MILL: Mutual Verification with Large Language Models for Zero-Shot Query Expansion

Anonymous ACL submission

Abstract

Query expansion, pivotal in search engines, enhances representation of user information needs with additional terms. While existing methods expand queries using retrieved or generated contextual documents, each approach has notable limitations. Retrieval-based methods often fail to accurately capture search intent, particularly with brief or ambiguous queries. Generation-based methods, utilizing large language models (LLMs), generally lack corpus-specific knowledge and entail high fine-tuning costs. To address these gaps, we propose a novel zero-shot query expansion framework utilizing LLMs for mutual verification. Specifically, we first design a query-query-document generation method, leveraging LLMs' zero-shot reasoning ability to produce diverse sub-queries and corresponding documents. Then, a mutual verification process synergizes generated and retrieved documents for optimal expansion. Our proposed method is fully zero-shot, and extensive experiments on three public benchmark datasets are conducted to demonstrate its effectiveness over existing methods. Our code is available online at https://anonymous.4open.science/r/MILL-AE47 to ease reproduction.

1 Introduction

Query expansion is a critical technique in search systems, aiming to effectively capture and represent users' information needs (Efthimiadis, 1996). Search engines, for instance, employ query expansion to resolve ambiguities in queries and align the vocabulary of queries and documents. Central to this task is the development of contextual documents, comprising additional query terms, to enhance effectiveness (Azad and Deepak, 2019).

Specifically, existing research predominantly falls into two categories: retrieval-based and generation-based methods. Retrieval-based methods (Lv and Zhai, 2010; Yan et al., 2003; Li et al., 2022) typically construct contextual documents from the targeted corpus, assuming that the top-retrieved documents (i.e., pseudo-relevance feedback (PRF)) are reasonable expansions of a given query. Generation-based methods (Jagerman et al., 2023; Mao et al., 2023; Wang et al., 2023) often utilize advanced generative models, such as Large Language Models, as an external knowledge base for producing contextual documents.

However, both methods have clear limitations. For retrieval-based methods, it has been observed in practice that the documents retrieved with the original query do not align well with the information needs, particularly when the original query itself is brief and ambiguous (Cao et al., 2008; Jagerman et al., 2023). For generation-based methods, directly using off-the-shelf LLMs in a few-shot or zero-shot manner can hardly align the model with a specific corpus (Wang et al., 2023). In contrast, the LLMs could easily generate useless out-of-domain information.

To this end, we propose a novel query expansion framework based on Large Language Models (LLMs), integrating both retrieved and generated documents to mitigate their respective limitations. First, to improve contextual document generation, we design a query-query-document prompt that leverages an LLM as a zero-shot reasoner to decompose a query into multiple sub-queries during contextual document generation. This helps the LLM generate diverse contextual information that is more likely to cover the underlying search intent.

Next, we propose a mutual verification framework that exploits generated and retrieved contextual documents for query expansion. To be more specific, we propose to filter out the uninformative generated documents via comparing their relevance with the top-retrieved documents. By doing this, the selected generated documents are intuitively more aligned with the target corpus. Conversely, we also filter out the noisy retrieved documents via comparing their relevance with the generated documents. The external contextual knowledge embedded in the generated documents can facilitate the retrieved documents to more accurately reveal search intent. We evaluate the proposed method on the downstream information retrieval task in a zero-shot manner. The results on three public datasets demonstrate that our proposed method significantly outperforms the state-of-the-art baselines. Overall, the contributions can be summarized as follows:

- We propose a **M**utual Ver**I**fication method with **L**arge **L**anguage model (denoted as MILL), a novel framework that combines generated and retrieved context for query expansion. MILL is able to mitigate the limitations of generated and retrieved context, and thus can provide more high-quality context for query expansion.
- To improve the generated contextual documents, we design a query-query-document prompting method, which elicits richer and more diverse knowledge from LLMs to cover the underlying search intents and information needs of users.
- MILL can perform high-quality query expansion in a zero-shot manner. We conduct extensive experiments on the downstream information retrieval task on three public datasets. The results demonstrate that MILL can significantly outperform existing retrieval and generation-based methods.

2 Problem Definition

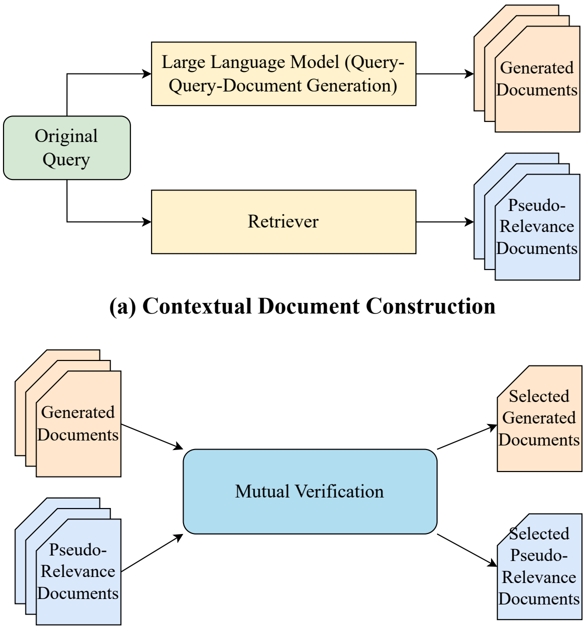
Given a user query q, query expansion is to apply a function f to expand q with additional contextual information: $q' = f_\theta(q)$, where θ represents the parameters. Using the expanded query q' should be able to achieve better downstream retrieval performance compared to the original query q. More formally, such an objective can be defined as

$$\underset{\theta}{\operatorname{argmax}}\, \mathcal{M}(q', R), \text{ where } q' = f_\theta(q). \quad (1)$$

where $\mathcal{M}$ denotes the evaluation metric of the retrieval performance (e.g., recall, NDCG), and R denotes the retrieval model.

3 Methodology

In this section, we introduce our proposed query expansion method in detail. Specifically, we give an overview of MILL in Section 3.1, elaborate the query-query-document generation in Section 3.2, and introduce the mutual verification framework in Section 3.3.


Original Query
Large Language Model (Query-Query-Document Generation)
Retriever
Generated Documents
Pseudo-Relevance Documents
Mutual Verification
Selected Generated Documents
Selected Pseudo-Relevance Documents

