Learning to Jointly Transform and Rank Difficult Queries

Amin Bigdeli[1], Negar Arabzadeh [1], and Ebrahim Bagheri [2]

[1] University of Waterloo, Waterloo, Canada, {abigdeli,narabzad}@uwaterloo.ca
[2] Toronto Metropolitan University, Toronto, Canada, bagheri@torontomu.ca

Abstract. Recent empirical studies have shown that while neural rankers exhibit increasingly higher retrieval effectiveness on tasks such as ad hoc retrieval, these improved performances are not experienced uniformly across the range of all queries. There are typically a large subset of queries that are not satisfied by neural rankers. These queries are often referred to as *difficult queries*. Given the fact that neural rankers operate based on the similarity between the embedding representations of queries and their relevant documents, the poor performance of difficult queries can be due to the sub-optimal representations learnt for difficult queries. As such, the objective of our work in this paper is to learn to rank documents and also transform query representations in tandem such that the representation of queries are transformed into one that shows higher resemblance to their relevant document. This way, our method will provide the opportunity to satisfy a large number of difficult queries that would otherwise not be addressed. In order to learn to jointly rank documents and transform queries, we propose to integrate two forms of triplet loss functions into neural rankers such that they ensure that each query is moved along the embedding space, through the transformation of its embedding representation, in order to be placed close to its relevant document(s). We perform experiments based on the MS MARCO passage ranking task and show that our proposed method has been able to show noticeable performance improvement for queries that were extremely difficult for existing neural rankers. On average, our approach has been able to satisfy 277 queries with an MRR@10 of 0.21 for queries that had a reciprocal rank of zero on the initial neural ranker.

1 Introduction

The Information Retrieval (IR) community has immensely benefited from Large Language Models (LLMs), such as BERT [9] for improving the performance of ad hoc retrieval [21,27]. There has been a significant boost in the performance of ad hoc retrieval task, which is primarily due to how representations are learnt for terms, queries and documents within a dense embedding space [16]. However, while the overall effectiveness of neural rankers have increased by at least two folds, e.g., based on the MS MARCO passage ranking dataset [18], these improvements have not been uniformly observed over different query subsets [3].

Researchers have traditionally understood that not all queries are satisfied to the same extent by various retrievers [11]; however, given the rather high performance of neural rankers, the number of queries that are not satisfied at all by neural rankers can be surprising [4,1]. A recent study on the effectiveness of neural rankers on the MS MARCO passage dataset showed that out of the $6,980$ queries in its development set, there are at least $2,500$ queries that are not addressed by any state-of-the-art neural ranker (queries with a reciprocal rank of zero). This indicates that neural rankers improve overall average retrieval effectiveness by focusing on a particular subset of queries at the expense of another subset [3,5].

In addition, there have works that have reported that the performance of neural rankers can be sensitive to the input query and the retrieval effectiveness is very dependent on how the input query is formulated [26,2,28]. Various researchers have already extensively explored how methods such as query expansion [8,30,6] and query reformulation [15,22,17] can be used to improve the effectiveness of rankers. However, most, if not all, of such methods are designed specifically for reformulating the textual surface form of the query by adding, replacing or removing terms from the original query such that the retrieval effectiveness of the reformulated query is stronger than that of the original query. While such methods have shown a strong impact on sparse retrievers [24], recent research has shown that they are not effective for improving the performance of neural rankers [2] and can even lead to degraded overall performance.

Based on observations, neural rankers (1) are quite effective in retrieving highly relevant documents for a subset of queries. According to [3], these queries are common among various neural rankers and are often referred to as easy queries; and, (2) fall short of effectively, or even minimally, addressing another large subset of queries, which we refer to as *difficult queries*. A typical neural ranker would not require any assistance in addressing easy queries; however, they would require additional mechanisms to help them effectively address difficult queries. To this end, we hypothesize that difficult queries are those whose embedding representations are not placed well within the embedding space, i.e., they are placed closer to irrelevant documents and further away from relevant ones. A possible solution to this problem would be to transform the representation of queries such that difficult queries would be placed closer to their relevant documents; hence, leading to their improved retrieval effectiveness.

