with DPH: This combination again contains a subset of 4 systems which are included in the Flat Combination. Comparing with 4 Systems with Graph, the BM25 model reranked by the graph model is here replaced by a combination of BM25 and DPH models:

- UniCOIL
- UniCOIL reranked by a MiniLM model [top 100 documents]
- Expanded BM25 reranked by T5 [top 100 documents]
- BM25 and DPH first combined, then reranked by T5 [top 100 documents]

The official run name used in the submissions is *c47*.

3.1 Official Evaluation Results

The evaluation of these submitted systems is displayed in Table 1. Based on this table we see that the single Graph-based reranking and the Hierarchical Combination outperform the other combinations for the measures taking into account top 1000 results (MAP, nDCG@1000 and Recall@1000). The 4-way combinations perform the worst according to these measures. As the Hierarchical Combination also includes the Graph-based reranking as one of the combined systems and as they perform similarly in terms of retrieving the top 1000 documents, we can claim, that the additional systems in the combination add no extra improvement to the Graph system. However, the Hierarchical Combination performs better than the Graph in terms of the measures working only with top 10 documents (nDCG@10 and P@10). The additional systems added to the Graph system thus do not change the performance of the retrieval on the top 1000 documents, but leads to improvement for the top 10 documents retrieval. However, clearly the best performance in terms of retrieving top 10 documents is achieved by the Double Reranking system (in terms of nDCG@10 and P@10 measures), followed by the Reranked Unicoil. This might confirm the usefulness of the duo architecture, especially for the retrieval on the top positions and it should lead to further experimentation with different cascading setups and rerankers.

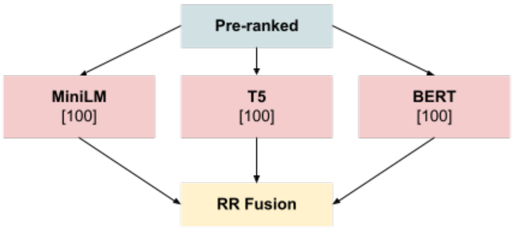
Run name	Pooling	MAP	nDCG@10	nDCG@1000	P@10	Recall@1000
Hierarchical Combination	Yes	**0.166**	0.570	**0.330**	0.492	0.327
Flat Combination	Yes	0.160	0.578	0.312	0.499	0.303
Double Reranking	Yes	0.162	**0.599**	0.302	**0.515**	0.273
Reranked Unicoil	No	0.161	0.591	0.301	0.501	0.273
Graph	No	0.161	0.548	0.326	0.468	**0.329**
4 Systems w. Graph	No	0.153	0.576	0.295	0.497	0.273
4 Systems w. DPH	No	0.149	0.570	0.295	0.479	0.273

Table 1: Full ranking sub-task UGA official results. The highest scores for each measure are highlighted.

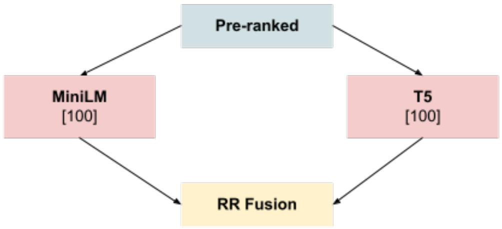
4 Reranking submissions

In the reranking submissions, we only used the reranking methods described above and we either combined them using a fusion or using a chain of double reranking. The graphical overview of all submissions is displayed in Figure 2.

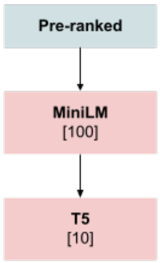
3-way Combination: We reranked the original run by the MiniLM model, T5 model and BERT model. Then we combined all three reranked runs into a single run. The official run name for this submission is *fused_3runs*.


Pre-ranked
MiniLM
[100]
T5
[100]
BERT
[100]
RR Fusion

(a) 3-way Combination (fused_3runs)


Pre-ranked
MiniLM
[100]
T5
[100]
RR Fusion

(b) 2-way Combination (fused_2runs)


Pre-ranked
MiniLM
[100]
T5
[10]

(c) Double Reranking (hierarchical_2runs)

Fig. 2: Reranking architectures.

