Query Expansion by Prompting Large Language Models

Rolf Jagerman
Google Research
jagerman@google.com

Honglei Zhuang
Google Research
hlz@google.com

Zhen Qin
Google Research
zhenqin@google.com

Xuanhui Wang
Google Research
xuanhui@google.com

Michael Bendersky
Google Research
bemike@google.com

ABSTRACT

Query expansion is a widely used technique to improve the recall of search systems. In this paper, we propose an approach to query expansion that leverages the generative abilities of Large Language Models (LLMs). Unlike traditional query expansion approaches such as Pseudo-Relevance Feedback (PRF) that relies on retrieving a good set of pseudo-relevant documents to expand queries, we rely on the generative and creative abilities of an LLM and leverage the knowledge inherent in the model. We study a variety of different prompts, including zero-shot, few-shot and Chain-of-Thought (CoT). We find that CoT prompts are especially useful for query expansion as these prompts instruct the model to break queries down step-by-step and can provide a large number of terms related to the original query. Experimental results on MS-MARCO and BEIR demonstrate that query expansions generated by LLMs can be more powerful than traditional query expansion methods.

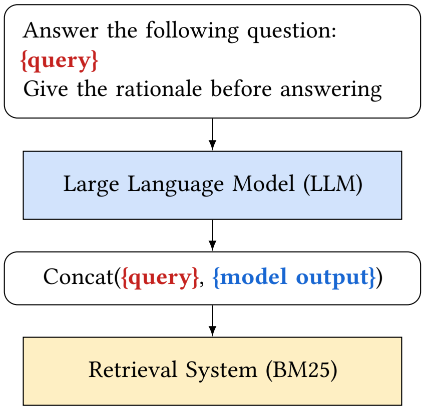
Answer the following question:
{query}
Give the rationale before answering
Large Language Model (LLM)
Concat({query}, {model output})
Retrieval System (BM25)

Figure 1: High-level overview of using a zero-shot Chain-of-Thought (CoT) prompt to generate query expansion terms.

1 INTRODUCTION

Query expansion is a widely used technique that improves the recall of search systems by adding additional terms to the original query. The expanded query may be able to recover relevant documents that had no lexical overlap with the original query. Traditional query expansion approaches are typically based on Pseudo-Relevance Feedback (PRF) [1, 20, 21, 23], which treats the set of retrieved documents from the original query as "pseudo-relevant" and uses those documents' contents to extract new query terms. However, PRF-based approaches assume that the top retrieved documents are relevant to the query. In practice the initial retrieved documents may not be perfectly aligned with the original query, especially if the query is short or ambiguous. As a result, PRF-based approaches may fail if the initial set of retrieved documents is not good enough.

In this paper we propose the use of Large Language Models (LLMs) [3, 8, 19] to aid in query expansion. LLMs have seen a growing interest in the Information Retrieval (IR) community in recent years. They exhibit several properties, including the ability to answer questions and generate text, that make them powerful tools. We propose using those generative abilities to generate useful query expansions. In particular we investigate ways to prompt an LLM and have it generate a variety of alternative and new terms for the original query. This means that, instead of relying on the knowledge within PRF documents or lexical knowledge bases, we rely on the knowledge inherent in the LLM. An example of the proposed methodology is presented in Figure 1.

Our main contributions in this work are as follows: First, we formulate various prompts to perform query expansion (zero-shot, few-shot and CoT) with and without PRF to study their relative performance. Second, we find that Chain-of-Thought (CoT) prompts perform best and hypothesize that this is because CoT prompts instruct the model to break its answer down step-by-step which includes many keywords that can aid in query expansion. Finally, we study the performance across various model sizes to better understand the practical capabilities and limitations of an LLM approach to query expansion.

2 RELATED WORK

Query expansion is widely studied [4, 11]. At its core, query expansion helps retrieval systems by expanding query terms into new terms that express the same concept or information need, increasing the likelihood of a lexical match with documents in the corpus. Early works on query expansion focused on either using lexical knowledge bases [2, 18, 29] or Pseudo-Relevance Feedback (PRF) [1, 20, 23]. PRF-based approaches are particularly useful in practice because they do not need to construct a domain-specific knowledge base and can be applied to any corpus. Orthogonal to query expansion is *document expansion* [10, 16, 25, 33] which applies similar techniques but expands document terms during indexing instead of query terms during retrieval.