The goal in this paper is to transform the representation of queries in order to improve the performance of difficult queries. To achieve this, we propose to perform (1) query representation transformation, and (2) passage ranking tasks in tandem where the transformation of the query does not happen on its surface form but rather happens through the translation of the query representation into a position within the embedding space that places it closer to its relevant passage and moves it away from other irrelevant passages. We perform our experiments on the MS MARCO passage ranking dataset and over a range of state-of-the-art neural ranking methods. Based on our experimental results, we show that our strategy to learn to transform queries and rank documents in tandem results in higher retrieval effectiveness specially on difficult queries.

The key contributions of our work can be enumerated as: (1) we propose an approach to learn to rank documents and transform queries at the same time; (2) we show that our approach is able to consistently improve the performance of the most difficult queries for various neural rankers. Furthermore, the advantages of our proposed training strategy include (a) it is widely applicable to a range of neural rankers as it does not require any changes to the architecture of the neural ranking methods; and (b) it does not require any additional overhead for learning to transform queries as this happens at the same time as when the ranking method is being trained.

2 Proposed Approach

Typically neural rankers are only trained based on relevance triplets where the network is finetuned to transform the query representation whereby the transformed representation is closer to the relevant document. Our proposed approach in this paper is based on the idea of learning to rank documents and transform queries in tandem. As such, and in order to train a neural-based retrieval method R, we need to consider two different types of triplets, namely (1) relevance triplets $< q, d_q^+, d_q^- >$ where q is the original query from a set of training queries Q, d_q^+ is the relevant document for query q, d_q^- is a negative sample document for query q; and, (2) transformation triplets $< q, q_q^+, q_q^- >$ where q_q^+ is an ideal transformation of q and q_q^- is an irrelevant (negative) query to q. We propose that the inclusion of transformation triplets in the training process of neural rankers will enable neural rankers to transform the representation of queries such that they are placed closer to the representation of their ideal alternative query.

To operationalize the training of neural rankers based on these two types of triplets, marginal ranking loss [21] is used for both the relevance triplets and transformation triplets to calculate the total loss of the neural network as follows:

$$L_{Relevance} = \frac{1}{|Q|} \sum_{i=1}^{N^+} \sum_{j=1}^{N^-} \max(0, m - ||q - d_i^+|| + ||q - d_j^-||) \tag{1}$$

$$L_{QueryTransformation} = \frac{1}{|Q|} \sum_{i=1}^{N^+} \sum_{j=1}^{N^-} \max(0, m - ||q - q_i^+|| + ||q - q_j^-||) \tag{2}$$

We define the total loss for the neural ranker as the sum of the loss of the relevance triplets and query transformation triplets expressed as $L = L_{Relevance} + L_{QueryTransformation}$. As a result, during the training procedure of the model, the goal of the loss function is to tune the network in a way that $sim(q, d_q^+) > sim(q, d_q^-)$ and $sim(q, q_q^+) > sim(q, q_q^-)$ conditions are maximized. In other words, the network is fine-tuned to transform the query representation in a way that the transformed representation is closer to the relevant document. On this basis, the model not only learns relevance but also transforms the representation of the original query to be better suited for retrieving the relevant document.

The training of a neural ranker based on the proposed loss function will require access to data samples to form the two types of triplets. It is not difficult to obtain relevance triplets based on relevance judgment collections. Let us define $D^k_{q_i} = \{d^1_{q_i}, d^2_{q_i}, ..., d^k_{q_i}\}$ as a list of k retrieved documents by a first-stage retrieval method M for query q_i from Q. Also let $D^+_{q_i} = \{d^1_{q_i}, d^2_{q_i}, ..., d^n_{q_i}\}$ and $D^-_{q_i} = \{d^1_{q_i}, d^2_{q_i}, ..., d^l_{q_i}\}$ be the list of n judged documents and a set of l irrelevant documents selected randomly from $D^k_{q_i}$ for query q_i, respectively. Relevance triplets can be randomly sampled from $D^+_{q_i}$ and $D^-_{q_i}$. The challenge; however, is to produce transformation triplets as they require access to ideal query representation for the initial input query. For this purpose, let us assume that we have access to a transformation function $\mathcal{T}$ capable of generating synthetic queries for a judged document $d^r_{q_i}$ randomly selected from $D^+_{q_i}$ of query q_i.

