Can Query Expansion Improve Generalization of Strong Cross-Encoder Rankers?

Minghan Li
University of Waterloo
Waterloo, Canada
m692li@uwaterloo.ca

Honglei Zhuang
Google
Mountain View, US
hlz@google.com

Kai Hui
Google
Mountain View, US
kaihuibj@google.com

Zhen Qin
Google
New York, US
zhenqin@google.com

Jimmy Lin
University of Waterloo
Waterloo, Canada
jimmylin@uwaterloo.ca

Rolf Jagerman
Google
Amsterdam, Netherlands
jagerman@google.com

Xuanhui Wang
Google
Mountain View, US
xuanhui@google.com

Michael Bendersky
Google
Mountain View, US
bemike@google.com

ABSTRACT

Query expansion has been widely used to improve the search results of first-stage retrievers, yet its influence on second-stage, cross-encoder rankers remains under-explored. A recent study shows that current expansion techniques benefit weaker models but harm stronger rankers. In this paper, we re-examine this conclusion and raise the following question: Can query expansion improve generalization of strong cross-encoder rankers? To answer this question, we first apply popular query expansion methods to different cross-encoder rankers and verify the deteriorated zero-shot effectiveness. We identify two vital steps in the experiment: high-quality keyword generation and minimally-disruptive query modification. We show that it is possible to improve the generalization of a strong neural ranker, by generating keywords through a reasoning chain and aggregating the ranking results of each expanded query via self-consistency, reciprocal rank weighting, and fusion. Experiments on BEIR and TREC Deep Learning 2019/2020 show that the nDCG@10 scores of both MonoT5 and RankT5 following these steps are improved, which points out a direction for applying query expansion to strong cross-encoder rankers.

CCS CONCEPTS

• **Information systems → Retrieval models and ranking**.

KEYWORDS

Query Expansion, Large Language Models, Cross-Encoder Rankers

ACM Reference Format:
Minghan Li, Honglei Zhuang, Kai Hui, Zhen Qin, Jimmy Lin, Rolf Jagerman, Xuanhui Wang, and Michael Bendersky. 2024. Can Query Expansion Improve Generalization of Strong Cross-Encoder Rankers?. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24), July 14–18, 2024, Washington, DC, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3626772.3657979

SIGIR '24, July 14–18, 2024, Washington, DC, USA

ACM ISBN 979-8-4007-0431-4/24/07
https://doi.org/10.1145/3626772.3657979

Methods	DL19	DL20
RankT5$_{\text{base}}$ [49]	0.737	0.736
- w/ RM3 [19]	0.681	0.695
- w/ query2doc [40]	0.572	0.637
- w/ query2keyword [17]	0.690	0.688
- w/ Ours	**0.751**	**0.752**
MonoT5$_{\text{base}}$ [28]	0.695	0.720
- w/ RM3	0.673	0.693
- w/ query2doc	0.473	0.613
- w/ query2keyword	0.672	0.682
- w/ Ours	**0.724**	**0.730**

Table 1: nDCG@10 on TREC DL 2019/2020. Directly applying existing query expansion methods on strong cross-encoder rankers can cause effectiveness deterioration.

1 INTRODUCTION

Query expansion has been a core technique in information retrieval for over half a century [1, 33, 35]. The goal is to increase the retrieval accuracy by adding additional terms to the query. Conventional methods such as RM3 [19] leverage Pseudo-Relevance Feedback (PRF) to select terms from the documents retrieved for the original query as expansions [20]. Recently, large language models (LLMs) demonstrate their effectiveness in generating expansion terms for retrieval, which is known as generative query expansion [17, 40]. However, both methods or their combinations are mainly considered for improving the recall and precision of *first-stage retrievers*, yet their influence on the generalization of *second-stage, cross-encoder rerankers* remains under-explored.

This problem is interesting as additional terms usually contain more information about the query, yet they are rarely used in cross-encoder ranking. A recent study from Weller et al. [44] explores generative query expansion for different retrievers and rankers. They found that weaker models benefit more from expansions while stronger rankers are hurt in most cases. This counter-intuitive observation calls for further examination of current expansion methods and whether there is a way to improve the results of strong rankers using query expansion. To verify the conclusion, we also apply some popular query expansion methods to two state-of-the-art cross-encoder rankers, RankT5 [49] and MonoT5 [28]. As shown in Table 1, the results are consistent with Weller et al. [44] where we observe that the nDCG@10 scores of both rankers on

TREC DL 2019/2020 and BEIR are compromised using either PRF or LLMs. More results are shown in Section 4 and 5.

