CIP at TREC 2022 Deep Learning Track

Jian Luo[1,2], Xinlin Peng[1,2], Xuanang Chen[1,2], Ben He[1,2],
Le Sun[2], and Yingfei Sun[1]

[1] University of Chinese Academy of Sciences, Beijing, China
[2] Chinese Information Processing Laboratory, Institute of Software,
Chinese Academy of Sciences, Beijing, China
{luojian222, pengxinlin22, chenxuanang19}@mails.ucas.ac.cn
{benhe, yfsun}@ucas.ac.cn, sunle@iscas.ac.cn

Abstract. This paper describes the CIP participation in the TREC 2022 Deep Learning Track. We submitted runs to both full ranking and re-ranking subtasks of the passage ranking task. In the full ranking subtask, we adopt a query noise resistant dense retrieval model RoDR. In the re-ranking subtask, we adopt localized contrastive estimation loss and hinge loss rather than pointwise cross-entropy loss for training re-rankers. Besides, We utilize both the MS MARCO v1 and v2 passage datasets to generate hopefully sufficient training data, and our models are fine-tuned on these two kinds of training data one by one selectively. Additionally, we introduce docT5query to further enhance the performance.

1 Introduction

The CIP participation in the TREC 2022 Deep Learning (DL) track focuses on the full ranking and re-ranking subtasks of the passage ranking task.

In the full ranking subtask, our submissions are based on the RoDR model [1], which adopts a query noise resistant dense retrieval training method. We submitted six runs for this subtask including three official runs and three additional runs. The official runs are re-ranked after dense retrieval, and the additional runs don't go though the re-ranking process. The training of dense retrieval models is performed on both MS MARCO v1 and v2 passage datasets. The BM25 [2] retrieval results of docT5query [3] are used to integrated with dense retrieval results to boost the performance.

In the re-ranking subtask, our main model is based on the BERT re-ranker [4]. Besides, we adopt Localized Contrastive Estimation (LCE) [5] and hinge loss rather than normal cross-entropy loss for model training. We use both the MS MARCO v1 and v2 passage datasets in turn with the initial top100 ranking files for fine-tuning the BERT re-ranker. Akin to the setting in full ranking subtask, the BM25 [2] retrieval results of docT5query [3] are also used in this part.

2 Method

2.1 Full Ranking

Dense retrieval (DR) technique has been shown effective in passage retrieval. DR models employ pre-trained language models (PLMs), like BERT [6], and dual-encoder architecture to separately encode queries and passages into low-dimensional dense vectors, and adopt a lightweight similarity mechanism (e.g., dot product) for efficient ad-hoc retrieval. In this subtask, we use a query noise resistant DR model RoDR [1]. A noisy query training set for the origin query training set is generated and RoDR uses KL loss to align the in-batch local ranking between noisy query and origin query. The final training loss is the sum of KL loss and standard cross entropy DR loss. Except for standard DR, two more powerful DR models (Tas-Balanced [7] and PAIR [8]) are used for effectiveness. The docT5query is used to generate queries for passages. Each passage is appended with its predicted queries and then indexed by BM25. After dense retrieval, these BM25 scores on docT5query are interpolated with the dense retrieval scores for benefiting from multi-way matching.

2.2 Re-ranking

As the task of query-based passage re-ranking can be treated as a binary classification problem, pre-trained language models such as BERT have been widely-used. Even through being simply fine-tuned for passage re-ranking using the cross-entropy loss, binary classification model based on BERT-Large achieved decent results [4]. In this subtask, we use Localized Contrastive Estimation (LCE) [5] loss to fine-tune our BERT re-ranker, because it can further improve the re-ranking effectiveness in our experiments. For each query, we generate a passage group which contains a relevant (positive) passage example and several non-relevant (negative) passage examples. The negative passage examples were sampled from the top retrieval results. Based on our participation in the TREC 2021 DL track [9], we remain to use the pairwise hinge loss to fine-tune the BERT re-ranker. Similar to full ranking subtask, we combine the re-ranking results with BM25 results on docT5query for better effectiveness. Note that when applying re-ranking models to our full rank retrieval results, we do not integrate the re-ranking results with docT5query results.

3 Experiments

3.1 Data

Noisy query. Following RoDR [1], we choose eight types of textual noise (like *injected misspellings*) for training. Each query in training set will be transferred to its noisy variation. In noisy queries training set, the number of queries for each noise type is the same.