2-way Combination: We reranked the original run by the MiniLM model and T5 model. Then we combined both reranked runs into a single run. This run might be directly compared with the double reranking run, which uses the same rerankers, but combined together in a different architecture and to the 3-way combination that uses one additional reranker. The official run name for this submission is *fused_2runs.*

Double Reranking: Similarly, as in the full reranking run, we used the duo architectures and stacked two rerankers on a top of each other. We first reranked all 100 passages by the MiniLM model and then reranked again the top 10 passages by the T5 model. The official run name for this submission is *hierarchical_2runs.*

Official evaluation results: The evaluation of our reranking submitted systems is displayed in Table 2. Compared to Table 1, we do not present the Recall@1000 values as reranking has no effect on it and the Recall@1000 for all the systems is 0.138. Clearly, these results are lower than the results for the fullranking task, in terms of all considered measures. This shows the importance of using adequate and varied inputs for the reranking step. What we also notice is that all three runs achive very similar results. The only exception is the hierarchical combination of 3 systems (3-way Combination) which performs better than both combinations of 2 systems (2-way Combination and Double Reranking) in terms of the P@10 score.

Run name	Pooling	MAP	nDCG10	nDCG@1000	P@10
3-way Combination	Yes	**0.090**	**0.509**	0.209	**0.391**
2-way Combination	Yes	**0.090**	0.506	**0.210**	0.374
Double Reranking	Yes	0.089	0.500	**0.210**	0.368

Table 2: Reranking sub-task UGA official results. The highest scores for each measure are highlighted.

5 Conclusion

We described our submissions for the fullranking and reranking sub-tasks of TREC Deep Learning 2022 Passage Retrieval Task. For the fullranking subtask, a graph-based reranking of the passages identified using document ranking performs especially well for retrieving top 1000 documents. Its combination with additional well performing systems then further leads to a strong performance on the top 10 documents. The best performing system for the retrieval of the top 10 documents combines double reranked UniCOIL system with reranked BM25 system with query and document expansions applied. Our experiments thus lead to further questions about how to best combine multiple rankers and rerankers, which we plan to explore in future.

Acknowledgments

This work is supported by the ANR Kodicare bi-lateral project, grant ANR-19-CE23-0029 of the French Agence Nationale de la Recherche, and by the Austrian Science Fund (FWF).

References

1. L. Albarede, P. Mulhem, L. Goeuriot, C. L. Pape-Gardeux, S. Marie, and T. Chardin-Segui. Passage retrieval on structured documents using graph attention networks. In M. Hagen, S. Verberne, C. Macdonald, C. Seifert, K. Balog,

K. Nørvåg, and V. Setty, editors, *Advances in Information Retrieval - 44th European Conference on IR Research, ECIR 2022, Stavanger, Norway, April 10-14, 2022, Proceedings, Part II*, volume 13186 of *Lecture Notes in Computer Science*, pages 13–21. Springer, 2022.

2. O. Khattab and M. Zaharia. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of SIGIR 2020*, SIGIR '20, page 39–48, New York, NY, USA, 2020. Association for Computing Machinery.
3. J. Lin and X. Ma. A Few Brief Notes on DeepImpact, COIL, and a Conceptual Framework for Information Retrieval Techniques, 2021. https://arxiv.org/abs/2106.14807.
4. R. McCreadie, C. Macdonald, I. Ounis, J. Peng, and R. L. T. Santos. University of glasgow at TREC 2009: Experiments with terrier. In E. M. Voorhees and L. P. Buckland, editors, *Proceedings of The Eighteenth Text REtrieval Conference, TREC 2009, Gaithersburg, Maryland, USA, November 17-20, 2009*, volume 500-278 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 2009.
5. R. Nogueira, Z. Jiang, R. Pradeep, and J. Lin. Document ranking with a pretrained sequence-to-sequence model. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 708–718, Online, Nov. 2020. Association for Computational Linguistics.
6. R. Nogueira and J. Lin. From doc2query to docTTTTTquery. Technical report, University of Watrloo, 2019.
7. R. Nogueira, W. Yang, K. Cho, and J. Lin. Multi-Stage Document Ranking with BERT, 2019. https://arxiv.org/abs/1910.14424.
8. I. Ounis, G. Amati, P. V., B. He, C. Macdonald, and Johnson. Terrier Information Retrieval Platform. In *Proceedings of the 27th European Conference on IR Research (ECIR 2005)*, volume 3408 of *Lecture Notes in Computer Science*, pages 517–519. Springer, 2005.
9. R. Pradeep, R. Nogueira, and J. Lin. The expando-mono-duo design pattern for text ranking with pretrained sequence-to-sequence models, 2021.
10. A. Vaswani, N.Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.
11. P. Yang, H. Fang, and J. Lin. 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*, SIGIR '17, page 1253–1256, New York, NY, USA, 2017. Association for Computing Machinery.