(a) Contextual Document Construction

(b) Mutual Verification

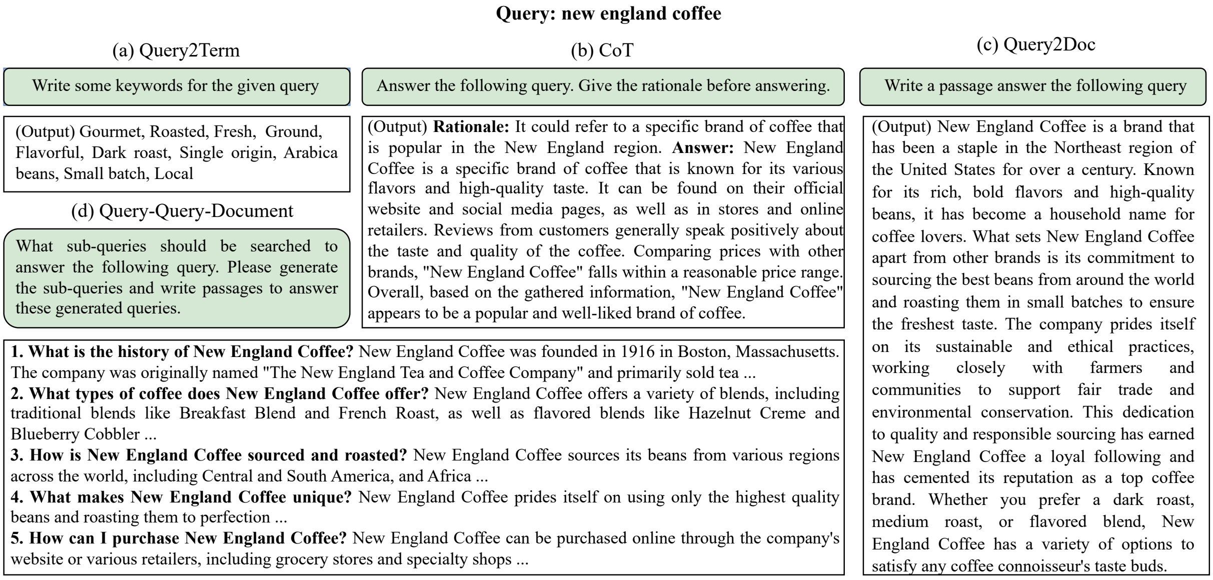
Figure 1: Overview of MILL.

3.1 Overview

The overall workflow of MILL is depicted in Figure 1, which comprises two steps, i.e., contextual document construction (Figure 1(a)) and mutual verification (Figure 1(b)). In particular, the two steps focus on the diversity and quality of contextual documents, respectively.

In the **contextual document construction** stage, we aim to construct diverse sets of contextual documents via both retrieval and generation. To create diverse contextual documents via generation, we propose a query-query-document prompt, which instructs an LLM to generate sub-queries and contextual documents in a step-by-step manner. This can better leverage the reasoning ability of LLMs that decompose a given query expansion task into multiple sub-tasks, where the generated documents could be more diverse and informative.

In the **mutual verification** stage, we aim to identify those high-quality contextual documents constructed in the first stage. In particular, the mutual verification leverages the strengths of generated documents in implying the search intent and the domain-specific nature of PRF documents, enabling a reciprocal selection between the two types of contextual documents. As a result, the finalized documents are more high-quality query expansion to be applied in downstream retrieval tasks.



Query: new england coffee

(a) Query2Term

Write some keywords for the given query

(Output) Gourmet, Roasted, Fresh, Ground, Flavorful, Dark roast, Single origin, Arabica beans, Small batch, Local

(b) CoT

Answer the following query. Give the rationale before answering.

(Output) **Rationale:** It could refer to a specific brand of coffee that is popular in the New England region. **Answer:** New England Coffee is a specific brand of coffee that is known for its various flavors and high-quality taste. It can be found on their official website and social media pages, as well as in stores and online retailers. Reviews from customers generally speak positively about the taste and quality of the coffee. Comparing prices with other brands, "New England Coffee" falls within a reasonable price range. Overall, based on the gathered information, "New England Coffee" appears to be a popular and well-liked brand of coffee.

(c) Query2Doc

Write a passage answer the following query

(Output) New England Coffee is a brand that has been a staple in the Northeast region of the United States for over a century. Known for its rich, bold flavors and high-quality beans, it has become a household name for coffee lovers. What sets New England Coffee apart from other brands is its commitment to sourcing the best beans from around the world and roasting them in small batches to ensure the freshest taste. The company prides itself on its sustainable and ethical practices, working closely with farmers and communities to support fair trade and environmental conservation. This dedication to quality and responsible sourcing has earned New England Coffee a loyal following and has cemented its reputation as a top coffee brand. Whether you prefer a dark roast, medium roast, or flavored blend, New England Coffee has a variety of options to satisfy any coffee connoisseur's taste buds.

(d) Query-Query-Document

What sub-queries should be searched to answer the following query. Please generate the sub-queries and write passages to answer these generated queries.

1. What is the history of New England Coffee? New England Coffee was founded in 1916 in Boston, Massachusetts. The company was originally named "The New England Tea and Coffee Company" and primarily sold tea ...
2. What types of coffee does New England Coffee offer? New England Coffee offers a variety of blends, including traditional blends like Breakfast Blend and French Roast, as well as flavored blends like Hazelnut Creme and Blueberry Cobbler ...
3. How is New England Coffee sourced and roasted? New England Coffee sources its beans from various regions across the world, including Central and South America, and Africa ...
4. What makes New England Coffee unique? New England Coffee prides itself on using only the highest quality beans and roasting them to perfection ...
5. How can I purchase New England Coffee? New England Coffee can be purchased online through the company's website or various retailers, including grocery stores and specialty shops ...

Figure 2: Query-query-document prompt compared to Query2Term, CoT, and Query2Doc. Query-query-document instructs the LLM to expand the original query from multiple perspectives by inferring the sub-queries and generating corresponding contextual documents.

3.2 Query-Query-Document Generation

Recently, a handful of studies (Wang et al., 2023; Jagerman et al., 2023) have explored using Large Language Models (LLMs) to expand queries and gain initial success. However, most of them use a rather simple prompt for document generation, e.g., “write a passage that answers the given query”. For a brief or ambiguous query that has multiple possible intents, the generation results could easily miss the real search intent. Motivated by this, we design a novel zero-shot prompt, particularly for the query expansion task. This method can exploit the reasoning ability of LLMs to first decompose the original query into multiple sub-queries before document generation. This improves generation diversity, and the contextual documents are more likely to cover the real search intent.

As shown in Figure 2(d), we use the instruction "what sub-queries should be searched to answer the following query: {query}." to generate sub-queries that further clarify the original query. At the same time, we instruct the language model to generate contextual documents for each sub-query through "I will generate the sub-queries and write passages to answer these generated queries." By doing this, we finally have multiple sub-queries and their corresponding contextual documents, which are more likely to cover the user’s search intent. Note that the proposed method is zero-shot, which can be easily extended to few-shot.

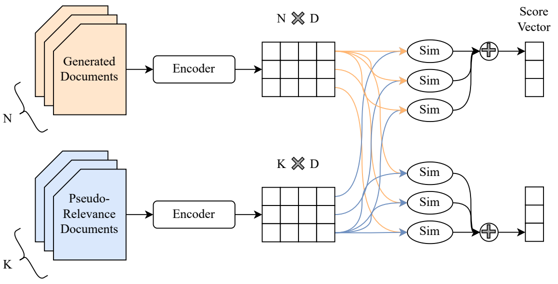


Figure 3: Overall of mutual verification.

3.3 Mutual Verification

Next, we elaborate on the mutual verification framework, where we leverage the aforementioned generated documents and pseudo-relevance documents (i.e., the retrieval-based contextual documents) to improve the overall quality of query expansion. The intuition is to leverage two types of information to complement each other, which are 1) the corpus-specific domain information of retrieved pseudo-relevance documents, and 2) the generated information of LLM reasoning that is more likely to uncover real search intent.

