Learning Query-Space Document Representations for High-Recall Retrieval

Sara Salamat[1(✉)], Negar Arabzadeh[2], Fattane Zarrinkalam[3], Morteza Zihayat[1], and Ebrahim Bagheri[1]

[1] Toronto Metropolitan University, Toronto, ON, Canada
{sara.salamat,mzihayat,bagheri}@torontomu.ca
[2] University of Waterloo, Waterloo, ON, Canada
narabzad@uwaterloo.ca
[3] University of Guelph, Guelph, ON, Canada
fzarrink@uoguelph.ca

Abstract. Recent studies have shown that significant performance improvements reported by neural rankers do not necessarily extend to a diverse range of queries. There is a large set of queries that cannot be effectively addressed by neural rankers primarily because relevant documents to these queries are not identified by first-stage retrievers. In this paper, we propose a novel document representation approach that represents documents within the query space, and hence increases the likelihood of recalling a higher number of relevant documents. Based on experiments on the MS MARCO dataset as well as the hardest subset of its queries, we find that the proposed approach shows synergistic behavior to existing neural rankers and is able to increase recall both on MS MARCO dev set queries as well as the hardest queries of MS MARCO.

1 Introduction

There have been recent works in the literature that have shown the approach adopted by neural ranking models that captures relevance through learning document and query representations that maximizes the similarity of relevant queries and documents and minimizes the relevance of dissimilar ones does not necessarily scale to a full range of different query types [8,11,28,29]. For instance, Arabzadeh et al. found that regardless of the underlying neural ranking architecture, neural rankers are not able to satisfy a large group of queries (an average precision of zero) within the MS MARCO collection. These queries were referred to as MS MARCO *Chameleons*. The long-tailed performance of state-of-the-art neural ranking models on gold standard collections, such as MS MARCO, indicates that it is important to identify ways through which all queries, especially those that are hard for neural ranking models, can be handled effectively.

One of the main observations about hard queries is that they not only struggle with poor precision but also struggle with low recall. In essence, the poor recall on such queries can also explain the low precision since due to the heavy computation cost of neural models for full-collection retrieval, most existing neural ranking models are specifically devoted to re-ranking a set of candidates retrieved by a first-stage retriever [14]. In any full ranking stack, whether it is industrial [3,13,26] or research-oriented

J. Kamps et al. (Eds.): ECIR 2023, LNCS 13981, pp. 599–607, 2023.
https://doi.org/10.1007/978-3-031-28238-6_51

[7,24], the goal of the first stage of the retrieval a.k.a *the recall stage*, is to collect all potential relevant documents w.r.t the query using computationally cheap and efficient methods. Further, in the ranking stack, the retrieved pool of candidates will get re-ranked with more expensive, complex, and accurate rerankers [10,12,16,20]. Hence, the main objective of the first stage of a full-ranking stack is to efficiently provide a high-recall pool of document candidates. As such, neural ranking models will only be able to show improved precision if relevant documents are already retrieved and included in the list of documents retrieved by the first-stage retriever. However, in practice, first-stage retrievers struggle with finding a sufficient number of relevant documents for harder queries (low recall), which translates into poor precision by neural rankers.

Existing research has hinted at the fact that the low recall can be due to the difficulty associated with learning appropriate representations for hard queries, their relevant documents, or both [22,30]. In other words, inappropriate query or document representations can significantly impact recall. For example, Bagheri et al. [2] have shown that the choice of document representation can have a notable impact on recall. As such, there have been approaches that explore ways through which more effective query and document representations can be learned. Nogueira et al. [17] have been among the first to explore how document representations could be slightly modified to improve retrieval effectiveness. They found that appending documents with artificially-generated queries from that document using a transformer architecture can lead to noticeable performance improvement. Similarly, Dai and Callan advocated for the idea of learning document term weights that could then lead to a more effective weighted document representation and hence more effective retrieval [5].