Based on $\mathcal{T}$, it would be possible to generate a set of queries for $d^r_{q_i}$ that would maximize retrieval effectiveness by retrieving $d^r_{q_i}$ in response to that query. For instance, given a document $d^r_{q_i}$, and the transformation function $\mathcal{T}$, we can generate $Q'_{d^r_{q_i}}$ by applying $\mathcal{T}$, expressed as $Q'_{d^r_{q_i}} = \mathcal{T}(d^r_{q_i})$.

By using $\mathcal{T}$, we can define $Q'_{q_i}{}^+ = \{q'_{q_i}{}^1, q'_{q_i}{}^2, ..., q'_{q_i}{}^p\}$ as a list of p queries generated based on the judged documents $D^+_{q_i}$ associated with query q_i. Having $Q'_{q_i}{}^+$, we can calculate the retrieval effectiveness of each of the generated queries using a first-stage retrieval method M and then select the best performing query $q'_{q_i}{}^+$ as the ideal query representation of query q_i. We note that earlier research, such as [19,20], have shown that $\mathcal{T}$ can be learnt using a T5 transformer architecture and trained on a large relevance judgment collection.

Furthermore and in order to pair query q_i with a negative query that is irrelevant to q_i, we randomly select a query from the training query set Q and consider it as $q'_{q_i}{}^-$ as suggested in the literature [31,12]. Representing $q'_{q_i}{}^+$ and $q'_{q_i}{}^-$ as the relevant and irrelevant pairs of q_i would help the model to transform the representation of q_i closer to $q'_{q_i}{}^+$ that is capable of gaining higher retrieval effectiveness for retrieving the relevant document of q_i and also place it far away from the negative query $q'_{q_i}{}^-$ that is irrelevant.

3 Experiments

3.1 Experimental Setup

Code. The artifacts of our work are released on Github[3] for general use.

Document and Query Collection. We conduct our experiments on the widely used MSMARCO collection that consists of over 8.8M passages and over 500k queries. For the query set, we use the small dev set of the MS MARCO collection, which is frequently used for evaluation purposes. This dataset consists of 6,980 queries along with their relevance judged documents.

[3] `https://github.com/aminbigdeli/query_transformation`

Table 1: Comparison between the performance (MRR@10) of different neural ranking models and ours.

Architecture	LLM	Original	Ours
Sentence Transformer	BERT	0.334	0.337
	MiniLM	0.319	$0.325^{\dagger}$
	DistilRoBERTa	0.305	0.308
ColBERT	BERT	0.338	$0.342^{\dagger}$
RepBERT	BERT	0.287	$0.290^{\dagger}$

Neural Rankers. To show the effectiveness of our proposed training strategy, we use different types of neural-based retrieval methods and compare the retrieval effectiveness of those models when they are trained on the original training dataset and when they are trained based on our proposed training strategy. The neural models used in our experiments include (1) S-BERT (Sentence Transformer) [21] with different pre-trained language models including BERT-base-uncased [10], DistilRoBERTa-base [23], and MiniLM [25]; (2) ColBERT [13], and (3) RepBERT [29].