More specifically, the inputs of mutual verification have two sets of contextual documents:

$$\mathcal{D}^{\text{LLM}} = \{d_n^{\text{LLM}}\} = \text{LLM}(p, q),\ n \in (0, N] \quad (2)$$

$$\mathcal{D}^{\text{PRF}} = \{d_k^{\text{PRF}}\} = R_r(q),\ k \in (0, K] \quad (3)$$

where $\mathcal{D}^{\text{LLM}}$ represents the N LLM-generated documents with query-query-document prompt (denoted as p), and $\mathcal{D}^{\text{PRF}}$ represents the K documents retrieved by a vanilla PRF method (denoted as R_r), e.g., BM25 retrieval. Note that each generated document comprises a series of sub-queries and their corresponding passages.

Next, we aim to rerank the documents in $\mathcal{D}^{\text{LLM}}$ and $\mathcal{D}^{\text{PRF}}$. In specific, we first use a off-the-shelf dense representation model to compute the representation (i.e., $\mathbf{x}_n^{\text{LLM}}$ or $\mathbf{x}_k^{\text{PRF}}$) of each document (i.e., d_n^{LLM} or d_k^{PRF}) as

$$\mathbf{x}_n^{\text{LLM}} = \text{Encoder}(d_n^{\text{LLM}}), \tag{4}$$

$$\mathbf{x}_k^{\text{PRF}} = \text{Encoder}(d_k^{\text{PRF}}), \tag{5}$$

where $\mathbf{x}_n^{\text{LLM}}$ denotes the vector for n-th generated document and $\mathbf{x}_k^{\text{PRF}}$ denotes the vector for k-th pseudo-relevance documents.

Then, we compute the semantic relevance between every pair of d_n and d_k with cosine similarity (denoted as sim$(\cdot)$), and assign a score to every document as

$$s_n^{\text{LLM}} = \sum\nolimits_{k=1}^{K} \text{sim}(\mathbf{x}_n^{\text{LLM}}, \mathbf{x}_k^{\text{PRF}}), \tag{6}$$

$$s_k^{\text{PRF}} = \sum\nolimits_{n=1}^{N} \text{sim}(\mathbf{x}_k^{\text{PRF}}, \mathbf{x}_n^{\text{LLM}}). \tag{7}$$

Here, we score every generated document d_n^{LLM} via aggregating its semantic relevance scores with all pseudo-relevance documents. Therefore, the score s_n^{LLM} can be interpreted as how well d_n^{LLM} is aligned with the target corpus. On the other hand, the score s_k^{PRF} can be viewed as how well the retrieved document d_k^{PRF} is likely to be a reasonable context judged by the reasoning results of LLM.

Finally, we select the top-scored documents in both sets as the final contextual documents as

$$\mathcal{D}_s^{\text{LLM}} = \{d_n^{\text{LLM}}\},\ n \in \{n\,|s_n^{\text{LLM}} \in TopN'(s^{\text{LLM}})\},$$
$$\mathcal{D}_s^{\text{PRF}} = \{d_k^{\text{PRF}}\},\ k \in \{k\,|s_k^{\text{PRF}} \in TopK'(s^{\text{PRF}})\}, \tag{8}$$

where $\mathcal{D}_s^{\text{LLM}}$ and $\mathcal{D}_s^{\text{PRF}}$ are the final selected document sets.

3.4 Query Expansion for Retrieval

After mutual verification, we integrate the selected generated documents and pseudo-relevance documents with the original query to perform the final retrieval task. In particular, we concatenate them as the new query q' as:

$$q' = \text{concat}(q,\ \mathcal{D}_s^{\text{PRF}},\ \mathcal{D}_s^{\text{LLM}}) \tag{9}$$

It is worth noting that the proposed query expansion method does not need any additional labeled data and model fine-tuning. Such a zero-shot method with off-the-shelf LLM and retriever has huge potential to be applied in various search systems.

4 Experiments

4.1 Datasets and Metrics

To evaluate the effectiveness of our proposed method, we conduct extensive experiments on the following public datasets: TREC-DL-2020, MSMARCO and BEIR.

- **TREC-DL-2020 (Craswell et al., 2021).** TREC-DL-2020[1] is the dataset used in the second year of the popular TREC Deep Learning Track. We choose the passage retrieval task, which contains 200 queries and 8.84 million passages.

- **MSMARCO (Nguyen et al., 2016).** MSMARCO[2] is a collection of datasets constructed to advance the development of deep learning in the search field. We choose the passage dataset as our experimental scenario and take the first 100 queries from the dev group as the test queries.

- **BEIR (Thakur et al., 2021).** BEIR[3] is a heterogeneous benchmark for comprehensive zero-shot evaluation of methods in various information retrieval tasks. We select 7 datasets with small test or dev sets from the 18 available datasets.

Following previous work (Claveau, 2021; Jagerman et al., 2023; Mao et al., 2023), we use the NDCG@N, MAP@N, Recall@N, and MRR@N as the evaluation metrics, each of which is reported with N $\in \{10, 100, 1000\}$.

4.2 Baselines

We conduct comparative experiments with the following baselines, which can be divided into two categories: (1) **Traditional query expansion methods**: Bo1 (Amati and Van Rijsbergen, 2002), KL (Amati and Van Rijsbergen, 2002), RM3 (Abdul-Jaleel et al., 2004), and AxiomaticQE (Fang and Zhai, 2006; Yang and Lin, 2019). (2) **LLM-based expansion methods**: Query2Term, Query2Term-FS (the few-shot version of Query2Term), Query2Term-PRF (PRF document augmented Query2Term), Query2Doc (Wang et al., 2023), Query2Doc-FS,

[1] https://microsoft.github.io/msmarco/TREC-Deep-Learning-2020

[2] https://microsoft.github.io/msmarco/

[3] https://github.com/beir-cellar/beir

Table 1: Overall comparison on TREC-DL-2020 and MSMARCO. The optimal results are highlighted in bold, while the suboptimal results are underscored. The results are reported on NDCG@N, AP@N, Recall@N, and MRR@N with N $\in \{10, 100, 1000\}$. The improvements are all significant (i.e., two-sided t-test with $p < 0.05$) between the optimal and suboptimal results.