Inspired by such studies that have shown the impact of document representation on recall, in this paper, we aim specifically for high-recall retrieval, especially for harder queries. We hypothesize that harder queries with poor recall are those whose relevant documents' representations are not similar to the query itself and, as such, the first-stage retriever is not able to retrieve the relevant documents in the first stage. In such cases, the relevant documents lack any notable resemblance to their relevant query; therefore, we propose to fully replace the original document with a more concise representation of that document. This representation is derived by learning a transformer architecture that learns to generate a query from a document when trained on a collection of gold query-document pairs. In our approach, the original document is replaced by a query-inspired representation of that document, which has the following benefits: (b1) given the new document representation is in the form of a query, learning embeddings that would match the query and the new document representation could be easier; and, (b2) the new document representation is a reformulation of the document but in query space; hence increases the chances of being effectively matched with the relevant query.

In order to evaluate our work, we conduct our experiments on the MS MARCO passage collection [15] and show that our proposed concise document representation so called as $q2q$ (Query to Query-space representation) is able to retrieve a non-overlapping set of relevant documents compared to the original first-stage retrievers. We show that by systematically integrating the results of our work with that of the first-stage retrievers, we are able to improve recall significantly on the queries from the MS

MARCO development set. We also show that such an improvement is not only observed over all of the queries but the improvements are much more substantial on the harder MS MARCO queries known as MS MARCO Chameleons.
Reproducibility: We publicly release the code, data, and trained models on https://github.com/sara-salamat/queryspace-representation.

2 Proposed Approach

The objective of our work is to facilitate high recall in first-stage retrievers by an alternative document representation that is closer to query representations. We propose to replace each document with its corresponding query representation where the query representation is generated by a transformer trained specifically on query-document pairs. On this basis, our approach consists of two steps: 1) learning alternative document representations; and 2) training a neural ranking model to learn the association between the query representation and the reformulated document representations.
Step 1. Learning Alternative Document Representations: In order to learn alternative document representations that are closer to the query space, we are inspired by methods such as Doc2Query, which modify document representations by appending additional query terms to the document. Unlike these methods and instead of expanding the document, we are interested in fully replacing the document representation with one in the query space. We believe such an approach will ensure that the document space is sufficiently close to the query space to lead to improved recall. To this end, we adopt a transformer architecture to generate queries from an input document. More formally, we let $\mathcal{G}$ be a query translation function, which is trained to generate queries from an input document. With $\mathcal{G}$, we will be able to generate query representations for each document in the document corpus $\mathcal{D}$ such that each generated query would be able to efficiently retrieve the document it was generated from. Therefore, given a document $d \in \mathcal{D}$, and the translation function $\mathcal{G}$, we generate $\hat{q}_d$ as $\hat{q}_d = \mathcal{G}(d)$.

It has been shown that $\mathcal{G}$ can be efficiently learned [17] by fine-tuning a transformer [23] based on a relevant judgment dataset. Simply put, based on the association between existing queries and their associated relevant documents, the transformer will learn to generate queries for a given document. Such a fine-tuned transformer will act as $\mathcal{G}$, and since the translation function is not deterministic, we can generate multiple queries for each document by translating the document several times. Hence, we can generate a query set $\hat{Q}_d$ per document $d \in \mathcal{D}$. Ideally, $\hat{Q}_d$ can be interpreted as a set of all queries that can be answered by document d. Moreover, for each document d, we define the query-to-query representation of document d as $q2q(d)$ through the concatenation of its corresponding generated query set, as follows:

$$q2q(d) = concat(\hat{q}_i) | \hat{q}_i \in \hat{Q}_d$$

In our work, we propose to use $q2q(d)$ as the alternative representation for document d.
Step 2. Training Neural Ranker based on Alternative Document Representation: Similar to the training strategy adopted for neural ranking models (dense retrievers), given a query q and its set of relevant documents R_q^+, we fine-tune a large pre-trained language model to maximize the similarity between representations of a query and the documents. In essence, a neural ranking model learns a mapping function ϕ,