Recent works on query expansion have leveraged neural networks to generate or select expansion terms [13, 24, 33, 34], generally by either training or fine-tuning a model. In contrast, our work leverages the abilities inherent in *general-purpose* LLMs without needing to train or fine-tune the model.

We note that our work is similar to the recent works of [7] and [31]: leveraging an LLM to expand a query. However, we differentiate our work in several important ways: First, we study a number of different prompts whereas [31] focuses on a single few-shot prompt and [7] does not study prompts. Second, unlike [31] and [7], we focus on generating *query expansion terms* instead of entire pseudo documents. To this end, we demonstrate the performance of our prompts on a variety of *smaller* model sizes which helps understand both the limitations and the practical capabilities of an LLM approach to query expansion. Finally, we experiment with entirely open-source models, inviting reproducibility and openness of research, while [31] experiments with a single type of model which is only accessible through a third-party API.

3 METHODOLOGY

We formulate the query expansion problem as follows: given a query q we wish to generate an *expanded query* q' that contains additional query terms that may help in retrieving relevant documents. In particular we study the use of an LLM to expand the query terms and generate a new query q'. Since the LLM output may be verbose, we repeat the original query terms 5 times to upweigh their relative importance. This is the same as the trick employed by [31]. More formally:

$$q' = \text{Concat}(q, q, q, q, q, \text{LLM}(prompt_q)), \tag{1}$$

where Concat is the string concatenation operator, q is the original query, LLM is a Large Language Model and $prompt_q$ is the generated prompt based on the query (and potentially side information like few-shot examples or PRF documents).

In this paper we study eight different prompts:

Q2D The Query2Doc [31] few-shot prompt, asking the model to write a passage that answers the query.

Q2D/ZS A zero-shot version of **Q2D**.

Q2D/PRF A zero-shot prompt like **Q2D/ZS** but which also contains extra context in the form of top-3 retrieved PRF documents for the query.

Q2E Similar to the Query2Doc few-shot prompt but with examples of query *expansion terms* instead of *documents*.

Q2E/ZS A zero-shot version of **Q2E**.

Q2E/PRF A zero-shot prompt like **Q2E/ZS** but with extra context in the form of PRF documents like **Q2D/PRF**.

CoT A zero-shot Chain-of-Thought prompt which instructs the model to provide rationale for its answer.

CoT/PRF A prompt like **CoT** but which also contains extra context in the form of top-3 retrieved PRF documents for the query.

Zero-shot prompts (**Q2D/ZS** and **Q2E/ZS**) are the simplest as they consist of a simple plaintext instruction and the input query. Few-shot prompts (**Q2D** and **Q2E**) additionally contain several examples to support in-context learning, for example they contain queries and corresponding expansions. Chain-of-Thought (**CoT**) prompts formulate their instruction to obtain a more verbose output from the model by asking it to break its response down step-by-step. Finally, Pseudo-Relevance Feedback (·/**PRF**) variations of prompts use the top-3 retrieved documents as additional context for the model. See Appendix A for the exact prompts that are used in the experiments.

4 EXPERIMENTS

To validate the effectiveness of the LLM-based query expansion we run experiments on two retrieval tasks: MS-MARCO [15] passage retrieval and BEIR [27]. For the retrieval system we use BM25 [21, 22] as implemented by Terrier [17][1]. We use the default BM25 parameters ($b = 0.75$, $k_1 = 1.2$, $k_3 = 8.0$) provided by Terrier.

4.1 Baselines

To analyze the LLM-based query expansion methods we compare against several classical PRF-based query expansion methods [1]:

- Bo1: Bose-Einstein 1 weighting
- Bo2: Bose-Einstein 2 weighting
- KL: Kullback-Leibler weighting

The implementations for these are provided by Terrier. In all cases we use the default Terrier settings for query expansion: 3 PRF docs and 10 expansion terms.

Furthermore, we include the prompt from Query2Doc [31] as a baseline. However, we do not compare against their exact setup since they use a significantly larger model than the models we study in this paper. The comparisons in this paper are focused on prompts and not on the exact numbers produced by different, potentially much larger, models. Furthermore, for models with a small receptive field (specifically the Flan-T5 models) we only use a 3-shot Q2D prompt instead of the standard 4-shot prompt to prevent the prompt from being truncated.