	Metrics	NDCG			AP			Recall			MRR		
		@10	@100	@1000	@10	@100	@1000	@10	@100	@1000	@10	@100	@1000
TREC-DL-2020	No expansion	49.36	50.26	59.81	14.27	31.42	35.87	17.61	50.47	75.12	80.21	80.21	80.21
	Traditional expansion methods												
	Bo1	49.47	53.25	63.11	14.79	34.43	39.67	17.74	54.66	79.48	80.83	80.99	80.99
	KL	49.27	53.20	63.01	14.68	34.31	39.53	17.66	54.70	79.39	80.83	80.99	80.99
	RM3	50.43	54.02	63.47	14.93	35.13	40.22	17.89	55.80	79.94	78.49	78.59	78.59
	AxiomaticQE	49.36	50.26	59.81	14.27	31.42	35.87	17.61	50.47	75.12	80.21	80.21	80.21
	LLM-based expansion methods												
	Query2Term	50.12	52.43	62.27	13.12	33.06	38.49	17.39	54.61	79.07	78.74	78.77	78.78
	Query2Term-FS	47.80	49.16	60.50	13.33	30.16	35.59	15.82	50.22	78.76	79.38	79.83	79.83
	Query2Term-PRF	47.76	48.92	59.57	12.32	29.03	33.70	14.70	49.29	76.68	78.97	79.29	79.29
	Query2Doc	61.22	60.13	69.97	**19.06**	41.31	47.03	21.57	57.58	83.38	88.27	88.44	88.44
	Query2Doc-FS	61.45	59.30	69.40	18.94	39.75	45.27	**21.65**	56.30	82.57	90.32	90.37	90.38
	Query2Doc-PRF	55.28	57.60	67.09	17.00	38.21	43.49	19.74	58.50	82.57	84.22	84.49	84.49
	CoT	58.39	56.74	67.02	18.15	37.32	42.34	21.51	54.02	80.11	88.02	88.02	88.03
	CoT-PRF	60.81	58.41	67.47	19.02	39.27	44.04	21.71	56.84	80.49	89.00	89.00	89.00
	MILL	**61.79**	**61.15**	**71.23**	19.05	**41.76**	**48.17**	21.61	**59.40**	**85.27**	**92.61**	**92.71**	**92.72**
MSMARCO	No expansion	28.69	34.02	36.23	23.56	24.65	24.72	44.50	69.00	86.50	22.65	23.76	23.83
	Traditional expansion methods												
	Bo1	29.18	33.44	35.89	23.61	24.33	24.43	46.50	67.50	86.50	24.07	24.82	24.91
	KL	29.20	33.59	36.17	23.93	24.73	24.83	45.50	66.50	86.50	**24.39**	**25.22**	**25.31**
	RM3	26.93	32.23	34.34	21.81	22.87	22.94	42.50	67.00	83.50	22.25	23.33	23.41
	AxiomaticQE	28.69	34.02	36.23	23.56	24.65	24.72	44.50	69.00	86.50	22.65	23.76	23.83
	LLM-based expansion methods												
	Query2Term	23.28	29.50	32.00	19.74	21.01	21.08	34.17	63.17	83.67	19.91	21.17	21.24
	Query2Term-FS	24.26	29.76	32.07	20.41	21.43	21.50	36.33	62.50	81.33	20.78	21.87	21.94
	Query2Term-PRF	21.56	27.02	29.26	16.04	17.05	17.12	38.67	64.83	83.33	16.04	17.11	17.17
	Query2Doc	25.83	31.31	33.82	20.27	21.33	21.42	43.50	69.00	88.83	20.39	21.50	21.58
	Query2Doc-FS	28.23	33.22	35.89	23.10	23.99	24.09	44.67	68.83	89.50	23.00	23.94	24.04
	Query2Doc-PRF	25.45	29.99	32.36	20.31	21.25	21.33	41.44	62.50	81.17	20.45	21.35	21.43
	CoT	26.13	31.84	34.25	21.38	22.44	22.54	41.00	68.33	86.83	21.47	22.55	22.64
	CoT-PRF	28.93	34.17	36.32	23.51	24.52	24.60	46.12	70.87	87.50	23.64	24.69	24.77
	MILL	**29.99**	**34.92**	**37.26**	**24.01**	**24.98**	**25.07**	**48.67**	**71.67**	**89.83**	24.02	25.02	25.10

Query2Doc-PRF, CoT (Jagerman et al., 2023), CoT-PRF. The details of baselines and the prompts used in this paper are introduced in Appendix A.1 and Appendix A.2. Besides, to conduct a fair comparison for the LLM-based baselines, we generate 3 expanded queries for each baseline and concatenate them as the final expansion result.

4.3 Implementation Details

We implement MILL and the baselines with PyTerrier (Macdonald and Tonellotto, 2020), a Python library helps conduct information retrieval experiments. For the BM25 retriever, we use the default parameters ($b = 0.75, k_1 = 1.2, k_3 = 8.0$) provided by PyTerrier. For MILL and all the LLM-based baselines, we use the text-davinci-003 API (Brown et al., 2020) provided by OpenAI to generate contextual documents. The generation parameters are set as temperature $= 0.7$ and top_p $= 1$. We use the text-embedding-ada-002 provided by OpenAI as the text encoder, where the length of the returned vector is 1536. For other hyperparameters, we set the selection number of generated documents and PRF documents as 3, and the number of candidates as 5. Besides, considering the verbose nature of the contextual documents, we follow the approach suggested in paper (Wang et al., 2023) that the expanded query involves 5 samplings of the original query to emphasize its significance.

4.4 Main Results

Tables 1 and 2 show the experimental results. The full results for the 7 selected datasets in BEIR are listed in Appendix A.3. We can draw the following key findings:

- Traditional query expansion methods exhibit positive effects for retrieval, while these carefully designed methods are outperformed by Query2Doc and CoT variants by a large margin. This implies that LLM-based methods are more promising for the query expansion task.

- Among LLM-based methods, CoT and Query2Doc variants are more effective than Query2Term variants. The reason could

Table 2: Overall comparison on 7 datasets in BEIR. The optimal results are highlighted in bold, while the suboptimal results are underscored. Full results including other evaluation metrics are listed in Appendix A.3. The improvements are all significant (i.e., two-sided t-test with $p < 0.05$) between the optimal and suboptimal results.

	Datasets	TREC-COVID	TOUCHE	SCIFACT	NFCORPUS	DBPEDIA	FIQA-2018	SCIDOCS
NDCG@1000	No expansion	42.04	55.32	70.27	30.02	38.70	35.28	25.14
	Traditional expansion methods							
	Bo1	44.73	56.62	68.34	37.01	39.05	34.97	26.14
	KL	44.88	56.72	67.83	37.18	38.87	35.12	26.15
	RM3	44.54	55.79	65.28	37.27	38.11	33.14	25.91
	AxiomaticQE	42.06	55.32	70.28	30.02	38.70	35.28	25.14
	LLM-based expansion methods							
	Query2Term	45.09	52.95	69.57	33.82	33.51	32.12	25.11
	Query2Term-FS	44.86	57.1	71.39	38.57	39.36	35.78	26.18
	Query2Term-PRF	42.94	53.72	60.79	38.21	34.83	31.50	24.97
	Query2Doc	45.41	60.32	71.19	38.76	44.79	37.63	27.40
	Query2Doc-FS	44.39	59.99	71.89	38.09	45.11	37.96	27.18
	Query2Doc-PRF	47.97	56.84	67.82	39.41	39.85	34.09	26.16
	CoT	46.93	60.77	71.63	38.88	43.05	37.28	27.50
	CoT-PRF	46.55	59.03	73.65	39.84	40.43	38.04	26.23
	MILL	**51.17**	**61.29**	**74.14**	**41.75**	**46.39**	**39.23**	**28.36**
Recall@1000	No expansion	40.52	85.05	97.00	36.06	63.61	77.42	55.04
	Traditional expansion methods							
	Bo1	43.64	86.00	97.67	54.38	64.90	79.18	57.47
	KL	43.63	86.14	97.67	54.79	64.71	78.84	57.38
	RM3	43.71	85.79	97.67	56.12	64.37	78.82	57.88
	AxiomaticQE	40.53	85.05	97.00	36.06	63.61	77.42	55.04
	LLM-based expansion methods							
	Query2Term	43.67	77.24	99.00	58.82	58.90	78.22	60.00
	Query2Term-FS	43.89	**85.33**	98.33	61.72	65.67	81.84	60.15
	Query2Term-PRF	41.99	83.29	97.50	60.55	61.11	76.31	59.25
	Query2Doc	43.71	84.08	99.00	61.09	70.29	82.72	61.63
	Query2Doc-FS	42.80	83.95	99.33	59.55	70.04	83.46	61.33
	Query2Doc-PRF	46.20	83.5	99.00	62.50	66.41	79.14	59.50
	CoT	45.01	84.42	98.67	60.63	69.24	83.56	60.90
	CoT-PRF	44.93	84.37	98.67	59.87	66.06	82.14	58.72
	MILL	**49.33**	84.99	**99.67**	**64.95**	**71.13**	**84.23**	**61.86**

be that generated documents contain more contextualized information than discrete keywords.