which maximizes the similarity of the representation of the query ($\phi(q)$) and its relevant documents ($\phi(R_q^+)$ and minimizes the similarity of the representation of the query and its irrelevant documents $\phi(R_q^-)$. Such a mapping function is often obtained by fine-tuning contextualized language models such as BERT [6]. In the context of our work, we fine-tune neural ranking architectures to maximize the similarity between the representation for q and the reformulated representation of a document based on its set of generated queries $q2q(R_q^+)$. This neural ranking model learns the new representations of the query and documents by maximizing the similarity of $\phi(q)$ and $\phi(q2q(R_q^+))$ minimizing the similarity with Negative sampled documents $\phi(q2q(R_q^-))$.

3 Experiments

Dataset. We evaluate our proposed approach on the MS MARCO passage collection [15]. We trained both our generation function and our neural ranking models on the MS MARCO training set and evaluated them on the 6,980 small MS MARCO dev set queries, which are intended for evaluation purposes.
Query Sets. We perform the evaluation on the small MS MARCO dev set queries as well as the set of its poorly performing queries, a.k.a. MS MARCO Chameleons [1]. The MS MARCO Chameleons consists of three sets: 1) Veiled Chameleons ("Hard" set); 2) Pygmy Chameleons ("Harder" set); and, 3) Lesser Chameleon ("Hardest" set). We refer the interested reader to [1] for more details on the Chameleons sets.
Query Translation Function. To generate alternative document representations, we fine-tuned a T5 transformer based on the query-document pairs of the MS MARCO train set. We ran experiments by representing documents through a set of k corresponding queries where $k \in \{5, 10, 20, 30\}$. We explore the impact of k in our experiments.
Dense Retriever. For the neural ranking model, we adopt the widely-used Sentence-BERT (SBERT) [18], which has shown to have strong retrieval performance and low computational overhead. To have a fair comparison, and due to computational limitations, we performed fine-tuning for 5 epochs on the MS MARCO train set with lr = $2e-5$. Our model uses Multiple Negatives Ranking Loss (MNRL) [9]. We used five negative samples for each query and used DistilBERT [19] as our base model for training.
Baselines. We compare our work with two state of the art document representation techniques, namely DeepCT [5] and Doc2Query [17]. DeepCT learns a weighted representation of the document based on a neural attention mechanism, while Doc2Query expands the initial document representation with additional query-related terms.

4 Results and Findings

Impact of the Number of Generated Queries. In Fig. 1(a), we report the performance of our proposed approach in terms of recall@k where $k \in \{10, 20, 100, 200, 500, 1000\}$ on MS MARCO dev set when representing the document with N-generated queries where $N \in \{5, 10, 20, 30\}$. As shown in this Figure, we observe that the number of queries used to form the alternative document representation has a notable impact on performance. This is especially noticeable as we increase the number of queries from 5

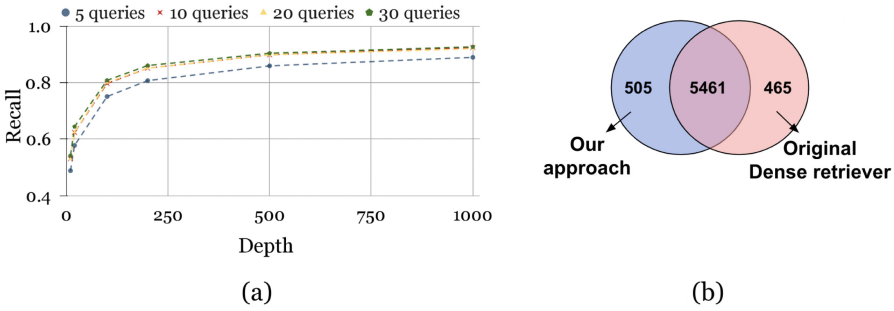

Fig. 1. (a) Performance of our proposed representation in terms of recall at different cutoffs. (b) The number of retrieved relevant documents on top-100 retrieved documents.

to 20. However, adding more queries after 20 does not lead to any statistically significant improvements in performance (paired t-test with 95% confidence interval). This observation is aligned with other document expansion work [17] where the authors also reported that after appending 20 queries to the document, there were no significant improvements observed in retrieval performance. Thus, for the rest of the experiments reported in this paper, we report the results with documents represented by 20 queries.