In this work, we identify two important steps for successfully using query expansion in cross-encoder rankers, which are not well explored by existing work focusing on retrievers: high-quality keyword generation and minimally-disruptive query modification. First, cross-encoder rankers mainly aim to improve precision-related metrics such as nDCG@10 which are sensitive to noisy keywords, while retrievers care more about recall where some low-quality expanded keywords are less influential. Therefore, strong rankers such as RankT5 have higher demand on the generation quality. Second, cross-encoder rankers are heavily based on token interactions, making them sensitive to the distributional shift in queries (e.g., number of tokens, input formats) compared to retrievers. Therefore, inserting documents in a query [40] might be less desirable.

In our experiments, we find that the most effective way is to first use an LLM to generate high-quality, short keywords through a reasoning chain [43]. We then follow self-consistency [42] to run the above process multiple times to filter noisy keywords and select top-k candidates, ensuring the quality of the generated keywords. To mitigate distributional shift in query, we insert each keyword independently with the query and use reciprocal rank weighting [10] to combine the ranking results. Our pipeline manages to improve the nDCG@10 over the baselines for cross-encoder rankers such as RankT5 [49] on both BEIR and TREC DL 2019/2020, while other baselines that have been found effective for retrievers fail to improve such strong rankers. Our study provides a preliminary yet novel research foundation for researchers to explore query expansion for cross-encoder rankers.

2 RELATED WORK

Query Expansion and Fusion. Early research on query expansion concentrated on utilizing either lexical knowledge bases [6, 7, 39] or Pseudo-Relevance Feedback (PRF) [16, 34, 46]. Recent studies show that scaling up LLMs through pre-training with more extensive and higher-quality corpora [8, 14, 29, 30, 38] can result in higher capabilities. Researchers have used large language models for generating keywords in the context of query expansion [9, 17, 40, 41].

The effectiveness of query variant fusion has been proven in previous works. Belkin et al. [3] pioneered the fusion of multiple query variations into a single ranked list. Bailey et al. [2] introduced Rank Biased Centroids for effective query variation fusion. Benham and Culpepper [5] furthered this by applying reciprocal rank fusion [10] and CombSUM [3] with double fusion.

LLM-Based Neural Rankers. MonoBERT [27] stands out as one of first cross-encoders for text reranking tasks. CEDR [25] introduces a more intricate approach by incorporating token representations at all layers of the Transformer using pre-BERT neural rerankers [15]. More potent rankers based on LLMs have emerged to directly score the relevance between queries and documents [28, 45, 49]. Most recently, LLMs have showcased remarkable efficacy when tasked with few/zero-shot text ranking such as LRL [24], RankGPT [36], RG-*k*L [48], RankVicuna [31], and RankLlama [23]. Alternatively, they can perform pairwise comparisons between passages, as demonstrated by PRP [32]. Despite the zero-shot effectiveness, the multiple decoding passes render them slow and non-parallelizable.

3 FRAMEWORK AND IMPLEMENTATION

3.1 Cross-Encoder Ranking

Given a query q, the text retrieval or ranking task is to return a sorted list of documents $\{d_1, d_2, ..., d_k\}$ from a large text corpus C to maximize a metric of interest. In this paper, we assume a set of candidate documents $\{d_1, d_2, ..., d_k\}$ generated by a first-stage retriever are given and focus on the second-stage reranker to reorder the candidate documents. The query-document pairs are encoded together for fine-grained token-level interactions:

$$s(q, d) = \phi(\text{concat}(q, d)), \tag{1}$$

where ϕ is the reranker and s is the similarity score. The "concat" function is implemented using special tokens as indicators, such as "Question: q Document: d". In the following subsection, we will introduce the framework we use for keyword generation and selection to improve the results of cross-encoders.

3.2 High-Quality Keyword Generation

The first step of query expansion is to generate keywords $\{w_1, w_2, ..., w_i\}$ semantically similar to the query q. There are generally two sources of signals: The classical approach which involves corpus-based signals through Pseudo-Relevance Feedback (PRF) or more recent approaches leveraging signals from LLMs by prompting [17]. Notice that some LLM-based methods like Q2D [40] and HyDE [13] generate much longer passages or documents rather than keywords, which drift too much from the distribution of queries and deteriorate cross-encoder reranker effectiveness. In the following components of this framework, we do not consider expansion other than keywords, but will show the results of using excessively long expansion in Section 4. We explore four different methods for keyword generation, including both LLM-based and PRF-based methods, as well as their combinations:

- **PRF-based** methods like RM3 [19] to extract keywords from the retrieved documents $d_1, d_2, ...$.
- **LLM-based** methods to generate keywords $w_1, w_2, ...$ like Q2K [17].
- **PRF + D2K**, which uses an LLM to extract keywords $w_1, w_2, ...$ from the retrieved documents $d_1, d_2, ...$.
- **Q2D2K**, which uses the LLM to generate detailed documents $d_1', d_2', ...$ first and then selects a set of keywords $w_1, w_2, ...$.