4.2 Language Models

We compare the prompts on two types of models, Flan-T5 [6, 19] and Flan-UL2 [26], at various model sizes:

- Flan-T5-Small (60M parameters)
- Flan-T5-Base (220M parameters)
- Flan-T5-Large (770M parameters)
- Flan-T5-XL (3B parameters)
- Flan-T5-XXL (11B parameters)
- Flan-UL2 (20B parameters)

We choose to use the Flan [6, 32] versions of the T5 [19] and UL2 [26] models as they are fine-tuned to follow instructions which is critical when using prompt-based approaches. Furthermore, all of these models are available as open-source[2].

4.3 Metrics

Since we are interested in query expansion, which is largely focussed on improving the recall of first-stage retrieval, we use Recall@1K as our core evaluation metric. We also report top-heavy ranking metrics using MRR@10 [30] and NDCG@10 [14] to better understand how the models change the top retrieved results. We report all our results with significance testing using a paired t-test and consider a result significant at $p < 0.01$.

[1] http://terrier.org/

[2] Models are available at https://huggingface.co/docs/transformers/model_doc/flan-t5 and https://huggingface.co/google/flan-ul2

5 RESULTS

5.1 MS-MARCO Passage Ranking

Table 1 presents the results on the MS-MARCO passage ranking task. The classical query expansion baselines (Bo1, Bo2 and KL), already provide a useful gain in terms of Recall@1K over the standard BM25 retrieval. In line with the results of [12], we observe that this increase in recall comes at the cost of top-heavy ranking metrics such as MRR@10 and NDCG@10.

Next, we see the results of LLM-based query expansion depend heavily on the type of prompts used. Similar to the findings of [31], the Query2Doc prompt (**Q2D**) can provide a substantial gain in terms of Recall@1K over the classical approaches. Interestingly, Query2Doc does not only improve recall, but also improves the top-heavy ranking metrics such as MRR@10 and NDCG@10, providing a good improvement across metrics. This contrasts with classical query expansion methods which typically sacrifice top-heavy ranking metrics in order to improve recall.

Finally, the best performance is obtained by **CoT** (and the corresponding PRF-enhanced prompt **CoT/PRF**). This particular prompt instructs the model to generate a verbose explanation by breaking its answer down into steps. We hypothesize that this verbosity may lead to many potential keywords that are useful for query expansion. Finally, we find that adding PRF documents to the prompt helps significantly in top-heavy ranking metrics like MRR@10 and NDCG@10 across models and prompts. A possible explanation for this is that LLMs are effective in distilling the PRF documents, which may already contain relevant passages, by attending over the most promising keywords and using them in the output. We provide a more concrete example of the prompt output in Appendix B.

5.2 BEIR

The BEIR datasets comprise many different zero-shot information retrieval tasks from a variety of domains. We compare the performance of the different prompts on the BEIR datasets in Table 2. The first thing to observe here is that the classical PRF-based query expansion baselines still work very well, especially on domain-specific datasets such as trec-covid, scidocs and touche2020. These datasets are largely academic and scientific in nature, and the PRF documents may provide useful query terms in these cases. In contrast, the general purpose LLMs may not have sufficient domain knowledge to be useful for these datasets. Second, we note that the question-answering style datasets (fiqa, hotpotqa, msmarco and nq) seem to benefit the most from an LLM approach to query expansion. It is likely that the language model is producing relevant answers towards the query which helps retrieve the relevant passages more effectively. Across all datasets, the **Q2D/PRF** prompt produces the highest average Recall@1K, with the **CoT** prompt as a close second.

Table 1: LLM-based query expansion on the MS-MARCO passage ranking dev set. $^{\blacktriangle}$ indicates a statistically significant (paired t-test, $p < 0.01$) improvement relative to the Q2D Flan-UL2 method. The best result per metric is bolded.

