GenQREnsemble: Zero-Shot LLM Ensemble Prompting for Generative Query Reformulation

Kaustubh D. Dhole, Eugene Agichtein

Department of Computer Science
Emory University
Atlanta, USA
{kaustubh.dhole, eugene.agichtein}@emory.edu

Abstract. Query Reformulation(QR) is a set of techniques used to transform a user's original search query to a text that better aligns with the user's intent and improves their search experience. Recently, zero-shot QR has been shown to be a promising approach due to its ability to exploit knowledge inherent in large language models. By taking inspiration from the success of ensemble prompting strategies which have benefited many tasks, we investigate if they can help improve query reformulation. In this context, we propose an ensemble based prompting technique, GenQREnsemble which leverages paraphrases of a zero-shot instruction to generate multiple sets of keywords ultimately improving retrieval performance. We further introduce its post-retrieval variant, GenQREnsembleRF to incorporate pseudo relevant feedback. On evaluations over four IR benchmarks, we find that GenQREnsemble generates better reformulations with relative nDCG@10 improvements up to 18% and MAP improvements upto 24% over the previous zero-shot state-of-art. On the MSMarco Passage Ranking task, GenQREnsembleRF shows relative gains of 5% MRR using pseudo-relevance feedback, and 9% nDCG@10 using relevant feedback documents.

Keywords: Query Reformulation · Zero-Shot · Prompting · Relevance Feedback

1 Introduction

Users searching for relevant documents might not always be able to accurately express their information needs in their initial queries. This could result in queries being vague or ambiguous or lacking the necessary domain vocabulary. Query Reformulation (QR) is a set of techniques used to transform a user's original search query to a text that better aligns with the user's intent and improves their search experience. Such reformulation alleviates the vocabulary mismatch problem by expanding the query with related terms or paraphrasing it into a suitable form by incorporating additional context.

Recently, with the success of large language models (LLMs) [5,8], a plethora of QR approaches have been developed. The generative capabilities of LLMs have been exploited to produce novel queries [17], as well as useful keywords to be appended to the users' original queries [14]. By gaining exposure to enormous amounts of text during pre-training, prompting has become a promising avenue for utilizing knowledge inherent in an LLM for the benefit of subsequent downstream tasks [18] especially QR [16,19].

Unlike training or few-shot learning, zero-shot prompting does not rely on any labeled examples. The advantage of a zero-shot approach is the ease with which a standalone generative

model can be used to reformulate queries by prompting a templated piece of instruction along with the original query. Particularly, zero-shot QR can be used to generate keywords by prompting the user's original query along with an instruction that defines the task of query reformulation in natural language like `Generate useful search terms for the given query:'List all the breweries in Austin'.`

However, such a zero-shot prompting approach is still contingent on the exact instruction appearing in the prompt providing plenty of avenues of improvement. While LLMs have been known to vary significantly in performance across different prompts [10,11] and generation settings [20], many natural language tasks have benefited by exploiting such variation via ensembling multiple prompts or generating diverse reasoning paths [25,26,24]. Whether such improvements also transfer to tasks like QR is yet to be determined. In Figure 1, a vast difference is noticed in the keywords generated when the input instruction is altered to a semantically similar variant. We hypothesize that QR might naturally benefit from such variation – An ensemble of zero-shot reformulators with paraphrastic instructions can be tasked to look at the input query in diverse ways so as to elicit different expansions. This work proposes the following contributions:

Instruction	Expansions Generated
Increase the search efficacy by offering beneficial expansion keywords for the query	age goldfish grow outsmart outlive ageing species...
Enhance search outcomes by recommending beneficial expansion terms to supplement the query	Goldfish breed sizes What kind of goldfish grows the fastest...

Fig. 1. Keywords generated for the query ("do goldfish grow") differ drastically when generated from two paraphrastic instructions prompted to `flan-t5-xxl` [12].