PRF + D2K and Q2D2K are inspired by Q2D [40] and HyDE [13]. These methods are problematic when the generated documents are directly used as queries for cross-encoder rankers due to the query distribution shift, but are useful when the generated documents are summarized into keywords for expansions. The assumption is that documents generated by a well pre-trained LLM can already answer the question or at least contain helpful keywords. The intuition is also similar to recitation-augmented language models [37] where more knowledge can be elicited before fulfilling the task.

As mentioned in Section 1, the precision-related metrics, which cross-encoder rankers aim to optimize, are more sensitive to the noise in the expanded keywords. If there are noisy keywords generated from the previous stage, the effectiveness of rankers are more likely to be affected compared to retrievers. Hence, we argue that

Methods	AA	SF	NQ	Fe	Qu	HQ	FQ	CF	BA	SD	T2	TN	S1	DB	NF	R0	TC	BEIR	Wiki+News
RankT5$_{\text{base}}$	0.323	0.750	0.565	0.826	0.814	0.732	**0.414**	0.255	0.541	0.174	0.383	0.439	**0.299**	0.448	0.375	0.525	**0.782**	0.508	0.557
- w/ RM3	**0.330**	0.722	0.546	0.817	**0.831**	0.689	0.377	0.257	0.473	0.165	0.360	0.443	0.291	0.393	0.358	0.481	0.746	0.487	0.539
- w/ query2doc	0.271	0.671	0.511	0.810	0.749	0.640	0.300	0.184	0.358	0.140	0.275	0.419	0.231	0.367	0.331	0.447	0.716	0.437	0.502
- w/ query2keyword	0.315	0.739	0.546	0.841	0.807	0.725	0.381	0.257	0.500	0.168	0.353	**0.470**	0.280	0.398	0.365	0.496	0.741	0.493	0.556
- w/ Ours	0.324	**0.752**	**0.577**	**0.846**	0.822	**0.744**	0.412	**0.261**	**0.542**	**0.176**	**0.390**	0.454	0.292	**0.452**	**0.377**	**0.541**	0.781	**0.514***	**0.570***
MonoT5$_{\text{base}}$	0.242	0.713	0.553	0.808	0.785	0.705	0.364	0.216	0.495	0.161	**0.406**	0.417	0.286	0.423	0.312	0.415	**0.685**	0.470	0.519
- w/ RM3	**0.249**	0.698	0.544	0.817	**0.799**	0.666	0.354	0.219	0.453	0.156	0.390	0.427	**0.291**	0.412	**0.314**	0.396	0.632	0.465	0.511
- w/ query2doc	0.188	0.640	0.508	0.784	0.353	0.594	0.302	0.108	0.353	0.127	0.339	0.398	0.242	0.368	0.293	0.391	0.590	0.392	0.459
- w/ query2keyword	0.232	0.697	0.542	0.828	0.774	0.702	0.341	0.215	0.449	0.156	0.378	**0.453**	0.284	0.397	0.293	0.359	0.642	0.455	0.517
- w/ Ours	0.240	**0.720**	**0.569**	**0.831**	0.793	**0.719**	**0.369**	**0.223**	**0.499**	**0.163**	0.404	0.431	0.288	**0.443**	0.313	**0.420**	0.671	**0.476***	**0.532***

Table 2: nDCG@10 scores on BEIR. TC=TREC-COVID, NF=NFCorpus, NQ=NaturalQuestions, HQ=HotpotQA, FQ=FiQA, AA=ArguAna, T2=Touché-2020, Qu=Quora, DB=DBPedia, SD=SCIDOCS, Fe=FEVER, CF=Climate-FEVER, SF=SciFact, S1=Signal-1M, BA=BioASQ, R0=Robust04, TN=TREC-NEWS. $\ast$: pass the paired t-test against the other baselines ($p < 0.01$).

it is necessary to add a filtering stage in this framework to remove noisy keywords and increase the reliability of the expansion. We leverage self-consistency [42] in LLM literature for filtering. For LLM-based keyword generation methods which involves stochasticity, we repeat the keyword generation method multiple times and select the top-k keywords that have the highest majority votes (i.e., frequency). For deterministic methods like RM3, we simply take the keywords with the highest RM3 keyword weights.