	Recall@1K	MRR@10	NDCG@10
BM25	87.82	18.77	23.44
BM25 + Bo1	88.68	17.75	22.48
BM25 + Bo2	88.32	17.58	22.30
BM25 + KL	88.62	17.71	22.44
Flan-T5-XXL (11B)			
Q2D	88.76	19.07	23.76
Q2D/ZS	88.88	18.55	23.13
Q2D/PRF	89.31	22.13$^{\blacktriangle}$	26.43$^{\blacktriangle}$
Q2E	87.74	18.74	23.37
Q2E/ZS	87.93	18.79	23.45
Q2E/PRF	88.20	19.20	23.83
CoT	89.86	19.16	23.82
CoT/PRF	89.02	22.08$^{\blacktriangle}$	26.32$^{\blacktriangle}$
Flan-UL2 (20B)			
Q2D	89.87	19.22	23.96
Q2D/ZS	86.60	15.56	19.54
Q2D/PRF	89.28	21.42$^{\blacktriangle}$	25.82$^{\blacktriangle}$
Q2E	88.04	18.84	23.52
Q2E/ZS	88.11	18.87	23.56
Q2E/PRF	88.43	19.24	23.90
CoT	**90.61**$^{\blacktriangle}$	20.05$^{\blacktriangle}$	24.85$^{\blacktriangle}$
CoT/PRF	89.30	**22.62**$^{\blacktriangle}$	**26.89**$^{\blacktriangle}$

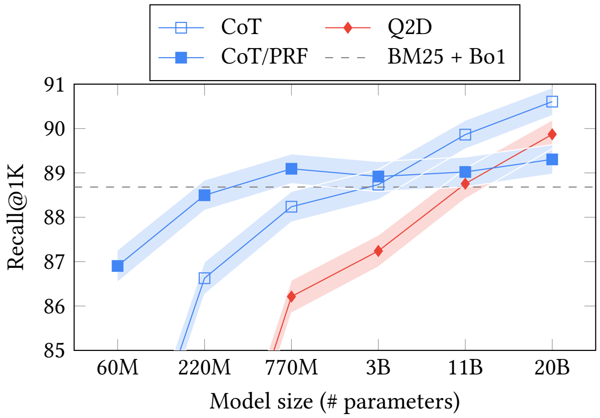

Figure 2: Performance on MS-MARCO passage ranking dev set across different model sizes. The shaded areas indicate a 99% confidence interval.

5.3 The Impact of Model Size

To understand the practical capabilities and limitations of an LLM-based query expander, we compare different model sizes in Figure 2. We range the model size from 60M parameters (Flan-T5-small) up to 11B (Flan-T5-XXL) and also try a 20B parameter model (Flan-UL2) but note that the latter also has a different pre-training objective. In general we observe the expected trend that larger models tend to perform better. The **Q2D** approach requires at least an 11B parameter model to reach parity with the BM25+Bo1 baseline. In contrast, the **CoT** approach only needs a 3B parameter model to reach parity. Furthermore, adding PRF documents to the **CoT** prompt seems to help stabilize the performance for smaller model sizes but does inhibit its performance at larger capacities. A possible explanation for this behavior is that the PRF documents decreases the creativity of

Table 2: Recall@1K of various prompts on BEIR using Flan-UL2. $^{\blacktriangle}$ indicates a statistically significant (paired t-test, $p < 0.01$) improvement relative to the best classical QE method. The best result per dataset is highlighted in bold.