- We propose a novel method,**GenQREnsemble** – a zero-shot **Ensemble** based **Gen**erative **Q**uery **R**eformulator which exploits multiple zero-shot instructions for QR to generate a more effective query reformulation than possible with an individual instruction. (Section 3)
- We further introduce an extension **GenQREnsembleRF** to incorporate **R**elevance **F**eedback into the process. (Section 3)
- We evaluate the proposed methods over four standard IR benchmarks, demonstrating significant relative improvements vs. recent state of the art, of up to 18% on nDCG@10 in pre-retrieval settings, and of up to 9% nDCG@10 on post-retrieval (feedback) settings, demonstrating increased generalizability of our approach.

Next, we summarize the prior work to place our contributions in context.

2 Related Work

Query reformulation has been shown to be effective in many settings [1]. It can be done pre-retrieval, or post-retrieval, via incorporating evidence from feedback, obtained either from a user or from top-ranked results in the sparse retrieval setting [2], and in the dense retrieval setting [3,4].

Recently, zero-shot approaches to query reformulation have received considerable attention. Wang et al. [14] design a query reformulator by fine-tuning a sequence-to-sequence transformer, T5 [13] on pairs of raw and transformed queries. Their zero-shot prompting approach uses an instruction-tuned model, FlanT5 [12] to generate keywords for query expansion and incorporating PRF. Jagerman et al. [16] demonstrate LLMs can be more powerful than traditional methods

for query expansion. Mo et al. [22] propose a framework to reformulate conversational search queries using LLMs. Gao et al. [36]'s framework performs retrieval through fake documents generated by prompting LLMs with user queries. Alaofi et al. [45] prompt LLMs with information descriptions to generate query variants.

However, using a single query reformulation can sometimes degrade performance compared to the original query. To address this drawback, prior efforts have incorporated ensemble strategies via keywords from numerous sources or fusing documents from different queries. Gao et al. [40], combine features derived from various translation models to generate better query rewrites. Si et al. [41] perform QR by utilizing multiple external biomedical resources. Hsu and Taksa [42] present a data fusion framework suggesting that diverse query formulations represent distinct evidence sources for inferring document relevance. Later, Mohankumar et al. [39] generated diverse queries by introducing a diversity-driven RL algorithm. For other tasks, recent works demonstrated the benefits of ensemble strategies for prompting LLMs, including self-consistency [24] for arithmetic and common sense tasks, Chain of Verification [23] for improving factuality, and Diverse [25] for question answering. However, zero-shot based ensemble methods for LLM have not been explored for the Query Reformulation task, as we propose in this paper.

3 Proposed Approach: GenQREnsemble

In this section, we describe two variations of our proposed approach, for the pre- and post-retrieval settings. In the pre-retrieval setting, a Query Reformulation R transforms a user's expressed query q_0 into a novel reformulated version q_r to improve retrieval effectiveness for a given search task (e.g., passage or document retrieval). We also consider the post-retrieval setting, wherein the reformulator can incorporate additional contextual information like document or passage-level feedback.

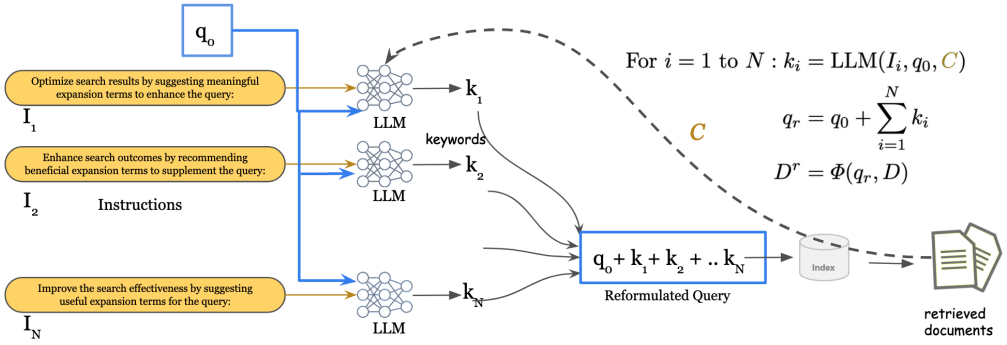

Fig. 2. The complete flow and algorithm shown on the top right.