3.3 Minimally-Disruptive Query Modification

One way to insert keywords is to directly concatenate them with the query. The concatenation function we use is “Question: q w_1 w_2 ...w_i Document: d”. However, as mentioned in Section 1, the increasing number of keywords or excessively long expansion may overwhelm the original query, especially for cross-encoder models which rely more on query-document token-interaction. In Section 4, we will show that even increasing the number of keywords to 3 will result in degraded precision. To mitigate the distributional shift in query, another way is to concatenate each keyword individually with the query and fuse the final ranking results together [22]. Inserting only 1 keyword is the minimally-disruptive expansion we found for cross-encoder, which is proved to be very robust on multiple datasets. Specifically, the new similarity scoring function will be:

$$s(q, d) = \sum_i \alpha_i \cdot \phi(\text{concat}(q, w_i, d)), \tag{2}$$

where the concatenation function concat(q, w_i, d) is implemented as “Question: q w_i Document: d” and α_i is the weight for the expansion w_i. We find this formulation more effective than concatenating all keywords at once as the number of keywords increases.

Previous works [4, 47] have also found that ensembling runs from different models or data augmentation can be effective for ranking. After obtaining the candidate keywords, we concatenate each keyword independently with the original query and then feed it in the cross-encoder ranker model to rerank the top-1000 candidate documents retrieved by BM25 to get a ranked list. For the fusion weights in Equation (2), we follow the previous work [10, 12] to weight the ranked lists using the reciprocal rank of the top-1 document in retrieved list: $\alpha_i = \frac{1}{\text{Rank}(d^+, D_i)}$ where Rank(d^+, D_i) is the rank of the top-1 document d^+ retrieved for the original query in a candidate list of expansion w_i. Finally, we combine the ranking list of the original query in case all the expansions are not helpful.

Methods	DL19	DL20	BEIR	Wiki+News
Q2D2K-fusion	**0.751**	**0.752**	**0.514**	**0.570**
Q2K-fusion	0.750	0.748	0.510	0.5628
PRF + D2K-fusion	0.745	0.733	0.510	0.565
RM3-fusion	0.741	0.737	0.510	0.560

Table 3: nDCG@10 score of Q2D2K, Q2K, RM3, and PRF + D2K with fusion based on RankT5.

4 EXPERIMENTS

Models and Datasets. For the keyword generation, we use Flan-PaLM2-S [14]. For cross-encoder ranking, we test two different rankers: MonoT5 [28] and RankT5 [49]. We reproduce the MonoT5 model using the point-wise loss in Zhuang et al. [49]. For in-domain evaluation, we evaluate on TREC DL 2019 and 2020 [11], containing 43 and 54 test queries, respectively. The relevance sets are densely labelled with scores from 0 to 4. For out-of-domain evaluation, we evaluate on 17 datasets from BEIR.

Evaluation. We report the nDCG@10 metric [18] as datasets in BEIR and TREC DL are densely labeled, and the top-10 setting reflects the common use case in applications. As the LLM is fine-tuned on instructions set which has large overlap with the Wikipedia and News corpus in BEIR, we also report the average score on Natural Questions, FEVER, Climate-FEVER, HotpotQA, TREC-News, and Robust04 datasets (Wiki+News) besides the main score.

Inference Pipeline. We use BM25’s top-1000 retrieval results as the candidates for reranking using Pyserini [21]. For RM3, we use Pyterrier [26] to extract keywords from the retrieved documents.

For keywords generation method Q2D2K, we follow the instruction template in Promptagator [12] and ask Flan-PaLM2-S to generate 2 documents based on the query and then extract 5 keywords for each document. We then repeat the process 3 times to obtain 30 keywords. For keyword generation method PRF + D2K, we replace the generated documents by BM25 retrieved documents. By default, we use Q2D2K as the keyword generation component. We then run self-consistency and select the top-3 keywords from the keyword candidates and concatenate each of them individually with the query before feeding them into the ranker.

For fusion, we use the reciprocal ranks of the top-1 document retrieved for the original query in each expansion’s reranked list

Methods	DL19	DL20	BEIR	Wiki+News
RankT5$_{\text{base}}$	0.737	0.736	0.508	0.557
+mean pooling	0.748	0.750	0.514	0.572
+reciprocal rank	0.751	0.751	0.514	**0.573**
+original query	**0.751**	**0.752**	**0.514**	0.570
MonoT5$_{\text{base}}$	0.695	0.720	0.470	0.519
+mean pooling	0.700	0.727	0.471	**0.533**
+reciprocal rank	0.713	**0.733**	0.475	0.532
+original query	**0.724**	0.730	**0.476**	0.532

Table 4: Incremental ablation on the keyword fusion process. Details are introduced in Section 5.

as weights. We finally fuse the aggregated expansion results with the original reranking list from the cross-encoder ranker as regularization, with a coefficient of 0.3. Coefficients range from 0.1 to 0.3 work similarly but not included due to space limit.