		Classical QE			**LLM-based QE**							
Dataset	BM25	Bo1	Bo2	KL	Q2D	Q2D/ZS	Q2D/PRF	Q2E	Q2E/ZS	Q2E/PRF	CoT	CoT/PRF
arguana	98.93	**99.00**	**99.00**	**99.00**	98.86	98.93	98.93	98.93	98.93	98.93	98.93	98.86
climate-fever	46.60	45.69	45.38	45.65	47.62	47.66	**47.94**	46.08	46.44	46.44	47.42	46.81
cqadupstack	65.55	**66.82**	66.57	66.70	65.51	64.19	65.01	65.69	65.71	65.90	66.39	66.12
dbpedia	63.72	64.77	64.55	64.60	**65.89**	65.47	65.78	63.55	63.92	63.93	65.77	65.06
fever	75.73	76.28	75.83	76.32	**79.06**$^{\blacktriangle}$	78.87$^{\blacktriangle}$	77.29	75.78	75.79	76.27	78.21$^{\blacktriangle}$	77.25
fiqa	77.42	79.18	79.06	78.84	78.34	78.26	78.69	77.33	77.31	77.68	**80.08**	79.03
hotpotqa	85.78	84.84	81.71	84.65	86.90$^{\blacktriangle}$	85.71	87.58$^{\blacktriangle}$	85.60	85.54	87.25$^{\blacktriangle}$	87.54$^{\blacktriangle}$	**88.79**$^{\blacktriangle}$
msmarco	73.61	75.08	75.14	74.66	76.77	75.73	78.75	73.87	73.79	74.14	**79.58**	78.36
nfcorpus	38.70	57.30	57.67	56.46	55.34	**59.81**	59.68	43.38	44.12	47.06	52.63	53.32
nq	78.96	81.09	80.64	80.82	85.18$^{\blacktriangle}$	84.71$^{\blacktriangle}$	83.53$^{\blacktriangle}$	79.30	79.11	80.35	**85.46**$^{\blacktriangle}$	83.11$^{\blacktriangle}$
quora	99.26	99.20	99.12	99.20	99.00	98.84	98.92	99.25	**99.29**	99.26	99.17	99.21
scidocs	57.46	59.78	**61.03**	59.86	59.09	59.78	60.10	57.88	57.70	58.32	58.51	59.69
scifact	97.17	**97.57**	**97.57**	**97.57**	**97.57**	**97.57**	**97.57**	97.17	97.17	97.17	**97.57**	97.17
touche2020	84.96	85.94	**86.38**	86.01	83.61	83.44	84.54	85.21	85.02	86.04	85.51	84.58
trec-covid	42.58	45.21	**45.58**	45.39	43.52	38.05	44.17	43.16	43.12	43.85	43.43	44.02
Average	72.43	74.52	74.35	74.38	74.82	74.47	**75.23**	72.81	72.86	73.50	75.08	74.76

the model, as it may focus too much on the provided documents. Although this helps prevent the model from making errors at smaller model sizes, it also inhibits the creative abilities that we wish to leverage at larger model sizes. The **CoT/PRF** prompt is able to outperform the other prompts at the 770M parameter model size, making it a good candidate for possible deployment in realistic search settings where serving a larger model may be impossible. Overall, it is clear that large models are able to provide significant gains which may limit the practical application of an LLM approach to query expansion. Distillation has been shown to be an effective way to transfer the ability of a large model to a smaller one. We leave the study of distillation of these models for query expansion as future work.

6 LIMITATIONS & FUTURE WORK

There are a number of limitations in our work: First, we only study sparse retrieval (BM25) which is where query expansion is important. Dense retrieval systems (e.g. dual encoders) are less prone to the vocabulary gap and, as a result, are less likely to benefit from a query expansion. Wang et al. [31] has already studied this setting in more detail and we leave the analysis of our prompts for a dense retrieval setting as future work. Second, our work focuses on Flan [32] instruction-finetuned language models. We chose these models due to their ability to follow instructions and the fact that these models are open-source. Our work can naturally be extended to other language models [3, 5, 9, 28] and we leave the study of such models as a topic for future research. Third, we study specific prompt templates (see Appendix A) and there may be other ways to formulate the different prompts. Finally, the computational cost of LLMs may be prohibitive to deploy LLM-based query expansions in practice. It may be possible to distill the output of the large model into a smaller servable model. How to productionize LLM-based query expansions is left as an open problem.

7 CONCLUSION

In this paper we study LLM-based query expansions. In contrast to traditional PRF-based query expansion, LLMs are not restricted to the initial retrieved set of documents and may be able to generate expansion terms not covered by traditional methods. Our proposed method is simple: we prompt a large language model and provide it a query, then we use the model's output to expand the original query with new terms that help during document retrieval.

Our results show that Chain-of-Thought prompts are especially promising for query expansion, since they instruct the model to generate verbose explanations that can cover a wide variety of new keywords. Furthermore, our results indicate that including PRF documents in various prompts can improve top-heavy ranking metric performance during the retrieval stage *and* is more robust when used with smaller model sizes, which can help practical deployment of LLM-based query expansion.

As demonstrated in this paper, IR tasks like query expansion can benefit from LLMs. As the capabilities of LLMs continue to improve, it is promising to see their capabilities translate to various IR tasks. Furthermore, as LLMs become more widely available, they will be easier to use and deploy as core parts of IR systems which is exciting for both practitioners and researchers of such systems.