- Using pseudo-relevance documents and few-shot examples as instructions in LLM-based methods does not necessarily yield positive gains. For instance, Query2Doc-PRF is worse than Query2Doc in TREC-DL-2020 and MSMARCO. This shows that the query expansion task is non-trivial to be aligned to a specific corpus with straightforward prompting techniques.

- MILL is more effective than all the baselines in general. Despite the MRR@10 on MSMARCO, MILL achieves either the best or the second best performance on all metrics and datasets in Tables 1 and 2. It is also worth noting that MILL is a zero-shot method that is more applicable in various real-world applications.

4.5 Ablation Study

We design the following variants of MILL to conduct the ablation study:

- **w/o QQD**: In contextual document generation, we replace the query-query-document prompt with a vanilla query-to-document prompt, i.e, "Write a passage answer the following query: {query}".

- **w/o Pseudo-relevance Document Selection (PDS)**: We directly use K' top-retrieved documents of the original query as $\mathcal{D}_s^{\text{PRF}}$, without reranking and selection using generated documents $\mathcal{D}^{\text{LLM}}$.

- **w/o Generated Document Selection (GDS)**: We directly use N' generation documents as $\mathcal{D}_s^{\text{LLM}}$, without reranking and selection using pseudo-relevance documents $\mathcal{D}^{\text{PRF}}$.

Table 3 shows the results of the ablation study on TREC-DL-2020, where we can draw the following conclusions: (1) **MILL** is better than **w/o QQD**, which demonstrates the effectiveness of our proposed query-query-document prompt. This shows that query-query-document prompt can effectively leverage the reasoning capabilities of LLMs, assisting LLMs to reveal more diverse and specific search intent. (2) **MILL** is superior to both **w/o**

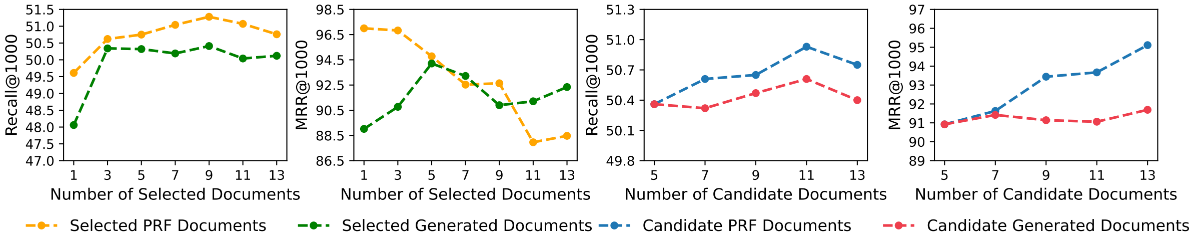


Figure 4: Varying the number of candidate and selected documents.

Table 3: Ablation results on TREC-DL-2020.

Metrics		w/o QQD	w/o PDS	w/o GDS	MILL
NDCG	@100	59.89	60.29	59.60	**61.15**
	@1000	69.46	70.27	69.73	**71.23**
AP	@100	41.56	41.56	41.33	**41.76**
	@1000	47.39	47.75	47.37	**48.17**
Recall	@100	58.55	58.76	59.29	**59.40**
	@1000	83.98	85.15	84.51	**85.27**
MRR	@100	87.69	89.23	88.97	**92.71**
	@1000	87.69	89.23	88.98	**92.72**

PDS and **w/o GDS**, which verifies the effectiveness of the mutual verification. By mutually selecting the generated and pseudo-relevance documents, it effectively mitigates the corpus unalignment problem of LLMs and compensates for the inaccurate search intent of conventional pseudo-relevance documents. (3) We can also find that **w/o PDS** performs better than **w/o GDS**. This indicates that the selection of high-quality generated documents has more performance gain for query expansion.

4.6 Varying the Number of Documents

In the aforementioned experiments, the default number of candidate (i.e., both generated and retrieved) documents is set to $K = N = 5$, and the number of final selected documents is set to $K' = N' = 3$. In this subsection, we vary the number of candidates and selected documents and report the performance of MILL on TREC-COVID, w.r.t. Recall@1000 and MRR@1000. More details and results can be found in Appendix A.4.

From Figure 4, we have observations: (1) More selected pseudo-relevance documents decreases MRR@1000 dramatically. This shows that more selected pseudo-relevance documents usually bring more noise to query expansion. In contrast, the generated documents are rather robust, where more selections does not significantly undermines the performance. (2) When we introduce more candidate documents, the mutual verification framework is able to effectively select pseudo-relevance documents, where both Recall@100 and MRR@1000 increase. This shows that LLM-generated documents are very useful for filtering out noisy pseudo-relevance documents. On the other hand, more generated candidate documents does not bring further performance gain, when the number of selected documents is fixed.

4.7 Case Study

We show an illustrative example in Table 4, which contains the original query, the pseudo-relevance document, the LLM-generated document, and the ground truth (i.e., the human labeled relevant document). Those words of ground truth passage that appear in the pseudo-relevance document is highlighted in bold, and those in the generated documents of different sub-queries are marked with different colors. We can see that the generated document is able to provide more useful information for identifying the ground truth passage.

5 Related Work

5.1 Query Expansion

Query expansion is a prevalent technique in search platforms, which restructures the original query to more accurately express search intent and enhance the alignment with corpus (Bhogal et al., 2007; Carpineto and Romano, 2012; Efthimiadis, 1996). Early studies employed lexical knowledge bases (Qiu and Frei, 1993; Voorhees, 1994) or Pseudo-relevance Feedback (PRF) (Amati and Van Rijsbergen, 2002; Robertson, 1990; Rocchio Jr, 1971; Lv and Zhai, 2010; Yan et al., 2003; Li et al., 2022) for expanding the query with additional information. PRF documents can conveniently supplement information for any query, but they also encounter the issue of misalignment with the original query (Jagerman et al., 2023).

Recently, Large Language Models (LLMs) have also been applied to the task of query expansion. In particular, Query2Doc (Wang et al., 2023) pro-

Table 4: A query expansion example from the TREC-DL-2020 dataset using MILL.