Performance Comparison. To evaluate the performance of our proposed approach, we compare its performance to that of the base dense retriever in Table 1. As shown, the performance of the two models is quite competitive and similar to each other in terms of recall at different depths. We note that the original dense retriever is performing slightly better at different depths. However, upon further in-depth inspection of performance, we find that while the models have comparable quantitative performance, they do not necessarily have overlapping retrieval performance in practice. In other words, the similar measured performance is not due to a similar retrieval at the query level since the two models are showing retrieval effectiveness on non-overlapping sets of relevant documents. Figure 1(b) exhibits this performance where the number of unique relevant documents as well as the number of overlapping relevant documents retrieved by two rankers are shown. As seen, there are 505 unique relevant documents that are retrieved by our method that are not identified by the base retriever and similarly 465 unique relevant documents that were not identified by our method while there are 5,461 shared relevant documents between the two methods. This is a clear indication of synergistic behavior between the two models. As we will show later in our experiments, our method has been able to identify relevant documents for harder queries that are not retrieved by the base method. As such, as proposed in literature [4,21,27], we adopt the pairwise reciprocal rank fusion between the original runs and our approach to interpolate the two runs and benefit from the complementary behavior of the two models.

Table 1 shows the results of the integration of our method with the base retriever using the pairwise reciprocal rank fusion. From Table 1, we observe that (1) selecting the pool of candidates from the combined pool of retrieved documents from the base retriever as well as our proposed approach leads to a constant increase in recall. The observed differences are statistically significant on all query sets at all depths (paired t-test with 95% confidence interval). (2) As noted earlier in the paper, first stage retriever

Table 1. Recall values of our proposed approach at different cut-offs on MS MARCO dev set as well as the Chameleons query subsets (hard, harder, and hardest).

		Recall cut-off					
	Retrieval method	10	20	100	200	500	1000
MS MARCO Dev Set	SBERT	0.5457	0.6423	0.8053	0.8549	0.9003	0.9259
	Doc2query	0.4502	0.543	0.7193	0.786	0.8536	0.8919
	DeepCT	0.4761	0.5725	0.7537	0.8097	0.872	0.9035
	$q2q$	0.5291	0.6244	0.7996	0.8509	0.898	0.9197
	Interpolated $q2q$	0.5731	0.6691	0.847	0.8926	0.9334	0.9500
	%Improvement	**5.02%**	**4.17%**	**5.18%**	**4.41%**	**3.68%**	**2.60%**
Veiled (hard)	SBERT	0.2065	0.3344	0.6153	0.7085	0.7996	0.8491
	Doc2query	0.1203	0.2071	0.4642	0.5777	0.7075	0.7785
	deepct	0.0872	0.2046	0.5137	0.618	0.7381	0.8012
	$q2q$	0.2038	0.3277	0.6200	0.7123	0.7995	0.8436
	Interpolated $q2q$	0.2354	0.3638	0.6797	0.7714	0.849	0.8887
	%Improvement	**14.00%**	**8.79%**	**10.47%**	**8.88%**	**6.18%**	**4.66%**
Pygmy (harder)	SBERT	0.1441	0.2616	0.5602	0.6674	0.7731	0.8289
	Doc2query	0.0695	0.136	0.3944	0.5173	0.6606	0.7432
	deepct	0.0369	0.1303	0.445	0.5596	0.6948	0.7674
	$q2q$	0.1499	0.2587	0.5706	0.6756	0.7761	0.8273
	Interpolated $q2q$	0.1677	0.2866	0.626	0.7341	0.8258	0.8725
	%Improvement	**16.38%**	**9.56%**	**11.75%**	**9.99%**	**6.82%**	**5.26%**
Lesser (hardest)	SBERT	0.0871	0.1806	0.4818	0.6051	0.7214	0.7871
	Doc2query	0.0269	0.0627	0.2889	0.4123	0.5778	0.6818
	deepct	0.0012	0.0605	0.3437	0.4692	0.6307	0.7159
	$q2q$	0.095	0.1799	0.4907	0.6133	0.7302	0.7949
	Interpolated $q2q$	0.1035	0.2021	0.5396	0.6718	0.7812	0.8399
	%Improvement	**18.83%**	**11.90%**	**12.00%**	**11.02%**	**8.29%**	**6.71%**