Results. Table 1 and 2 show the main results on TREC DL19/20 and BEIR benchmark, where directly applying the query expansion pipeline in retrieval to cross-encoder ranker reranking results in deteriorated effectiveness even with top-3 keyword concatenation, while our method can improve over the original cross-encoder ranker scores on both TREC DL and BEIR. The improvement on TREC DL and Wiki+News BEIR datasets are more significant compared to other datasets in BEIR, which results from the instruction fine-tuning step of PaLM2 as previously mentioned.

5 ABLATION ANALYSIS

RQ1: Whether to use LLMs or PRF for keyword generation? To compare the keyword generation quality of different methods, we fix the keyword insertion and fusion procedure while varying the keyword generation methods. Table 3 shows the comparison results. We can see that the keywords generation quality is reflected in the nDCG@10 scores, where our Q2D2K method manages to outperform other LLM or PRF based methods on DL 19/20 and BEIR, reflecting higher keyword quality. Besides that, fusion is also very important for maintaining cross-encoder ranker zero-shot effectiveness which will be discussed in detail in RQ2.

RQ2: How to use these keyword expansions? Figure 1 plots the number of keywords used for ranking fusion and its influence on the final nDCG@10 score. We can see that the improvement of our proposed method (which uses Reciprocal Rank Weighting) peaks at 3 expanded keywords and gradually diminishes with the addition of more keywords. Although the generated keywords are useful, they still bring noise which escalates as the keyword number increases.

Table 4 shows the ablation of the keyword fusion process, which mainly includes ranking average, reciprocal rank weighting, and combining with original query ranking results. Adding mean fusion and reciprocal weighting consistently brings improvement to the model. As for the original query fusion, we view it as an expansion and combine its ranking with the other expansion's ranking results using a default value of 0.3 for zero-shot evaluation. The improvement is less consistent as we did not perform hyper-parameter search on the fusion coefficient due to zero-shot evaluation, but instead use a default value of 0.3 for combining with the original query ranking results. Figure 1 also shows that fusion is more robust compared to keyword concatenation as the keywords increase.

Methods	DL19	DL20
Mean pooling	0.748	0.750
Reciprocal rank	**0.751**	**0.751**
Top-k overlap	0.741	0.750
Ranker's entropy	0.729	0.746
KL divergence	0.741	0.749
WS distance	0.743	0.751

Table 5: Weighting methods for fusion on DL 19/20 based on RankT5. nDCG@10 scores are reported.

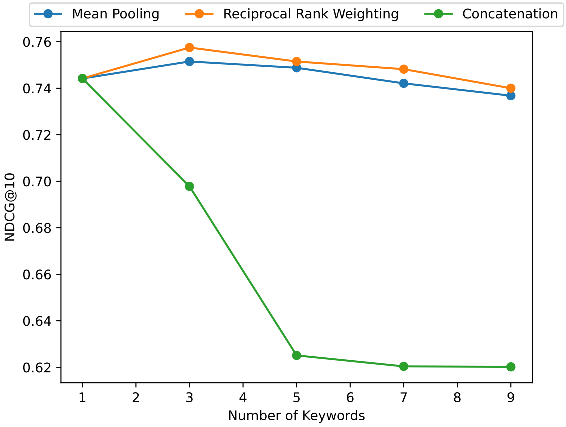

Figure 1: nDCG@10 score on DL19 using different number of keywords for RankT5.

Table 5 shows different fusion methods we tried on TREC DL 19/20. Mean pooling and reciprocal rank weighting are reported in Table 4. For top-k overlap, we take the overlap between the original query's candidate list and the other expansion's candidate lists as weights for α_i in Equation (2). For ranker's entropy, we normalized the retrieved scores into a distribution for each expansion and use the reciprocal entropy of the distribution as weights. For KL divergence and Wasserstein distance, they are similar to the ranker's entropy except that they calculate the distances between the original query's distribution and the other expansion's distributions. We also take the reciprocal of this distance as weights for fusion. We can see that among all the fusion techniques, the reciprocal rank weighting method has the best nDCG@10 scores on both DL 19 and 20, demonstrating the robustness of this simple fusion method.

6 CONCLUSION

In this paper, we examine the possibility of improving the generalization of cross-encoder rankers using query expansion based on the study of Weller et al. [44]. Our solution is to leverage an LLM to generate high-quality, concise keywords through a reasoning chain and individually evaluate the ranking scores of each expansion before aggregating them together. We observe significant improvement on BEIR and TREC DL 2019/2020 over directly using the popular query expansion methods.

REFERENCES

[1] Nasreen Abdul-Jaleel, James Allan, W. Bruce Croft, Fernando Diaz, Leah Larkey, Xiaoyan Li, Mark D. Smucker, and Courtney Wade. 2004. UMass at TREC 2004: Novelty and HARD. *Computer Science Department Faculty Publication Series* (2004), 189.