REFERENCES

[1] Gianni Amati and Cornelis Joost Van Rijsbergen. 2002. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Transactions on Information Systems (TOIS)* 20, 4 (2002), 357–389.

[2] Jagdev Bhogal, Andrew MacFarlane, and Peter Smith. 2007. A review of ontology based query expansion. *Information processing & management* 43, 4 (2007), 866–886.

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.

[4] Claudio Carpineto and Giovanni Romano. 2012. A survey of automatic query expansion in information retrieval. *Acm Computing Surveys (CSUR)* 44, 1 (2012), 1–50.

[5] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311* (2022).

[6] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416* (2022).

[7] 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.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[9] Nan Du, Yanping Huang, Andrew M Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, et al. 2022. Glam: Efficient scaling of language models with mixture-of-experts. In *International Conference on Machine Learning*. PMLR, 5547–5569.

[10] Miles Efron, Peter Organisciak, and Katrina Fenlon. 2012. Improving retrieval of short texts through document expansion. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*. 911–920.

[11] Efthimis N Efthimiadis. 1996. Query Expansion. *Annual review of information science and technology (ARIST)* 31 (1996), 121–87.

[12] D Harman. [n.d.]. Relevance feedback and other query modification techniques. *Information Retrieval: Data Structures & Algorithms* ([n. d.]), 241–263.

[13] 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.

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

[15] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. *choice* 2640 (2016), 660.

[16] Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. 2019. Document expansion by query prediction. *arXiv preprint arXiv:1904.08375* (2019).

[17] Iadh Ounis, Gianni Amati, Vassilis Plachouras, Ben He, Craig Macdonald, and Douglas Johnson. 2005. Terrier information retrieval platform. In *Advances in Information Retrieval: 27th European Conference on IR Research, ECIR 2005, Santiago de Compostela, Spain, March 21-23, 2005. Proceedings 27*. Springer, 517–519.

[18] Yonggang Qiu and Hans-Peter Frei. 1993. Concept based query expansion. In *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*. 160–169.

[19] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research* 21, 1 (2020), 5485–5551.

[20] Stephen E Robertson. 1990. On term selection for query expansion. *Journal of documentation* 46, 4 (1990), 359–364.

[21] 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.

[22] Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. 1995. Okapi at TREC-3. *Nist Special Publication Sp* 109 (1995), 109.

[23] Joseph John Rocchio Jr. 1971. Relevance feedback in information retrieval. *The SMART retrieval system: experiments in automatic document processing* (1971).

[24] Dwaipayan Roy, Debjyoti Paul, Mandar Mitra, and Utpal Garain. 2016. Using word embeddings for automatic query expansion. *arXiv preprint arXiv:1606.07608* (2016).

[25] Tao Tao, Xuanhui Wang, Qiaozhu Mei, and ChengXiang Zhai. 2006. Language model information retrieval with document expansion. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*. 407–414.

[26] Yi Tay, Mostafa Dehghani, Vinh Q Tran, Xavier Garcia, Dara Bahri, Tal Schuster, Huaixiu Steven Zheng, Neil Houlsby, and Donald Metzler. 2022. Unifying language learning paradigms. *arXiv preprint arXiv:2205.05131* (2022).

[27] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*. https://openreview.net/forum?id=wCu6T5xFjeJ

[28] Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. 2022. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239* (2022).

[29] Ellen M Voorhees. 1994. Query expansion using lexical-semantic relations. In *SIGIR'94: Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, organised by Dublin City University*. Springer, 61–69.

[30] Ellen M Voorhees et al. 1999. The trec-8 question answering track report.. In *Trec*, Vol. 99. 77–82.

[31] Liang Wang, Nan Yang, and Furu Wei. 2023. Query2doc: Query Expansion with Large Language Models. *arXiv preprint arXiv:2303.07678* (2023).

[32] Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652* (2021).

[33] Zhi Zheng, Kai Hui, Ben He, Xianpei Han, Le Sun, and Andrew Yates. 2020. BERT-QE: contextualized query expansion for document re-ranking. *arXiv preprint arXiv:2009.07258* (2020).

[34] 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.