Pre-retrieval: We propose **GenQREnsemble** – an ensemble prompting based query reformulator which uses N diverse paraphrases of a QR instruction to enhance retrieval. Specifically,

we first use an LLM to paraphrase the instruction I_1 to create N instructions with different surface forms viz. I_1 to I_N. This is required to be done once. Each instruction is then prompted along with the user's query q_0 to generate instruction-specific keywords. All the keywords are then appended to the original query, resulting in a reformulated query, which is then executed against a document index D to retrieve relevant documents D^r. The complete process and algorithm are shown in Figure 2.

Post-retrieval: To assess how well our method can incorporate additional context like document feedback, we introduce **GenQREnsembleRF**. Here, we prepend the N instructions described earlier with a fixed context capturing string "`Based on the given context information {C},`" used[1] in [14] to create their PRF counterparts – where `C` is a space (' ') delimited concatenation of feedback documents $\texttt{C} = d_1 + \ldots + d_m$, obtained either as pseudo-relevance feedback from initial retrieval or manually chosen by the user.

4 Experiments

We now describe the experiments and analysis performed for different retrieval settings.

To instruct the LLM to generate query reformulations, we start with the instruction empirically chosen by Wang et al. [14] – as our base QR instruction I_1. We use this instruction to generate N paraphrases of the instruction ($N = 10$). To this aim, we invoke GPT-3.5 API with the paraphrase generating prompt, namely, I_p=`Generate 10 paraphrases for the following instruction:–` and the base QR instruction I_1 to obtain I_2 to I_{10}. These paraphrases serve as our instruction set for subsequent experiments.

#	Instruction
1	Improve the search effectiveness by suggesting expansion terms for the query
2	Recommend expansion terms for the query to improve search results
3	Improve the search effectiveness by suggesting useful expansion terms for the query
4	Maximize search utility by suggesting relevant expansion phrases for the query
5	Enhance search efficiency by proposing valuable terms to expand the query
6	Elevate search performance by recommending relevant expansion phrases for the query
7	Boost the search accuracy by providing helpful expansion terms to enrich the query
8	Increase the search efficacy by offering beneficial expansion keywords for the query
9	Optimize search results by suggesting meaningful expansion terms to enhance the query
10	Enhance search outcomes by recommending beneficial expansion terms to supplement the query

Fig. 3. Reformulation instructions generated (N=10).

For generating the actual query reformulations, we employ `flan-t5-xxl` [12], an instruction-tuned model. The FlanT5 set of models is created by fine-tuning the text-to-text transformer model, `T5` [13] on instruction data of a variety of NL tasks. We use the checkpoint[2] provided through HuggingFace's Transformers library [6]. Nucleus sampling is performed with a cutoff probability of 0.92 keeping the top 200 tokens (top_k) and a repetition penalty of 1.2.

For evaluation, we use four popular benchmarks through IRDataset [15]'s interface: 1)**TP19**: TREC 19 Passage Ranking which uses the MSMarco dataset [21,16] consisting of search engine queries. 2)**TR04**: TREC Robust 2004 Track, a task intended for testing poorly performing topics. In our experiments, we use the Title as our choice of query. And two tasks from the BEIR [27] benchmark 3)**WT**: Webis Touche [30] for argument retrieval 4)**DE**: DBPedia Entity Retrieval [28].