[2] Peter Bailey, Alistair Moffat, Falk Scholer, and Paul Thomas. 2017. Retrieval Consistency in the Presence of Query Variations. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Shinjuku, Tokyo, Japan) *(SIGIR '17)*. Association for Computing Machinery, New York, NY, USA, 395–404. https://doi.org/10.1145/3077136.3080839

[3] N.J. Belkin, P. Kantor, E.A. Fox, and J.A. Shaw. 1995. Combining the Evidence of Multiple Query Representations for Information Retrieval. *Information Processing and Management* 31, 3 (1995), 431–448. https://doi.org/10.1016/0306-4573(94)00057-A The Second Text Retrieval Conference (TREC-2).

[4] Michael Bendersky, Honglei Zhuang, Ji Ma, Shuguang Han, Keith Hall, and Ryan McDonald. 2020. RRF102: Meeting the TREC-COVID Challenge with a 100+ Runs Ensemble. *arXiv preprint arXiv:2010.00200* (2020).

[5] Rodger Benham and J. Shane Culpepper. 2017. Risk-Reward Trade-Offs in Rank Fusion. In *Proceedings of the 22nd Australasian Document Computing Symposium* (Brisbane, QLD, Australia) *(ADCS '17)*. Association for Computing Machinery, New York, NY, USA, Article 1, 8 pages. https://doi.org/10.1145/3166072.3166084

[6] J. Bhogal, A. Macfarlane, and P. Smith. 2007. A Review of Ontology Based Query Expansion. *Inf. Process. Manage.* 43, 4 (jul 2007), 866–886.

[7] Abdoulahi Boubacar and Zhendong Niu. 2013. Concept Based Query Expansion. In *2013 Ninth International Conference on Semantics, Knowledge and Grids*. 198–201.

[8] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. *arXiv preprint arXiv:2005.14165* (2020).

[9] Vincent Claveau. 2021. Neural Text Generation for Query Expansion in Information Retrieval. In *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*. 202–209.

[10] Gordon V. Cormack, Charles L. A. Clarke, and Stefan Buettcher. 2009. 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* (Boston, MA, USA) *(SIGIR '09)*. Association for Computing Machinery, New York, NY, USA, 758–759. https://doi.org/10.1145/1571941.1572114

[11] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Fernando Campos, and Ellen M. Voorhees. 2020. Overview of the TREC 2020 Deep Learning Track. *arXiv preprint arXiv:2003.07820* (2020).

[12] Zhuyun Dai, Vincent Y Zhao, Ji Ma, Yi Luan, Jianmo Ni, Jing Lu, Anton Bakalov, Kelvin Guu, Keith Hall, and Ming-Wei Chang. 2023. Promptagator: Few-shot Dense Retrieval From 8 Examples. In *The Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=gmL46YMpu2J

[13] Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023. Precise Zero-Shot Dense Retrieval without Relevance Labels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, Toronto, Canada, 1762–1777. https://doi.org/10.18653/v1/2023.acl-long.99

[14] Google, Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, Eric Chu, Jonathan H. Clark, Laurent El Shafey, Yanping Huang, Kathy Meier-Hellstern, Gaurav Mishra, Erica Moreira, Mark Omernick, Kevin Robinson, Sebastian Ruder, Yi Tay, Kefan Xiao, Yuanzhong Xu, Yujing Zhang, Gustavo Hernandez Abrego, Junwhan Ahn, Jacob Austin, Paul Barham, Jan Botha, James Bradbury, Siddhartha Brahma, Kevin Brooks, Michele Catasta, Yong Cheng, Colin Cherry, Christopher A. Choquette-Choo, Aakanksha Chowdhery, Clément Crepy, Shachi Dave, Mostafa Dehghani, Sunipa Dev, Jacob Devlin, Mark Díaz, Nan Du, Ethan Dyer, Vlad Feinberg, Fangxiaoyu Feng, Vlad Fienber, Markus Freitag, Xavier Garcia, Sebastian Gehrmann, Lucas Gonzalez, Guy Gur-Ari, Steven Hand, Hadi Hashemi, Le Hou, Joshua Howland, Andrea Hu, Jeffrey Hui, Jeremy Hurwitz, Michael Isard, Abe Ittycheriah, Matthew Jagielski, Wenhao Jia, Kathleen Kenealy, Maxim Krikun, Sneha Kudugunta, Chang Lan, Katherine Lee, Benjamin Lee, Eric Li, Music Li, Wei Li, YaGuang Li, Jian Li, Hyeontaek Lim, Hanzhao Lin, Zhongtao Liu, Frederick Liu, Marcello Maggioni, Aroma Mahendru, Joshua Maynez, Vedant Misra, Maysam Moussalem, Zachary Nado, John Nham, Eric Ni, Andrew Nystrom, Alicia Parrish, Marie Pellat, Martin Polacek, Alex Polozov, Reiner Pope, Siyuan Qiao, Emily Reif, Bryan Richter, Parker Riley, Alex Castro Ros, Aurko Roy, Brennan Saeta, Rajkumar Samuel, Renee Shelby, Ambrose Slone, Daniel Smilkov, David R. So, Daniel Sohn, Simon Tokumine, Dasha Valter, Vijay Vasudevan, Kiran Vodrahalli, Xuezhi Wang, Pidong Wang, Zirui Wang, Tao Wang, John Wieting, Yuhuai Wu, Kelvin Xu, Yunhan Xu, Linting Xue, Pengcheng Yin, Jiahui Yu, Qiao Zhang, Steven Zheng, Ce Zheng, Weikang Zhou, Denny Zhou, Slav Petrov, and Yonghui Wu. 2023. PaLM 2 Technical Report. *arXiv preprint arXiv:2305.10403* (2023).