Query	What is a nonconformity earth science
Expanded query	Pseudo-relevance document: There are three types of unconformities(below): angular **unconformity**, **nonconformity**, and disconformity. Anangular **unconformity** (left) is an erosional **surface** separating steeply dipping **rock** layers below from gently dipping layers above. **Unconformity** is the general term for **missing** pages of Earth history. LLM-generated document: **1.** What is the definition of a **nonconformity** in earth science? A **nonconformity** in earth science refers to a type of **unconformity**, which is a **break in the geologic record** where layers of **rock** are **missing**. Specifically, a **nonconformity** is where **sedimentary rocks** are deposited on top of **igneous or metamorphic rocks**. **2.** How are **nonconformities** formed inearth science? **Nonconformities** are formed throughthe process of **erosion**, where layers of **rock** are gradually wornaway. This exposes the underlying **igneous or metamorphic rocks**, which are then **covered by** new **sedimentary rocks** over time. **3.** What are the different types of **nonconformities** in earth science? There are three main types of **nonconformities**: angular, **nonconformity**, and disconformity. Angular **nonconformities occur** when **sedimentary rocks** are deposited at an angle on top of tilted or folded **igneous or metamorphic rocks**. **Nonconformities occur** when **sedimentary rocks** are deposited on top of **eroded igneous ormetamorphic rocks**. Disconformities **occur** when there is a gap in the **sedimentary rock** layers, indicating **a period of erosion** or non-deposition. **4.** What are some examples of **nonconformities found** in the Earth's geology? **One well-known example** of a **nonconformity** is the **Grand Canyon** in Arizona, USA. The **sedimentary layers** seen in the **canyon** were deposited on top of tilted and folded igneous and **metamorphic rocks**, indicating a long history of **erosion and deposition**.
Ground Truth	**Nonconformities** are **unconformities** that separate **igneous or metamorphic rocks** from overlying **sedimentary rocks**. They usually indicate that along **period of erosion occurred** prior to **deposition of the sediments** (several km of **erosion** necessary). They are a feature of stratified **rocks**, and are therefore usually found in **sediments** (but may also occur in stratified volcanics). They are **surfaces** between two **rock** bodies that constitute a substantial **break (hiatus) in the geologic record** (sometimes people say inaccurately that time is **missing**). **Nonconformity**. When **igneous or metamorphic rocks** are **eroded** and then **covered by** younger **sedimentary rocks**, the contact is called a **nonconformity**. **One of the most famous** of these is **found in the Grand Canyon**, where the oldest **sedimentary rocks** are more than a billion years younger than the 1.6 billion-year-old **metamorphic rocks** on which they rest.

poses a query-document prompt framework, leveraging the semantic understanding and generative capabilities of LLMs to extend the original query. Another recent study (Jagerman et al., 2023) applies LLMs directly for query expansion across multiple datasets, finding that employing the chain of thoughts (CoT) (Wei et al., 2022b) approach achieves the best results. Moreover, LLMCS (Mao et al., 2023) applies LLMs for query expansion in conversational search, constructing the context search intents as a prompt and combining the chain of thoughts and self-consistency techniques to enhance search performance. In our paper, we focus on alleviating the limitations of both PRF-based and generation-based method. We propose a query-query-document generation method and a mutual verification framework to effective leverage both retrieved and generated contextual documents.

5.2 Large Language Models

Large Language Models (LLMs) have strong and robust abilities in language understanding and generation (Zhao et al., 2023; Kojima et al., 2022; Huang et al., 2022; Wang et al., 2022), especially with increased model parameters (Zhao et al., 2023; Jagerman et al., 2023; Wei et al., 2022a). LLMs have the instruction-following ability (Longpre et al., 2023; Wei et al., 2021) and can be boasted through a few contexts (Min et al., 2022; Dong et al., 2022), enhancing the performance of LLMs in downstream specific tasks. Moreover, these methods are straightforward and effective, for they require minimal human effort to provide instructions or in-context examples but reach good results. For example, Flan-T5 (Chung et al., 2022) achieves remarkable results in various NLP downstream tasks by instruction tuning the base model. Recently, many studies (Wei et al., 2022b; Besta et al., 2023; Yao et al., 2023; Wang et al., 2022) explored the reasoning capabilities of LLMs and discovered that LLMs are powerful zero-shot reasoners. Chain of thoughts (Wei et al., 2022b) (CoT) prompts LLMs to think step by step to activate reasoning capabilities in LLMs. Self-consistency (Wang et al., 2022) runs multiple CoT and takes a voting mechanism to enhance reasoning accuracy.

6 Conclusion

In this paper, we propose a novel zero-shot Large Language Models (LLMs) based framework for query expansion. First, we design a query-query-document prompt scheme that allows LLMs to generate diverse contextual documents via zero-shot reasoning. Next, we introduce a mutual verification method that allows retrieved and generated contextual documents to complement each other as query expansion. The experimental results show that our method is superior to the state-of-the-art baselines on three public datasets.

7 Limitations

One limitation of our work is the retrieval efficiency. On one hand, during retrieval, MILL needs to perform multiple autoregressive generations for each query based on the query-query-document prompt, and then use mutual verification methods with PRF documents to obtain selected documents. On the other hand, the extended length of the query increases the time required to search the inverted index. To address the issue of multi-round autoregressive generation, N generated documents can be produced in parallel, which will improve generation efficiency. Regarding the issue of extended query length, we can further utilize simple rule-based filtering methods (e.g., deleting words with limited semantic information or truncating documents with word counts) to compress the query.

References

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, page 189.

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):357–389.

Hiteshwar Kumar Azad and Akshay Deepak. 2019. Query expansion techniques for information retrieval: a survey. Information Processing & Management, 56(5):1698–1735.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Michal Podstawski, Hubert Niewiadomski, Piotr Nyczyk, et al. 2023. Graph of thoughts: Solving elaborate problems with large language models. arXiv preprint arXiv:2308.09687.

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

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, 33:1877–1901.

Guihong Cao, Jian-Yun Nie, Jianfeng Gao, and Stephen Robertson. 2008. Selecting good expansion terms for pseudo-relevance feedback. In Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, pages 243–250.

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

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

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, pages 202–209.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. 2021. Overview of the trec 2020 deep learning track. arXiv preprint arXiv:2102.07662.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. 2022. A survey for in-context learning. arXiv preprint arXiv:2301.00234.

Efthimis N Efthimiadis. 1996. Query expansion. Annual review of information science and technology (ARIST), 31:121–87.

Hui Fang and ChengXiang Zhai. 2006. Semantic term matching in axiomatic approaches to information retrieval. In Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, pages 115–122.

Jiaxin Huang, Shixiang Shane Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. 2022. Large language models can self-improve. arXiv preprint arXiv:2210.11610.

Rolf Jagerman, Honglei Zhuang, Zhen Qin, Xuanhui Wang, and Michael Bendersky. 2023. Query expansion by prompting large language models. arXiv preprint arXiv:2305.03653.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. Advances in neural information processing systems, 35:22199–22213.

Hang Li, Shengyao Zhuang, Ahmed Mourad, Xueguang Ma, Jimmy Lin, and Guido Zuccon. 2022. Improving query representations for dense retrieval with pseudo relevance feedback: A reproducibility study. In European Conference on Information Retrieval, pages 599–612. Springer.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, et al. 2023. The flan collection: Designing data and methods for effective instruction tuning. arXiv preprint arXiv:2301.13688.

Yuanhua Lv and ChengXiang Zhai. 2010. Positional relevance model for pseudo-relevance feedback. In Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval, pages 579–586.

Craig Macdonald and Nicola Tonellotto. 2020. Declarative experimentation ininformation retrieval using pyterrier. In Proceedings of ICTIR 2020.

Kelong Mao, Zhicheng Dou, Haonan Chen, Fengran Mo, and Hongjin Qian. 2023. Large language models know your contextual search intent: A prompting framework for conversational search. arXiv preprint arXiv:2303.06573.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? In EMNLP.

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.

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, pages 160–169.

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

Joseph John Rocchio Jr. 1971. Relevance feedback in information retrieval. The SMART retrieval system: experiments in automatic document processing.

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).

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, pages 61–69. Springer.