A PROMPTS

Table 3 contains all the prompts tried in this paper. In each prompt {query} denotes the query for which we want to generate a query expansion. We denote with {query 1}, . . ., {query 4} the sample queries from the MS-MARCO train set. Similarly, {doc 1}, . . ., {doc 4} represent relevant passages corresponding to the sampled queries, and, {expansion 1}, . . ., {expansion 4} represent corresponding expansions generated with Terrier KL method (at most 20 terms) from those relevant passages. Finally we denote with {PRF doc 1}, . . ., {PRF doc 3} the top 3 retrieved documents using the original query, acting as Pseudo-Relevance Feedback documents. For the CoT prompt, we note that the model tends to output “The final answer:” or “So the final answer is:” towards the end and we filter those two sentences out prior to concatenating the model output with the query.

Table 3: The various query expansion prompts.

ID	Prompt

Q2D [31]	Write a passage that answers the given query:
	
	Query: {query 1}
	Passage: {doc 1}
	
	Query: {query 2}
	Passage: {doc 2}
	
	Query: {query 3}
	Passage: {doc 3}
	
	Query: {query 4}
	Passage: {doc 4}
	
	Query: {query}
	Passage:

Q2D/ZS	Write a passage that answers the following query: {query}

Q2D/PRF	Write a passage that answers the given query based on the context:
	
	Context: {PRF doc 1}
	{PRF doc 2}
	{PRF doc 3}
	Query: {query}
	Passage:

Q2E	Write a list of keywords for the given query:
	
	Query: {query 1}
	Keywords: {expansion 1}
	
	Query: {query 2}
	Keywords: {expansion 2}
	
	Query: {query 3}
	Keywords: {expansion 3}
	
	Query: {query 4}
	Keywords: {expansion 4}
	
	Query: {query}
	Keywords:

Q2E/ZS	Write a list of keywords for the following query: {query}

Q2E/PRF	Write a list of keywords for the given query based on the context:
	
	Context: {PRF doc 1}
	{PRF doc 2}
	{PRF doc 3}
	Query: {query}
	Keywords:

CoT	Answer the following query: {query} Give the rationale before answering

CoT/PRF	Answer the following query based on the context:
	
	Context: {PRF doc 1}
	{PRF doc 2}
	{PRF doc 3}
	Query: {query}
	
	Give the rationale before answering

B EXAMPLE OUTPUT

Table 4 shows the results of a query expansion for both the Flan-T5-Large (770M) model size and the Flan-UL2 (20B) model size. First, note that at the smaller model size, the CoT and Q2D prompts are not producing the correct answer which is harmful for retrieval performance. The CoT/PRF prompt, being more grounded in its PRF documents, avoids this problem and correctly produces the answer "Tata Motors" which helps retrieve the relevant passage. At the larger model size (Flan-UL2), all prompts Q2D, CoT and CoT/PRF produce the correct answer "Tata Motors". However, the CoT prompt provides the most verbose explanation towards its answer and has many term overlaps with the relevant passage, improving its overall retrieval performance.

Table 4: Example of query expansions with Flan-UL2. Green highlights the correct answer while blue indicate specific term overlaps that indicate potentially useful query expansions.

	Query	who owns jaguar motors? (ID=1045405)
	Passage	Jaguar is the luxury vehicle brand of Jaguar Land Rover, a British multinational car manufacturer with its headquarters in Whitley, Coventry England, owned by the Indian company Tata Motors since 2008.
Flan-T5-Large (770M)	Q2D	Jaguar Motors is a British luxury car manufacturer based in the United Kingdom.
	CoT	Jaguar Land Rover is the owner of Jaguar. The answer: Jaguar Land Rover.
	CoT/PRF	The relevant information is: Jaguar is owned by the Indian automobile manufacturer Tata Motors Ltd. So the final answer is the Indian automobile manufacturer Tata Motors Ltd.
Flan-UL2 (20B)	Q2D	Jaguar is a division of Tata Motors, a company owned by the Tata family.
	CoT	Jaguar Land Rover is a British multinational car manufacturer, founded by William Lyons in 1931. Its headquarters are in Whitley, Coventry, United Kingdom and is a constituent of the FTSE 250 Index. The company is a wholly owned subsidiary of Tata Motors of India. So the final answer is Tata Motors.
	CoT/PRF	Jaguar is owned by the Indian automobile manufacturer Tata Motors Ltd. The final answer: Tata Motors Ltd.