[15] Jiafeng Guo, Yixing Fan, Qingyao Ai, and W. Bruce Croft. 2016. A Deep Relevance Matching Model for Ad-hoc Retrieval. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM.

[16] Ayyoob Imani, Amir Vakili, Ali Montazer, and Azadeh Shakery. 2019. Deep Neural Networks for Query Expansion Using Word Embeddings. In *Advances in Information Retrieval: 41st European Conference on IR Research, ECIR 2019, Cologne, Germany, April 14–18, 2019, Proceedings, Part II 41*. Springer, 203–210.

[17] Rolf Jagerman, Honglei Zhuang, Zhen Qin, Xuanhui Wang, and Michael Bendersky. 2023. Query Expansion by Prompting Large Language Models. *arXiv preprint arXiv:2305.03653* (2023).

[18] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems* 20, 4 (2002), 422–446.

[19] Victor Lavrenko and W. Bruce Croft. 2001. Relevance Based Language Models *(SIGIR '01)*. Association for Computing Machinery, New York, NY, USA, 120–127.

[20] Hang Li, Ahmed Mourad, Shengyao Zhuang, Bevan Koopman, and Guido Zuccon. 2023. Pseudo Relevance Feedback with Deep Language Models and Dense Retrievers: Successes and Pitfalls. *ACM Trans. Inf. Syst.* 41, 3, Article 62 (apr 2023), 40 pages.

[21] Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. Pyserini: A Python Toolkit for Reproducible Information Retrieval Research with Sparse and Dense Representations. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Virtual Event, Canada) *(SIGIR '21)*. Association for Computing Machinery, New York, NY, USA, 2356–2362.

[22] Xiaolu Lu, Oren Kurland, J. Shane Culpepper, Nick Craswell, and Ofri Rom. 2019. Relevance Modeling with Multiple Query Variations. In *Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval* (Santa Clara, CA, USA) *(ICTIR '19)*. Association for Computing Machinery, New York, NY, USA, 27–34. https://doi.org/10.1145/3341981.3344224

[23] Xueguang Ma, Liang Wang, Nan Yang, Furu Wei, and Jimmy Lin. 2023. Fine-Tuning LLaMA for Multi-Stage Text Retrieval. *arXiv preprint arXiv:2310.08319* (2023).

[24] Xueguang Ma, Xinyu Zhang, Ronak Pradeep, and Jimmy Lin. 2023. Zero-Shot Listwise Document Reranking with a Large Language Model. *arXiv preprint arXiv:2305.02156* (2023).

[25] Sean MacAvaney, Andrew Yates, Arman Cohan, and Nazli Goharian. 2019. CEDR: Contextualized Embeddings for Document Ranking. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM.

[26] Craig Macdonald, Nicola Tonellotto, Sean MacAvaney, and Iadh Ounis. 2021. PyTerrier: Declarative Experimentation in Python from BM25 to Dense Retrieval. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management* (Virtual Event, Queensland, Australia) *(CIKM '21)*. Association for Computing Machinery, New York, NY, USA, 4526–4533.

[27] Rodrigo Nogueira and Kyunghyun Cho. 2020. Passage Re-ranking with BERT. *arXiv preprint arXiv:1901.04085* (2020).

[28] Rodrigo Nogueira, Zhiying Jiang, Ronak Pradeep, and Jimmy Lin. 2020. Document Ranking with a Pretrained Sequence-to-Sequence Model. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, Online, 708–718. https://doi.org/10.18653/v1/2020.findings-emnlp.63

[29] OpenAI. 2023. GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774* (2023).