[1] We found prepending the string in the prompt performs better than appending it at the end

[2] https://huggingface.co/google/flan-t5-xxl

4.1 Baselines:

We compare our work against the following using the Pyterrier [7] framework. For all the post-retrieval experiments, we use 5 documents as feedback.
With BM25 Retriever:

- BM25: Here, we retrieve using raw queries without any reformulation
- FlanQR [14]: We implement Wang et al's single-instruction zero-shot QR [14] which is also a specific case of GenQREnsemble when N=1
- BM25+RM3 [33]: BM25 retrieval with RM3 expanded queries (#feedback terms=10)
- BM25+FlanPRF [14]: BM25 retrieval with FlanPRF expanded queries

With Neural Reranking: Here, we re-evaluate the above settings in conjunction with a MonoT5 neural reranker [43] with all other parameters constant.

- BM25+MonoT5: BM25 retrieval using raw queries, re-ranked with MonoT5 model [43]
- FlanQR+MonoT5: BM25 retrieval with FlanQR reformulations, re-ranked with MonoT5 model
- BM25+RM3+MonoT5: BM25 retrieval with RM3 expanded queries, re-ranked with MonoT5 model
- BM25+FlanPRF+MonoT5: BM25 retrieval with FlanPRF expanded queries, re-ranked with MonoT5 model

5 Results and Analysis

We now report the results of query reformulation for pre- and post-retrieval settings.

5.1 Pre-Retrieval Performance

We first compare the retrieval performances of raw queries and reformulations from FlanQR, and GenQREnsemble in Table 1.GenQREnsemble outperforms the raw queries as well as generates better reformulations than FlanQR's reformulated queries across all the four benchmarks over a BM25 retriever, indicating the usefulness of paraphrasing initial instructions. On TP19, nDCG@10 and MAP improve significantly with relative improvements of 18% and 24% respectively. This is further validated through the querywise analysis shown in Figure 4 – Relative to BM25, nDCG@10 scores of GenQREnsemble (shown in green) are overall better than FlanQR (shown in blue).GenQREnsemble seems more robust too as it avoids drastic degradation in at least 6 queries on which FlanQR fails.

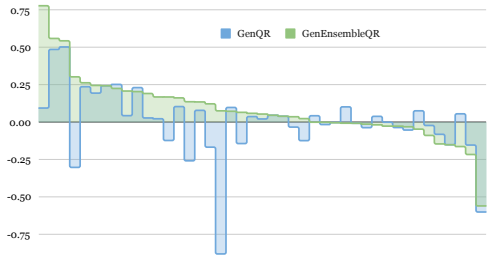

Fig. 4. nDCG@10 Scores of GenQREnsemble and FlanQR relative to BM25

We further look at GenQREnsemble under the neural reranker setting shown at the bottom half of Table 1. In three of the four settings, viz., TP19, WT, and DE, GenQREnsemble is preferable to its zero-shot variant, FlanQR. Evidently, the gains of both the zero-shot approaches in the

Table 1. Performance of GenQREnsemble on the four benchmarks. α denotes significant improvements (paired t-test with Holm-Bonferroni correction, $p < 0.05$) over FlanQR. +% indicates % improvements relative to FlanQR (as whole numbers).