Training data. Although the MS MARCO v2 corpus has been released, we still use the passage dataset in the MS MARCO v1 corpus. Thus, in our experiments, totally two datasets are used for our model training:

For full ranking models:

- **Passage v1:** In the MS MARCO v1 passage dataset, a training sample consists of a query, a noisy version of the query, a positive passage and n negative passages. The number of negative passages is set as eight and negative passages are sampled from the top candidates of the base model's retrieval results. We construct about 0.40 million training samples.
- **Passage v2:** In the MS MARCO v2 passage dataset, we construct about 0.28 million training samples with the same settings of the Passage v1.

For re-ranking models:

- **Passage v1:** In the MS MARCO v1 passage dataset, we generate 0.50 million training samples, each of those consists of a query, a positive passage and ten negative passages. The negative passages are sampled from the top candidates of the official provided top-1000 file for Passage v1 train queries.
- **Passage v2:** In the MS MARCO v2 passage dataset, other settings are as same as the Passage v1, while the negative passages are sampled from the top candidates of the official provided top-100 file for Passage v2. we construct about 0.28 million training samples with the same settings of the Passage v1.

Validation/Dev data. For our full rank models, we use the official dev sets (3,903 queries in Dev 1 and 4,281 queries in Dev 2) and official validation set (53 queries in TREC 2021 DL) of TREC 2022 DL for our model validation. When BERT re-ranker is trained, we use the official dev sets and official validation set. Additionally we use the official top-100 candidates of dev queries with Passage v2 for model validation.

Table 1. The summary of submitted runs. α means the interpolation weight of our models' retrieval/re-ranking results in docT5query ensemble.

Run ID	**Retriever**(α)	**Re-ranker**(α)	**Passage v1**	**Passage v2**
cip_f1	RoDR(0.5)	-	✓	✓
cip_f2	RoDR w/ TAS(0.5)	-	✓	✓
cip_f3	RoDR w/ PAIR(0.6)	-		✓
cip_r1	-	LCE-Reranker(0.6)		✓
cip_r2	-	Hinge-Reranker	✓	
cip_r3	-	LCE-Reranker(0.5)	✓	✓
cip_f1_r	RoDR w/ TAS(0.5)	LCE-Reranker	✓	✓
cip_f2_r	RoDR(0.5)	LCE-Reranker	✓	✓
cip_f3_r	RoDR w/ PAIR(0.6)	Hinge-Reranker	✓	✓

Table 2. Evaluation results on TREC 2022 DL test queries in the passage retrieval task. Dev shows the average score of the two dev sets (3,903 queries for Dev 1 and 4,281 queries for Dev 2) of TREC 2022 DL. Validation consists the 53 test queries from TREC 2021 DL track. The TREC 2022 DL test set consists of 76 queries. The best values are highlighted in boldface.

Run ID	**Dev**	**Validation**	**TREC 2022 DL Test**		
	NDCG@10	**NDCG@10**	**MAP**	**NDCG@10**	**P@10**
cip_f1	0.1270	0.6385	0.1503	0.4987	0.4079
cip_f2	0.1231	0.6279	0.1480	0.4929	0.4013
cip_f3	0.1240	0.6326	0.1406	0.4781	0.3763
cip_r1	0.1506	0.6014	0.0791	0.4264	0.2882
cip_r2	0.1462	0.6569	0.0930	0.4839	0.3605
cip_r3	0.1491	0.6212	0.0846	0.4549	0.3342
cip_f1_r	0.2080	0.5230	0.1599	0.5007	0.4461
cip_f2_r	**0.2094**	0.4814	**0.1752**	**0.5776**	**0.4882**
cip_f3_r	0.1762	**0.7034**	0.1736	0.5740	0.4789

3.2 Model

As for RoDR retriever, we adopt three kinds of base model: standard DR (based on bert-base-uncased), TAS-Balanced (based on distil-bert) [7] and PAIR (based on ERNIE-2.0 base) [8]. Those the models are fine-tuned on above two kinds of passage training data as described in Section 3.1 with the training order of Passage v1, Passage v2. And the trained models are denoted as RoDR w/ TAS and RoDR w/ PAIR. Besides, as for BERT re-ranker, we adopt the pre-trained BERT-Large models (bert-large-uncased), and they are fine-tuned on Passage v1 and Passage v2 datasets with above two kinds of training loss. The query has up to 32 tokens and the passage has up to 128 tokens for retrieval, while the query has up to 64 tokens and the passage has up to 512 tokens for re-ranking. As for docT5query ensemble, we adopt the original set of *MS MARCO V2 Passage Expansion*[1]. Twenty queries are generated and expanded to per passage in MS MARCO v2 then the expanded passages are indexed with Anserini and finally the BM25 retrieval is utilized to obtained the results. Ensemble is based on the top 2k of BM25 retrieval results because the top results of our neural models could be ranked very far behind in BM25 retrieval results. Then the scores of BM25 results and our models' results are normalized and combined with a weight α. For our submitted official runs, the ensemble results of full ranking are further re-ranked by our models in the re-ranking subtask. The summary of our submitted runs in full ranking and re-ranking subtasks is shown as Table 1.