methods and in general neural rankers struggle to satisfy hard queries especially those represented in the MS MARCO Chameleons dataset. We report performance of our work on the three variations of the Chameleons dataset. It is important to note that while our approach leads to a noticeable improvement of $\sim$5% on recall@100 on the whole MS MARCO dev set, this improvement is at least $\sim$10% on the Chameleons dataset (2x higher than the overall dataset). This is a significant observation, since as reported in [17], most queries in Chameleons showed an average precision of zero indicating that neural rankers are not able to retrieve any relevant documents for these queries. Therefore, a significant boost in the number of relevant documents returned by the first stage retriever has the potential to impact their overall retrieval effectiveness in the next stage. The statistically significant improvement over recall, especially on the Chameleons dataset, is an indication that our proposed representation is quite effective for hard queries. (3) Finally, when comparing our work with two state-of-the-art document representation baseline methods, namely, DeepCT and Doc2Query, we find that

our proposed $q2q$ method shows a better performance compared to both of the methods, with and without interpolation, on all four query sets at various cut-off points.

We note that in order to study the generalizability of our approach, we replicated the experiments with dense retrievers using other base language models, e.g., miniLM-L6-v2 [25]. While noting that the results were consistent with the above-mentioned findings, due to limited space, we have included these results in our Github repository.

5 Concluding Remarks

Our work in this paper builds on observations from the literature that have shown neural rankers are not as equally effective across a range of queries, i.e., while they significantly improve the performance of a subset of queries, they fail to satisfy others. We tend to improve the performance of the hardest queries for state-of-the-art neural rankers by attempting to provide high-recall at the first-stage retrieval. We observe that neural rankers struggle to learn suitable representations to connect hard queries to their relevant documents. As such, we propose to learn query-like representations for documents and show that training a dense retriever on the generated alternative document representations would be more effective for connecting queries to documents that would otherwise not be matched. The experiments confirm that our proposed representation $q2q$ is able to retrieve non-overlapping relevant documents compared to the original dense retrievers. Thus, integrating the original dense retriever runs with documents retrieved based on our proposed representation can increase the recall of the first stage retriever by 5% overall on MS MARCO dev set queries and over 10% on the hardest MS MARCO queries.

References

1. 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 and Knowledge Management, pp. 4426–4435 (2021)
2. Bagheri, E., Ensan, F., Al-Obeidat, F.: Impact of document representation on neural ad hoc retrieval. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, pp. 1635–1638. CIKM 2018, Association for Computing Machinery, New York, NY, USA (2018). https://doi.org/10.1145/3269206.3269314
3. Chen, Q., Zhao, H., Li, W., Huang, P., Ou, W.: Behavior sequence transformer for e-commerce recommendation in alibaba (2019)
4. 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, pp. 758–759. SIGIR 2009, Association for Computing Machinery, New York, NY, USA (2009). https://doi.org/10.1145/1571941.1572114
5. Dai, Z., Callan, J.: Context-aware sentence/passage term importance estimation for first stage retrieval. arXiv preprint arXiv:1910.10687 (2019)
6. 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)