Evaluation Set	TREC Passage 19			TREC Robust 04			Webis Touche			DBpedia Entity		
Setting	nDCG@10	MAP	MRR	P@10	nDCG@10	MRR	nDCG@10	MAP	MRR	nDCG@10	MAP	MRR
BM25	.480	.286	.642	.426	.434	.154	.260	.206	.454	.321	.168	.297
FlanQR	.477	.302	.593	.473	.483	.151	.315	.241	.511	.342	.196	.345
$\text{FlanQR}_{\beta=.05}$	.511	.323	.621	.469	.477	.150	.276	.221	.476	.353	.188	.339
GenQREnsemble	**.564**$^{\alpha}$+18%	**.375**$^{\alpha}$+24%	**.706**+19%	**.500**$^{\alpha}$+6%	**.513**$^{\alpha}$+6%	**.159**+6%	**.317**+1%	**.257**+6%	**.555**+9%	**.374**$^{\alpha}$+9%	**.212**$^{\alpha}$+8%	**.376**$^{\alpha}$+9%
$\text{GenQREnsemble}_{\beta=.05}$	**.575**$^{\alpha}$	**.377**$^{\alpha}$	**.714**	**.502**$^{\alpha}$	**.512**$^{\alpha}$	**.159**	.292	**.242**	.489	**.377**$^{\alpha}$	**.212**$^{\alpha}$	**.380**$^{\alpha}$
BM25+MonoT5	.718	.477	**.881**	**.492**	**.513**	**.173**	**.299**	.216	.525	.414	.249	.444
FlanQR+MonoT5	.707	.486	.847	.490	.510	.170	.292	.215	.530	.415	.255	.446
GenQREnsemble+MonoT5	**.722**+2%	**.503**+3%	.862+2%	.484-1%	.506-1%	.170	.298+3%	**.219**+2%	**.548**+3%	**.420**+1%	**.258**+1%	**.450**+1%

traditional setting are stronger vis-à-vis the neural setting. We hypothesize this could be due to GenQREnsemble and FlanQR both expanding the query via incorporating semantically similar but lexically different keywords. Comparatively, neural models are adept at capturing notions of semantic similarity and might benefit less with query expansion. This also is in line with Weller et al.'s [44] recent analysis on the non-ensemble variant.

5.2 Post-Retrieval Performance

Table 2. Comparison of PRF performance on the TREC 19 Passage Ranking Task

	With BM25 Retriever				With Neural Reranking			
Setting	nDCG@10	nDCG@20	MAP	MRR	nDCG@10	nDCG@20	MAP	MRR
BM25	.480	.473	.286	.642	.718	.696	.477	.881
RM3	.504	.496	.311	.595	.716	.699	.480	.858
FlanPRF	.576	.553	.363	.715	.722	.703	.486	.874
GenQREnsembleRF	**.585**+2%	**.560**+1%	**.373**+3%	**.753**+5%	**.729**+1%	**.706**+1%	**.501**+3%	**.894**+2%
FlanPRF (Oracle)	.753	.728	.501	.936	.742	.734	.545	.881
GenQREnsembleRF (Oracle)	**.820**$^{\alpha}$+9%	**.773**+6%	**.545**+9%	**.977**+4%	**.756**+2%	**.751**+2%	**.545**	**.897**+2%

We now investigate if GenQREnsembleRF can effectively incorporate PRF in Table 2. We find that GenQREnsembleRF improves retrieval performance as compared to other PRF approaches and is able to incorporate feedback from a BM25 retriever better than RM3 as well as its zero-shot counterpart. To assess if GenQREnsembleRF and FlanPRF can at all benefit from incorporating relevant documents, we perform oracle testing by providing the highest relevant gold documents as context. We find that GenQREnsembleRF is able to improve over GenQREnsemble (without feedback) showing that it is able to capture context effectively as well as benefit from it. Further, it can incorporate relevant feedback better than its single-instruction counterpart FlanPRF. We notice improvements even under the neural reranker setting as GenQREnsembleRF outperforms RM3 and FlanPRF. Besides, the oracle improvements are higher with only a BM25 retriever as compared to when a neural reranker is introduced.

We further evaluate the effect of varying the number of feedback documents from 0 to 5. We notice that resorting to an ensemble approach is highly beneficial. In the BM25 setting, the ensemble approach seems always preferable. Under the neural reranker setting too,GenQREnsembleRF almost always outperforms FlanPRF.