[30] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (Eds.), Vol. 35. Curran Associates, Inc., 27730–27744. https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf

[31] Ronak Pradeep, Sahel Sharifymoghaddam, and Jimmy Lin. 2023. RankVicuna: Zero-Shot Listwise Document Reranking with Open-Source Large Language Models. *arXiv preprint arXiv:2309.15088* (2023).

[32] Zhen Qin, Rolf Jagerman, Kai Hui, Honglei Zhuang, Junru Wu, Le Yan, Jiaming Shen, Tianqi Liu, Jialu Liu, Donald Metzler, Xuanhui Wang, and Michael Bendersky. 2024. Large Language Models are Effective Text Rankers with Pairwise Ranking Prompting. In *Findings of the Association for Computational Linguistics: NAACL 2024*.

[33] Stephen E. Robertson and K. Sparck Jones. 1976. Relevance Weighting of Search Terms. *Journal of the American Society for Information science* 27, 3 (1976), 129–146.

[34] Dwaipayan Roy, Debjyoti Paul, Mandar Mitra, and Utpal Garain. 2016. Using Word Embeddings for Automatic Query Expansion. *arXiv preprint arXiv:1606.07608* (2016).

[35] Gerard Salton. 1971. *The SMART retrieval system—experiments in automatic document processing*. Prentice-Hall, Inc.

[36] Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. 2023. Is ChatGPT Good at Search? Investigating Large Language Models as Re-Ranking Agents. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 14918–14937. https://doi.org/10.18653/v1/2023.emnlp-main.923

[37] Zhiqing Sun, Xuezhi Wang, Yi Tay, Yiming Yang, and Denny Zhou. 2023. Recitation-Augmented Language Models. In *The Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=-cqvvvb-NkI

[38] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. *arXiv preprint arXiv:2302.13971* (2023).

[39] Ellen M. Voorhees. 1994. Query Expansion Using Lexical-Semantic Relations. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Dublin, Ireland) *(SIGIR '94)*. Springer-Verlag, Berlin, Heidelberg, 61–69.

[40] Liang Wang, Nan Yang, and Furu Wei. 2023. Query2doc: Query Expansion with Large Language Models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 9414–9423. https://doi.org/10.18653/v1/2023.emnlp-main.585

[41] Xiao Wang, Sean MacAvaney, Craig Macdonald, and Iadh Ounis. 2023. Generative Query Reformulation for Effective Adhoc Search. *arXiv preprint arXiv:2308.00415* (2023).

[42] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In *The Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=1PL1NIMMrw

[43] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (Eds.), Vol. 35. Curran Associates, Inc., 24824–24837. https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf

[44] Orion Weller, Kyle Lo, David Wadden, Dawn Lawrie, Benjamin Van Durme, Arman Cohan, and Luca Soldaini. 2024. When do Generative Query and Document Expansions Fail? A Comprehensive Study Across Methods, Retrievers, and Datasets. In *Findings of the Association for Computational Linguistics: EACL 2024*, Yvette Graham and Matthew Purver (Eds.). Association for Computational Linguistics, St. Julian's, Malta, 1987–2003. https://aclanthology.org/2024.findings-eacl.134

[45] Xinyu Zhang, Minghan Li, and Jimmy Lin. 2023. Improving Out-of-Distribution Generalization of Neural Rerankers with Contextualized Late Interaction. *arXiv preprint arXiv:2302.06589* (2023).

[46] Zhi Zheng, Kai Hui, Ben He, Xianpei Han, Le Sun, and Andrew Yates. 2021. Contextualized Query Expansion via Unsupervised Chunk Selection for Text Retrieval. *Information Processing & Management* 58, 5 (2021), 102672.

[47] Honglei Zhuang, Zhen Qin, Shuguang Han, Xuanhui Wang, Mike Bendersky, and Marc Najork. 2021. Ensemble Distillation for BERT-Based Ranking Models. In *Proceedings of the 2021 ACM SIGIR International Conference on the Theory of Information Retrieval (ICTIR '21)*.

[48] Honglei Zhuang, Zhen Qin, Kai Hui, Junru Wu, Le Yan, Xuanhui Wang, and Michael Bendersky. 2024. Beyond Yes and No: Improving Zero-Shot Pointwise LLM Rankers via Scoring Fine-Grained Relevance Labels. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

[49] Honglei Zhuang, Zhen Qin, Rolf Jagerman, Kai Hui, Ji Ma, Jing Lu, Jianmo Ni, Xuanhui Wang, and Michael Bendersky. 2023. RankT5: Fine-Tuning T5 for Text Ranking with Ranking Losses *(SIGIR '23)*. Association for Computing Machinery, New York, NY, USA, 2308–2313.