7. Gallagher, L., Chen, R.C., Blanco, R., Culpepper, J.S.: Joint optimization of cascade ranking models. In: Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, pp. 15–23. WSDM 2019, Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3289600.3290986
8. Gao, L., Dai, Z., Chen, T., Fan, Z., Van Durme, B., Callan, J.: Complementing lexical retrieval with semantic residual embedding. arXiv preprint arXiv:2004.13969 (2020)
9. Henderson, M.L., et al.: Efficient natural language response suggestion for smart reply. CoRR abs/1705.00652 (2017). http://arxiv.org/abs/1705.00652
10. Jones, K.S., Walker, S., Robertson, S.E.: A probabilistic model of information retrieval: development and comparative experiments: part 2. Inform. Process. Manage. **36**(6), 809–840 (2000)
11. Karpukhin, V., et al.: Dense passage retrieval for open-domain question answering. arXiv preprint arXiv:2004.04906 (2020)
12. Lafferty, J., Zhai, C.: Document language models, query models, and risk minimization for information retrieval. In: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 111–119 (2001)
13. Liu, S., Xiao, F., Ou, W., Si, L.: Cascade ranking for operational e-commerce search. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2017). https://doi.org/10.1145/3097983.3098011
14. MacAvaney, S., Nardini, F.M., Perego, R., Tonellotto, N., Goharian, N., Frieder, O.: Efficient document re-ranking for transformers by precomputing term representations. CoRR abs/2004.14255 (2020). https://arxiv.org/abs/2004.14255
15. Nguyen, T., et al.: Ms marco: a human generated machine reading comprehension dataset. In: CoCo@ NIPs (2016)
16. Nogueira, R., Yang, W., Cho, K., Lin, J.: Multi-stage document ranking with bert. arXiv preprint arXiv:1910.14424 (2019)
17. Nogueira, R., Yang, W., Lin, J., Cho, K.: Document expansion by query prediction. arXiv preprint arXiv:1904.08375 (2019)
18. Reimers, N., Gurevych, I.: Sentence-bert: sentence embeddings using siamese bert-networks. arXiv preprint arXiv:1908.10084 (2019)
19. Sanh, V., Debut, L., Chaumond, J., Wolf, T.: Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. CoRR abs/1910.01108 (2019). http://arxiv.org/abs/1910.01108
20. Schütze, H., Manning, C.D., Raghavan, P.: Introduction to Information Retrieval, vol. 39. Cambridge University Press, Cambridge (2008)
21. Shehata, D., Arabzadeh, N., Clarke, C.L.: Early stage sparse retrieval with entity linking. In: Proceedings of the 31st ACM International Conference on Information and Knowledge Management, pp. 4464–4469 (2022)
22. Singhal, A., et al.: Modern information retrieval: a brief overview. IEEE Data Eng. Bull. **24**(4), 35–43 (2001)
23. Vaswani, A., et al.: Attention is all you need. CoRR abs/1706.03762 (2017). http://arxiv.org/abs/1706.03762
24. Wang, L., Lin, J.J., Metzler, D.: A cascade ranking model for efficient ranked retrieval. In: Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval (2011)
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. Adv. Neural Inform. Process. Syst. **33**, 5776–5788 (2020)
26. Wang, Z., Zhao, L., Jiang, B., Zhou, G., Zhu, X., Gai, K.: Cold: towards the next generation of pre-ranking system (2020)
27. Willett, P.: Combination of similarity rankings using data fusion. J. Chem. Inform. Model. **53**(1), 1–10 (2013)

28. Xiong, L., et al.: Approximate nearest neighbor negative contrastive learning for dense text retrieval. arXiv preprint arXiv:2007.00808 (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. Zhang, H., Abualsaud, M., Ghelani, N., Smucker, M.D., Cormack, G.V., Grossman, M.R.: Effective user interaction for high-recall retrieval: Less is more. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, pp. 187–196 (2018)