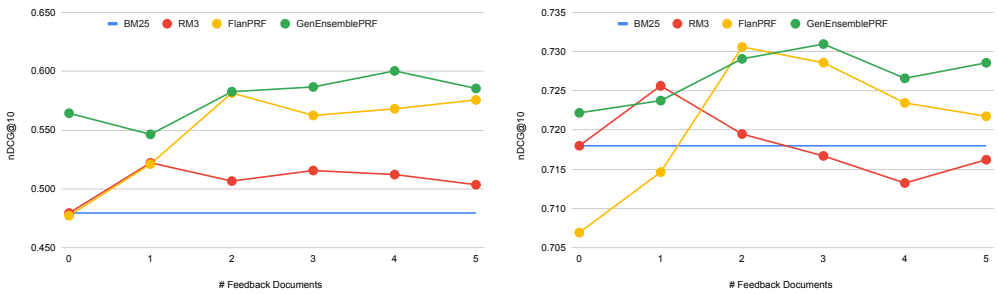

Fig. 5. Effect of feedback documents under sparse (BM25) and neural (MonoT5) rankers

6 Conclusions

Zero-shot QR is advantageous since it does not rely on any labeled relevance judgements and allows eliciting pre-trained knowledge in the form of keywords by prompting the model with the original query and appropriate instruction. By introducing GenQREnsemble, we show that zero-shot performance can be further enhanced by using multiple views of the initial instruction. We also show that the extension GenQREnsembleRF is able to effectively incorporate relevance feedback, either automated or from users. While generative QR greatly benefits from our ensemble approach, the proposed methods come at a cost of potentially increased latency, but this is becoming less problematic with the increased availability of batch inference for LLMs. The proposed ensemble approach could also be applied to other settings, for example, to address different aspects of queries or metrics to optimize, or to better control the generated reformulations.

References

1. Claudio Carpineto and Giovanni Romano. 2012. A Survey of Automatic Query Expansion in Information Retrieval. ACM Comput. Surv. 44, 1, Article 1 (January 2012), 50 pages. https://doi.org/10.1145/2071389.2071390
2. 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 (July 2023), 40 pages. https://doi.org/10.1145/3570724
3. Xiao Wang, Craig MacDonald, Nicola Tonellotto, and Iadh Ounis. 2023. ColBERT-PRF: Semantic Pseudo-Relevance Feedback for Dense Passage and Document Retrieval. ACM Trans. Web 17, 1, Article 3 (February 2023), 39 pages. https://doi.org/10.1145/3572405
4. HongChien Yu, Chenyan Xiong, and Jamie Callan. 2021. Improving Query Representations for Dense Retrieval with Pseudo Relevance Feedback. In Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM '21). Association for Computing Machinery, New York, NY, USA, 3592–3596. https://doi.org/10.1145/3459637.3482124
5. Brown, Tom, et al. "Language models are few-shot learners." Advances in neural information processing systems 33 (2020): 1877-1901.
6. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M. and Davison, J., 2020, October. Transformers: State-of-the-art natural language processing.

In Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations (pp. 38-45).
7. Macdonald, C., Tonellotto, N., MacAvaney, S. and Ounis, I., 2021, October. PyTerrier: Declarative experimentation in Python from BM25 to dense retrieval. In Proceedings of the 30th acm international conference on information and knowledge management (pp. 4526-4533).
8. Peng B, Li C, He P, Galley M, Gao J. Instruction tuning with gpt-4. arXiv preprint arXiv:2304.03277. 2023 Apr 6.
9. Craswell, N., Mitra, B., Yilmaz, E., Campos, D. and Voorhees, E.M., Overview of the TREC 2019 Deep Learning Track.
10. Zhao, Z., Wallace, E., Feng, S., Klein, D. and Singh, S., 2021, July. Calibrate before use: Improving few-shot performance of language models. In International Conference on Machine Learning (pp. 12697-12706). PMLR.
11. Dhole, K., Gangal, V., Gehrmann, S., Gupta, A., Li, Z., Mahamood, S., Mahadiran, A., Mille, S., Shrivastava, A., Tan, S. and Wu, T., 2023. NL-Augmenter: A Framework for Task-Sensitive Natural Language Augmentation. Northern European Journal of Language Technology, 9(1).
12. Chung HW, Hou L, Longpre S, Zoph B, Tay Y, Fedus W, Li E, Wang X, Dehghani M, Brahma S, Webson A. Scaling instruction-finetuned language models. arXiv preprint arXiv:2210.11416. 2022 Oct 20.
13. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W. and Liu, P.J., 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. The Journal of Machine Learning Research, 21(1), pp.5485-5551.
14. Wang, Xiao, Sean MacAvaney, Craig Macdonald, and Iadh Ounis. "Generative Query Reformulation for Effective Adhoc Search.". The First Workshop on Generative Information Retrieval, SIGIR 2023
15. Sean MacAvaney, Andrew Yates, Sergey Feldman, Doug Downey, Arman Cohan, and Nazli Goharian. 2021. Simplified Data Wrangling with ir_datasets. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21). Association for Computing Machinery, New York, NY, USA, 2429–2436. https://doi.org/10.1145/3404835.3463254
16. Jagerman, R., Zhuang, H., Qin, Z., Wang, X. and Bendersky, M., 2023. Query Expansion by Prompting Large Language Models. arXiv preprint arXiv:2305.03653.
17. Nogueira, R., Lin, J. and Epistemic, A.I., 2019. From doc2query to docTTTTTquery. Online preprint, 6, p.2.
18. Srivastava, A., Rastogi, A., Rao, A., Shoeb, A.A.M., Abid, A., Fisch, A., Brown, A.R., Santoro, A., Gupta, A., Garriga-Alonso, A. and Kluska, A., 2023. Beyond the Imitation Game: Quantifying and extrapolating the capabilities of language models. Transactions on Machine Learning Research.
19. Weller, O., Lo, K., Wadden, D., Lawrie, D., Van Durme, B., Cohan, A. and Soldaini, L., 2023. When do Generative Query and Document Expansions Fail? A Comprehensive Study Across Methods, Retrievers, and Datasets. arXiv preprint arXiv:2309.08541.
20. Wiher, G., Meister, C. and Cotterell, R., 2022. On decoding strategies for neural text generators. Transactions of the Association for Computational Linguistics, 10, pp.997-1012.
21. Nguyen, T., Rosenberg, M., Song, X., Gao, J., Tiwary, S., Majumder, R. and Deng, L., 2016. Ms marco: A human-generated machine reading comprehension dataset.
22. Fengran Mo, Kelong Mao, Yutao Zhu, Yihong Wu, Kaiyu Huang, and Jian-Yun Nie. 2023. ConvGQR: Generative Query Reformulation for Conversational Search. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 4998–5012, Toronto, Canada. Association for Computational Linguistics.
23. Dhuliawala, S., Komeili, M., Xu, J., Raileanu, R., Li, X., Celikyilmaz, A. and Weston, J., 2023. Chain-of-verification reduces hallucination in large language models. arXiv preprint arXiv:2309.11495.
24. Wang, X., Wei, J., Schuurmans, D., Le, Q.V., Chi, E.H., Narang, S., Chowdhery, A. and Zhou, D., 2023. Self-Consistency Improves Chain of Thought Reasoning in Language Models.