Liang Wang, Nan Yang, and Furu Wei. 2023. Query2doc: Query expansion with large language models. arXiv preprint arXiv:2303.07678.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. arXiv preprint arXiv:2203.11171.

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.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022a. Emergent abilities of large language models. arXiv preprint arXiv:2206.07682.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022b. Chain-of-thought prompting elicits reasoning in large language models. Advances in Neural Information Processing Systems, 35:24824–24837.

Rong Yan, Alexander Hauptmann, and Rong Jin. 2003. Multimedia search with pseudo-relevance feedback. In Image and Video Retrieval: Second International Conference, CIVR 2003 Urbana-Champaign, IL, USA, July 24–25, 2003 Proceedings 2, pages 238–247. Springer.

Peilin Yang and Jimmy Lin. 2019. Reproducing and generalizing semantic term matching in axiomatic information retrieval. In Advances in Information Retrieval: 41st European Conference on IR Research, ECIR 2019, Cologne, Germany, April 14–18, 2019, Proceedings, Part I 41, pages 369–381. Springer.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. arXiv preprint arXiv:2305.10601.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. arXiv preprint arXiv:2303.18223.

A Appendix

A.1 Baselines

There are two groups of baseline methods in our experiments: traditional query expansion methods and LLM-based expansion methods.

Traditional query expansion methods

- **Bo1 (Amati and Van Rijsbergen, 2002).** The Bose-Einstein 1 (Bo1) weighting approach is a method that reconstructs the query based on the frequency of terms found in the feedback documents associated with each query.

- **KL (Amati and Van Rijsbergen, 2002).** This method rewrites the queries similar to Bo1 but based on Kullback Leibler divergence.

- **RM3 (Abdul-Jaleel et al., 2004).** A method used for query expansion in information retrieval, which finds the most relevant terms to the query by using the top-ranked documents returned from the initial query and adds these terms to the original query to create an expanded query.

- **AxiomaticQE (Fang and Zhai, 2006; Yang and Lin, 2019).** Axiomatic query expansion (AxiomaticQE) rewrites and expands the origin query by axiomatic semantic term matching.

LLM-based expansion methods

- **Query2Term.** It uses LLMs to generate related terms to the origin query in a zero-shot manner. The zero-shot prompts only contain task instructions and the original query.

- **Query2Term-FS.** The few-shot version of Query2Term. The few-shot prompts are built upon zero-shot prompts by adding a few examples. In particular, Query2Term-FS expands upon Query2Term by incorporating additional sets of query-keywords examples.

- **Query2Term-PRF.** It uses the top-3 documents retrieved by the original query as context information to instruct the LLMs to expand the original query.

- **Query2Doc.** The zero-shot version of query2doc (Wang et al., 2023), whose structure is similar to Query2Term. It uses LLMs to generate related passages to the origin query.

- **Query2Doc-FS.** The few-shot version of query2doc (Wang et al., 2023). The prompt structure is similar to Query2Term-FS.

- **Query2Doc-PRF.** It constructs the prompt with pseudo-relevance feedback in a zero-shot manner based on Query2Doc-ZS, like the Query2Term-PRF.

- **CoT.** Chain-of-Thought (CoT) (Jagerman et al., 2023) instructs LLMs to generate text step by step, providing a detailed thought process before generating the final answer.

- **CoT-PRF.** A pseudo-relevance feedback based version of CoT similar to Query2Term-PRF.

A.2 Prompts

In this subsection, we will detail the prompts we used in the experiments.

Figure 5 shows the prompts for the variants of Query2Term. The core prompt is "Write some keywords for the given query: {query}."

Table 5: Prompts for Query2Term and its variants.

Method	Prompt
Query2Term	Write some keywords for the given query: {query}

Query2Term-FS	Write some keywords for the given query:
	
	Context:
	query:{query1}
	keywords:{keywords1}
	query:{query2}
	keywords: {keywords2}
	query: {query3}
	keywords:{keywords3}
	
	query: {query}
	keywords:

Query2Term-PRF	Write some keywords for the given query:
	
	Context:
	{PRF doc 1}
	{PRF doc 2}
	{PRF doc 3}
	
	query: {query}
	keywords:

Figure 6 shows the prompts for the Query2Doc variants. The main prompts are the sentence: "Write a passage answer the following query: {query}."

For the CoT and its variants, their prompts are in Figure 7. The prompts ask LLMs to give the rationale before answering.

A.3 More Results on BEIR

In this section, we list the full results for the 7 selected datasets from BEIR. Specifically, they

Table 6: Prompts for Query2Doc and its variants.

Method	Prompt
Query2Doc	Write a passage answer the following query: {query}

Query2Doc-FS	Write a passage answer the following query:
	
	Context:
	query:{query1}
	passage:{passage1}
	query:{query2}
	passage: {passage2}
	query: {query3}
	passage:{passage3}
	
	query: {query}
	passage:

Query2Doc-PRF	Write a passage answer the following query:
	
	Context:
	{PRF doc 1}
	{PRF doc 2}
	{PRF doc 3}
	
	query: {query}
	passage:

Table 7: Prompts for CoT and its variants.

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

CoT-PRF	Answer the following query:
	
	Context:
	{PRF doc 1}
	{PRF doc 2}
	{PRF doc 3}
	
	query: {query}
	Give the rationale before answering.

are TREC-COVID, TOUCHE, SCIFACT, NFCORPUS, DBPEDIA, FIQA-2018, and SCIDOCS. The optimal results are highlighted in bold, while the suboptimal results are underscored. The results are reported on NDCG@N, AP@N, Recall@N, and MRR@N with N (10, 100, 1000)

A.4 More Results for Experiments with Various Numbers of Documents

In this subsection, we will supplement the results on other metrics for the experiments with various numbers of documents. We use the gpt-3.5-turbo-instruct API provided by OpenAI to conduct these experiments.

The experiments concerning the number of selected documents are shown in Figure 5. When the number of selected generated documents changes, the number of candidate generated documents remains 15, and the number of PRF candidate documents and the number of selected PRF documents remain 5 and 3. When the number of selected PRF documents changes, the number of candidate PRF documents remains 15, and the number of generated candidate documents and the number of selected generated documents remain 5 and 3. We can find that the trends of selected PRF documents in NDCG, AP, and Recall are consistent, yet contrary to that of MRR. This is due to the fact that NDCG, AP, and Recall are more comprehensive indicators, whereas MRR only considers the ranking of the topmost relevant document retrieved.

In the experiments regarding the number of candidate documents, as shown in Figure 6, we can observe a similar trend across different metrics: as the number of generated document candidates increases, the metrics remain relatively stable. However, with an increase in the number of PRF document candidates, there is a noticeable growth in the metrics. This suggests that a specific number of generated documents, such as 5, can almost entirely cover the additional information provided by the generation process to aid in understanding the search intent of the original query. Meanwhile, PRF documents, derived from searches based on the original query, suggest that more PRF document candidates can cover a wider range of possible search intents, thereby enhancing the effectiveness of query expansion.

Table 8: Overall experimental results on TREC-COVID.