Model Details and Hyperparameters. To train S-BERT with different pre-trained natural language models, we used the bi-encoder architecture of Sentence Transformer library with batch size being set to 64, number of epochs to 5, and the maximum sequence length is set to 350. To train the RepBERT model, batch size is set to 26 , the number of epochs is 1, the maximum document length is set to 256, and the maximum query length is set to 20. The ColBERT model is also trained with the default parameters with a batch size of 32, number of epochs to 1, the maximum document length of 180, and a maximum query length of 32. In order to implement the transformation function $\mathcal{T}$ introduced in Section 2, as suggested by [19], we employed the T5 Transformer trained based on the 500k queries in the MS MARCO collection and their relevant documents and generate queries from relevant documents that are semantically similar to the training query and might achieve a higher retrieval effectiveness when used for retrieval. Having a set of generated queries, we select the one with the highest retrieval effectiveness and consider it as the positive pair for each of the training queries as released in [2]. As for the negative pair of each training query, we randomly select a query from the training set and consider it as the negative query. Finally, having the relevant and irrelevant documents as well as relevant and irrelevant queries for each of the training set queries, we can train the aforementioned networks by passing the representation of the query and each of the other sequences (relevant/irrelevant document and relevant/irrelevant query) to each of the networks for training.

3.2 Findings

The main purpose of performing query transformation is to help *difficult queries* achieve better retrieval effectiveness. As such, we first report the performance

Table 2: Comparison between the MRR@10 of the different neural ranking models trained based on the original dataset and our proposed one on difficult queries among five buckets each consisting of 698 queries.

Architecture	LLM	Training	0-10%	10-20%	20-30%	30-40%	40-50%
Sentence Transformer	`BERT`	Original	0	0	0	0.002	0.133
		Ours	0.024†	0.02†	0.022†	0.026†	0.148†
	`MiniLM`	Original	0	0	0	0	0.096
		Ours	0.022†	0.017†	0.015†	0.023†	0.115†
	`DistilRoBERTa`	Original	0	0	0	0	0.062
		Ours	0.021†	0.023†	0.022†	0.019†	0.078†
`ColBERT`	`BERT`	Original	0	0	0	0.017	0.142
		Ours	0.015†	0.013†	0.021†	0.034†	0.166†
`RepBERT`	`BERT`	Original	0	0	0	0	0.04
		Ours	0.011†	0.016†	0.014†	0.018†	0.05†

of our proposed fused approach [7] compared to the performance of the various state-of-the-art neural baselines [21,13,29] over all queries in Table 1. We note that statistical significance is measured based on paired t-test with p-value of 0.05 and denoted with superscript † in the tables. As seen in Table 1, our approach exhibits a slightly superior performance compared to the baselines. Therefore, across all queries, our proposed approach that learns to rank and transform queries maintains slightly better levels of retrieval effectiveness. We additionally show, as also already reported in the relevant literature [14], that **Pseudo-Relevant Feedback (PRF) methods** do not necessarily lead to improvement on these baselines. For instance when applying PRF on the baselines, namely SBERT-PRF and ColBERT-PRF, we obtain an MRR@10 of 0.3035 and 0.2816, respectively. These are consistently lower than the performance of our model.

Given our focus is on *difficult queries*, we evaluate the impact of our approach on the retrieval effectiveness of such queries. We define difficult queries for a ranking method to be those that have a poor performance when retrieved by that ranking method. Ensan et. al., suggested [11] that one can consider the lowest-performing queries to be the set of difficult queries for a ranking method. More specifically, we define the most difficult queries for a ranker to be those that fall in the lower half of retrieval effectiveness compared to other queries. To identify such queries, we rank-order queries based on their MRR@10 values and choose the bottom 50% of queries to represent difficult queries. Given there are 6,980 queries in the small dev set, the bottom 50% of the queries include 3,490 queries in total. We further split these queries into five finer-grained difficulty buckets and report the performance of the baseline rankers as well as the performance of our proposed method in each of these difficulty buckets. Table 2 reports the results of method performance across different difficulty buckets. As seen in the table, the bottom 3 buckets (bottom 30% of queries) in all neural rankers have a reciprocal rank

Table 3: The number of MS MARCO dev small queries helped by our method from the bottom 50% of the most difficult queries for each ranker.