25. Li, Y., Lin, Z., Zhang, S., Fu, Q., Chen, B., Lou, J.G. and Chen, W., 2023, July. Making language models better reasoners with step-aware verifier. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 5315-5333).
26. Arora, S., Narayan, A., Chen, M.F., Orr, L., Guha, N., Bhatia, K., Chami, I. and Re, C., 2022, September. Ask Me Anything: A simple strategy for prompting language models. In The Eleventh International Conference on Learning Representations.
27. Thakur, N., Reimers, N., Rücklé, A., Srivastava, A. and Gurevych, I., BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models.
28. Hasibi, F., Nikolaev, F., Xiong, C., Balog, K., Bratsberg, S.E., Kotov, A. and Callan, J., 2017, August. DBpedia-entity v2: a test collection for entity search. In Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 1265-1268).
29. Voorhees, E.M., 2005, June. The TREC robust retrieval track. In ACM SIGIR Forum (Vol. 39, No. 1, pp. 11-20). New York, NY, USA: ACM.
30. Bondarenko, A., Fröbe, M., Beloucif, M., Gienapp, L., Ajjour, Y., Panchenko, A., Biemann, C., Stein, B., Wachsmuth, H., Potthast, M. and Hagen, M., 2020. Overview of Touché 2020: argument retrieval. In Experimental IR Meets Multilinguality, Multimodality, and Interaction: 11th International Conference of the CLEF Association, CLEF 2020, Thessaloniki, Greece, September 22–25, 2020, Proceedings 11 (pp. 384-395). Springer International Publishing.
31. Voorhees, E., Alam, T., Bedrick, S., Demner-Fushman, D., Hersh, W.R., Lo, K., Roberts, K., Soboroff, I. and Wang, L.L., 2021, February. TREC-COVID: constructing a pandemic information retrieval test collection. In ACM SIGIR Forum (Vol. 54, No. 1, pp. 1-12). New York, NY, USA: ACM.
32. Amati, G. and Van Rijsbergen, C.J., 2002. Probabilistic models of information retrieval based on measuring the divergence from randomness. ACM Transactions on Information Systems (TOIS), 20(4), pp.357-389.
33. Abdul-Jaleel, N., Allan, J., Croft, W.B., Diaz, F., Larkey, L., Li, X., Smucker, M.D. and Wade, C., 2004. UMass at TREC 2004: Novelty and HARD. Computer Science Department Faculty Publication Series, p.189.
34. Harman, D., 1992. Evaluation Issues in Information Retrieval. Information Processing and Management, 28(4), pp.439-40.
35. Järvelin, K. and Kekäläinen, J., 2002. Cumulated gain-based evaluation of IR techniques. ACM Transactions on Information Systems (TOIS), 20(4), pp.422-446.
36. 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), pages 1762–1777, Toronto, Canada. Association for Computational Linguistics.
37. Paulus, R., Xiong, C. and Socher, R., 2018, February. A Deep Reinforced Model for Abstractive Summarization. In International Conference on Learning Representations.
38. Klein, G., Kim, Y., Deng, Y., Senellart, J. and Rush, A.M., 2017, July. OpenNMT: Open-Source Toolkit for Neural Machine Translation. In Proceedings of ACL 2017, System Demonstrations (pp. 67-72).
39. Mohankumar, A.K., Begwani, N. and Singh, A., 2021, August. Diversity driven query rewriting in search advertising. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (pp. 3423-3431).
40. Gao, J., Xie, S., He, X. and Ali, A., 2012, July. Learning lexicon models from search logs for query expansion. In Proceedings of EMNLP.
41. Si, L., Lu, J. and Callan, J., 2006, November. Combining Multiple Resources, Evidences and Criteria for Genomic Information Retrieval. In TREC.
42. Frank Hsu, D. and Taksa, I., 2005. Comparing rank and score combination methods for data fusion in information retrieval. Information retrieval, 8(3), pp.449-480.
43. Pradeep, R., Nogueira, R. and Lin, J., 2021. The expando-mono-duo design pattern for text ranking with pretrained sequence-to-sequence models. arXiv preprint arXiv:2101.05667.

44. Weller, O., Lo, K., Wadden, D., Lawrie, D., Van Durme, B., Cohan, A. and Soldaini, L., 2023. When do Generative Query and Document Expansions Fail? A Comprehensive Study Across Methods, Retrievers, and Datasets. arXiv preprint arXiv:2309.08541.
45. Marwah Alaofi, Luke Gallagher, Mark Sanderson, Falk Scholer, and Paul Thomas. 2023. Can Generative LLMs Create Query Variants for Test Collections? An Exploratory Study. In Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '23). Association for Computing Machinery, New York, NY, USA, 1869–1873. https://doi.org/10.1145/3539618.3591960