We carry out our experiments on five TITAN RTX 24G GPUs. For the DR retrieval training, the learning rate is set as 5e-6 for the whole fine-tuning procedure with batch size of 16. And for the BERT re-ranker training, the learning rate is set as 1e-5 and the batch size is set as 64. Besides, the DR retrieval is trained for 4 epoch using both Passage v1 and Passage v2. And the BERT

[1] https://github.com/castorini/docTTTTTquery

re-ranker is trained for 2 epochs using Passage v1 and 2 epochs using Passage v2. We save a checkpoint per 5,000 training steps, and select the checkpoint according to the best NDCG@10 in validation.

3.3 Results

The evaluation results of our submitted runs for passage full ranking and re-ranking subtasks are shown in Table 2. cip_f1, cip_f2, cip_f3, cip_f1_r, cip_f2_r and cip_f3_r are the runs of the full ranking subtask, and the first three are additional runs the last three are official runs. cip_r1, cip_r2 and cip_r3 are the runs of the re-ranking subtask. For a more comprehensive comparison, we also present the validation results on test queries from TREC 2021 DL test set. From the above results, we find that cip_f3_r outperforms other runs on Validation, meanwhile cip_f2_r behaves better than other runs on TREC 2022 DL test set in terms of NDCG@10, P@10 and MAP, and cip_f1_r achieved relatively balanced and good results on both validation and dev set of MSMARCO v2 in our experiment. Thus, cip_f2_r is the best performing run.

4 Conclusions

In this paper, we describe the system based on PLM for both passage full ranking and re-ranking subtasks in TREC 2022 Deep Learning track. Our experiments demonstrate again that adopting re-ranking after full ranking can obtain better result and integrating with sparse retrieval result can improve the dense retrieval result. Meanwhile, we use a query noise resistant training strategy in full ranking retrieval model. In future work, we plan to investigate the matching strategy between full ranking retrieval methods and re-ranking retrieval methods.

Acknowledgements

This work is supported by the University of Chinese Academy of Sciences.

References

1. Chen, X., Luo, J., He, B., Sun, L., Sun, Y.: Towards robust dense retrieval via local ranking alignment. In: IJCAI-22. pp. 1980–1986. International Joint Conferences on Artificial Intelligence Organization (2022)
2. Zeng, Z., Sakai, T.: BM25 pseudo relevance feedback using anserini at waseda university. In: SIGIR 2019. CEUR Workshop Proceedings, vol. 2409, pp. 62–63. CEUR-WS.org (2019)
3. Nogueira, R.F., Yang, W., Lin, J., Cho, K.: Document expansion by query prediction. CoRR **abs/1904.08375** (2019)
4. Nogueira, R., Cho, K.: Passage re-ranking with bert. arXiv preprint arXiv:1901.04085 (2019)

5. Gao, L., Dai, Z., Callan, J.: Rethink training of bert rerankers in multi-stage retrieval pipeline. In: The 43rd European Conference On Information Retrieval (ECIR) (2021)
6. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: NAACL-HLT (1). pp. 4171–4186. Association for Computational Linguistics (2019)
7. Hofstätter, S., Lin, S., Yang, J., Lin, J., Hanbury, A.: Efficiently teaching an effective dense retriever with balanced topic aware sampling. In: SIGIR. pp. 113–122 (2021)
8. Ren, R., Lv, S., Qu, Y., Liu, J., Zhao, W.X., She, Q., Wu, H., Wang, H., Wen, J.: PAIR: leveraging passage-centric similarity relation for improving dense passage retrieval. In: ACL/IJCNLP 2021. Findings of ACL, vol. ACL/IJCNLP 2021, pp. 2173–2183. Association for Computational Linguistics (2021)
9. Chen, X., He, B., Sun, L., Sun, Y.: CIP at TREC 2021 deep learning track. In: TREC (2021)