Architecture	LLM	Number of Queries	MRR@10
Sentence Transformer	`BERT`	270 out of 2,776	0.23
	`MiniLM`	275 out of 2,942	0.20
	`DistilRoBERTa`	308 out of 3,106	0.22
`ColBERT`	`BERT`	258 out of 2,671	0.17
`RepBERT`	`BERT`	291 out of 3,233	0.17

value of zero i.e., on these queries ($\sim 2,000$), none of the neural rankers are able to retrieve a relevant document in its top-10 ranked documents. In contrast, our approach has been able to effectively address a subset of the hard queries in each difficulty bucket. While the MRR@10 scores obtained by our method are relatively small, this should be interpreted in the context of the fact that no neural baseline included in our experiments was able to address any of the queries in this set and the absolute MRR@10 score for all of them is zero.

We further qualify the findings in Table 2 by exploring the number of queries that have been helped by our approach in each baseline. In particular, we report the number of queries that had an original reciprocal rank score of zero on the original neural ranker but later received a higher reciprocal rank score by our proposal approach. The number of such queries along with their average MRR@10 values are reported in Table 3. As seen in this table, on average, our proposed approach has been able to improve the performance of 277 queries per neural ranker. This means that there are on average 277 queries that were originally completely unsatisfied by the base ranker and then addressed by the proposed approach. The average improvement on MRR@10 over these queries is 0.21. This shows that when the representation of the original query is transformed in the embedding space, a noticeable number of difficult queries are then effectively, or at least partially, addressed by the neural ranker.

4 Concluding Remarks

In this paper, we have proposed a strategy to learn to rank documents while at the same time learning an embedding transformation from the original query to one that maximizes retrieval effectiveness. The idea of our approach is that extremely difficult queries can be addressed if their embedding representations are moved closer to an ideal query representation that has a high retrieval effectiveness. Based on our experiments on MS the MARCO passage collection development set, we have shown that the representation of the transformed queries from our proposed approach can lead to increased performance on difficult queries.

References

1. Arabzadeh, N., Bigdeli, A., Hamidi Rad, R., Bagheri, E.: Quantifying ranker coverage of different query subspaces. In: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 2298–2302 (2023)
2. Arabzadeh, N., Bigdeli, A., Seyedsalehi, S., Zihayat, M., Bagheri, E.: Matches made in heaven: Toolkit and large-scale datasets for supervised query reformulation. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management. pp. 4417–4425 (2021)
3. Arabzadeh, N., Mitra, B., Bagheri, E.: Ms marco chameleons: Challenging the ms marco leaderboard with extremely obstinate queries. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management. pp. 4426–4435 (2021)
4. Arabzadeh, N., Vtyurina, A., Yan, X., Clarke, C.L.A.: Shallow pooling for sparse labels. CoRR **abs/2109.00062** (2021), `https://arxiv.org/abs/2109.00062`
5. Arabzadeh, N., Yan, X., Clarke, C.L.A.: Predicting efficiency/effectiveness tradeoffs for dense vs. sparse retrieval strategy selection. CoRR **abs/2109.10739** (2021), `https://arxiv.org/abs/2109.10739`
6. Azad, H.K., Deepak, A.: Query expansion techniques for information retrieval: a survey. Information Processing & Management **56**(5), 1698–1735 (2019)
7. Bassani, E., Romelli, L.: ranx.fuse: A python library for metasearch. In: Hasan, M.A., Xiong, L. (eds.) Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, October 17-21, 2022. pp. 4808–4812. ACM (2022). https://doi.org/10.1145/3511808.3557207, `https://doi.org/10.1145/3511808.3557207`
8. Carpineto, C., Romano, G.: A survey of automatic query expansion in information retrieval. Acm Computing Surveys (CSUR) **44**(1), 1–50 (2012)
9. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
10. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
11. Ensan, F., Bagheri, E.: Document retrieval model through semantic linking. In: Proceedings of the tenth ACM international conference on web search and data mining. pp. 181–190 (2017)
12. Karpukhin, V., Oğuz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., Yih, W.t.: Dense passage retrieval for open-domain question answering. arXiv preprint arXiv:2004.04906 (2020)
13. Khattab, O., Zaharia, M.: Colbert: Efficient and effective passage search via contextualized late interaction over bert. In: Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval. pp. 39–48 (2020)
14. Li, H., Mourad, A., Zhuang, S., Koopman, B., Zuccon, G.: Pseudo relevance feedback with deep language models and dense retrievers: Successes and pitfalls. ACM Transactions on Information Systems **41**(3), 1–40 (2023)
15. Li, X., Mao, J., Ma, W., Wu, Z., Liu, Y., Zhang, M., Ma, S., Wang, Z., He, X.: A cooperative neural information retrieval pipeline with knowledge enhanced automatic query reformulation. In: Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining. pp. 553–561 (2022)

16. Lin, J., Nogueira, R., Yates, A.: Pretrained transformers for text ranking: BERT and beyond. CoRR **abs/2010.06467** (2020), https://arxiv.org/abs/2010.06467
17. Lioma, C., Ounis, I.: A syntactically-based query reformulation technique for information retrieval. Information processing & management **44**(1), 143–162 (2008)
18. Nguyen, T., Rosenberg, M., Song, X., Gao, J., Tiwary, S., Majumder, R., Deng, L.: Ms marco: A human generated machine reading comprehension dataset. choice **2640**, 660 (2016)
19. Nogueira, R., Lin, J., Epistemic, A.: From doc2query to docTTTTTquery. Online preprint **6** (2019)
20. Nogueira, R.F., Yang, W., Lin, J., Cho, K.: Document expansion by query prediction. CoRR **abs/1904.08375** (2019), http://arxiv.org/abs/1904.08375
21. Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. arXiv preprint arXiv:1908.10084 (2019)
22. Rieh, S.Y., et al.: Analysis of multiple query reformulations on the web: The interactive information retrieval context. Information Processing & Management **42**(3), 751–768 (2006)
23. Sanh, V., Debut, L., Chaumond, J., Wolf, T.: Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108 (2019)
24. Tamannaee, M., Fani, H., Zarrinkalam, F., Samouh, J., Paydar, S., Bagheri, E.: Reque: a configurable workflow and dataset collection for query refinement. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management. pp. 3165–3172 (2020)
25. Wang, W., Wei, F., Dong, L., Bao, H., Yang, N., Zhou, M.: Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. Advances in Neural Information Processing Systems **33**, 5776–5788 (2020)
26. Wang, X., Macdonald, C., Ounis, I.: Deep reinforced query reformulation for information retrieval. arXiv preprint arXiv:2007.07987 (2020)
27. Xiong, L., Xiong, C., Li, Y., Tang, K.F., Liu, J., Bennett, P., Ahmed, J., Overwijk, A.: Approximate nearest neighbor negative contrastive learning for dense text retrieval. arXiv preprint arXiv:2007.00808 (2020)
28. Zerveas, G., Zhang, R., Kim, L., Eickhoff, C.: Brown university at trec deep learning 2019. arXiv preprint arXiv:2009.04016 (2020)
29. Zhan, J., Mao, J., Liu, Y., Zhang, M., Ma, S.: Repbert: Contextualized text embeddings for first-stage retrieval. arXiv preprint arXiv:2006.15498 (2020)
30. Zheng, Z., Hui, K., He, B., Han, X., Sun, L., Yates, A.: Bert-qe: contextualized query expansion for document re-ranking. arXiv preprint arXiv:2009.07258 (2020)
31. Zhou, K., Gong, Y., Liu, X., Zhao, W.X., Shen, Y., Dong, A., Lu, J., Majumder, R., Wen, J.R., Duan, N., et al.: Simans: Simple ambiguous negatives sampling for dense text retrieval. arXiv preprint arXiv:2210.11773 (2022)