	Metrics	NDCG			AP			Recall			MRR		
		@10	@100	@1000	@10	@100	@1000	@10	@100	@1000	@10	@100	@1000
TREC-COVID	No expansion	62.59	47.41	42.04	1.46	8.16	19.79	1.74	11.91	40.52	83.37	83.37	83.37
	Traditional expansion methods												
	Bo1	64.82	49.50	44.73	1.56	8.80	22.01	1.77	12.48	43.64	86.62	86.77	86.77
	KL	65.80	49.93	44.88	1.59	8.90	22.26	1.79	12.51	43.63	86.62	86.79	86.79
	RM3	64.05	48.50	44.54	1.55	8.62	21.87	1.78	11.22	43.71	82.96	83.06	83.06
	AxiomaticQE	62.74	47.45	42.06	1.47	8.17	19.81	1.74	11.91	40.53	84.37	84.37	84.37
	LLM-based expansion methods												
	Query2Term	66.81	50.15	45.09	1.65	8.95	22.10	1.86	12.40	43.67	83.33	83.43	83.43
	Query2Term-FS	64.39	49.71	44.86	1.58	8.75	21.72	1.82	12.42	43.89	85.00	85.05	85.05
	Query2Term-PRF	61.80	47.34	42.94	1.53	8.28	20.84	1.75	11.69	41.99	84.55	84.55	84.55
	Query2Doc	69.00	50.82	45.41	1.73	9.32	22.36	1.95	12.63	43.71	86.21	86.39	86.39
	Query2Doc-FS	68.40	49.67	44.39	1.70	8.79	21.57	1.91	12.15	42.80	69.32	89.38	89.38
	Query2Doc-PRF	71.32	54.58	47.97	1.75	10.14	24.71	2.02	13.55	46.20	84.56	84.69	84.69
	CoT	72.58	52.99	46.93	1.85	9.89	23.63	2.04	13.15	45.01	88.06	88.17	88.17
	CoT-PRF	68.58	52.37	46.55	1.75	9.56	23.42	2.00	13.04	44.93	89.25	89.25	89.25
	MILL	**73.05**	**58.06**	**51.17**	**1.91**	**11.40**	**27.76**	**2.11**	**14.65**	**49.33**	**89.63**	**89.63**	**89.63**

Table 9: Overall experimental results on TOUCHE.

	Metrics	NDCG			AP			Recall			MRR		
		@10	@100	@1000	@10	@100	@1000	@10	@100	@1000	@10	@100	@1000
Touche	No expansion	34.28	45.48	55.32	13.06	20.96	22.47	20.69	54.92	85.05	62.28	62.71	62.71
	Traditional expansion methods												
	Bo1	35.62	46.98	56.62	14.19	22.19	23.69	21.35	56.47	86.00	63.54	64.07	64.07
	KL	35.52	46.96	56.72	14.00	22.18	23.68	20.99	56.78	86.14	63.98	64.51	64.51
	RM3	34.66	46.54	55.79	13.72	22.00	23.42	22.03	57.79	85.79	56.73	57.09	57.09
	AxiomaticQE	34.28	45.48	55.32	13.06	20.96	22.47	20.69	54.92	85.05	62.28	62.71	62.71
	LLM-based expansion methods												
	Query2Term	34.51	44.05	52.95	13.11	19.88	21.13	20.05	49.98	77.24	65.60	66.13	66.14
	Query2Term-FS	35.10	47.93	57.10	14.97	23.28	24.66	21.71	57.88	**85.33**	57.71	58.23	58.23
	Query2Term-PRF	31.83	44.19	53.72	12.60	19.78	21.22	20.16	53.83	83.29	54.77	55.45	55.45
	Query2Doc	42.36	51.12	60.32	**17.44**	25.51	26.91	23.80	56.10	84.08	**75.63**	**75.97**	**75.97**
	Query2Doc-FS	40.71	51.30	59.99	16.91	25.72	27.02	23.01	57.46	83.95	70.84	71.06	71.06
	Query2Doc-PRF	37.21	47.43	56.84	14.78	22.39	23.81	21.11	54.30	83.50	69.59	69.95	69.97
	CoT	41.91	51.57	60.77	17.28	25.61	27.03	23.18	56.42	84.42	*75.00*	75.09	75.09
	CoT-PRF	39.33	50.08	59.03	16.66	24.54	25.93	23.30	57.10	84.37	69.45	69.58	69.58
	MILL	**43.22**	**53.05**	**61.29**	17.12	**26.31**	**27.68**	**24.43**	**59.55**	84.99	74.01	74.01	74.01

Table 10: Overall experimental results on SCIFACT.

	Metrics	NDCG			AP			Recall			MRR		
		@10	@100	@1000	@10	@100	@1000	@10	@100	@1000	@10	@100	@1000
SCIFACT	No expansion	67.22	69.66	70.27	62.11	62.67	62.70	81.43	92.27	97.00	63.24	63.66	63.68
	Traditional expansion methods												
	Bo1	65.14	67.63	68.34	59.30	59.92	59.95	81.59	92.20	97.67	60.42	60.87	60.89
	KL	64.68	67.08	67.83	58.69	59.28	59.31	81.59	91.87	97.67	59.76	60.18	60.21
	RM3	62.22	64.54	65.28	55.45	55.97	55.99	81.34	91.93	97.67	56.24	56.58	56.61
	AxiomaticQE	67.22	69.66	70.28	62.11	62.68	62.70	81.43	92.27	97.00	63.24	63.66	63.68
	LLM-based expansion methods												
	Query2Term	66.13	68.87	69.57	60.54	61.18	61.21	81.70	93.73	99.00	61.60	62.14	62.16
	Query2Term-FS	68.34	70.71	71.39	62.92	63.50	63.54	83.32	93.47	98.33	64.13	64.60	64.62
	Query2Term-PRF	57.67	59.91	60.79	49.72	50.22	50.25	80.46	90.90	97.50	50.58	50.93	50.96
	Query2Doc	67.92	70.60	71.19	62.59	63.24	63.27	82.82	94.43	99.00	63.81	64.34	64.36
	Query2Doc-FS	68.61	71.39	71.89	63.37	64.02	64.04	83.17	**95.43**	99.33	64.55	65.07	65.08
	Query2Doc-PRF	64.53	66.96	67.82	58.60	59.15	59.19	81.31	92.53	99.00	59.74	60.12	60.15
	CoT	68.58	71.13	71.63	63.30	63.87	63.89	83.03	94.77	98.67	64.77	65.18	65.19
	CoT-PRF	70.98	72.95	73.65	66.20	66.64	66.67	84.56	93.27	98.67	67.09	67.47	67.49
	MILL	**71.37**	**73.47**	**74.14**	**66.34**	**66.85**	**66.88**	**85.24**	94.50	**99.67**	**67.69**	**68.07**	**68.09**

Table 11: Overall experimental results on NFCORPUS.

Metrics		NDCG			AP			Recall			MRR		
		@10	@100	@1000	@10	@100	@1000	@10	@100	@1000	@10	@100	@1000
NFCORPUS	No expansion	32.22	27.29	30.02	12.08	14.36	14.89	14.78	24.38	36.06	53.44	53.82	53.83
	Traditional expansion methods												
	Bo1	33.49	30.21	37.01	12.73	15.98	17.09	16.26	29.71	54.38	52.74	53.24	53.28
	KL	33.56	30.22	37.18	12.73	15.89	17.01	16.3	29.61	54.79	53.49	53.99	54.03
	RM3	33.41	30.31	37.27	12.36	15.68	16.8	16.82	30.46	56.12	52.35	52.81	52.85
	AxiomaticQE	32.22	27.29	30.02	12.08	14.36	14.89	14.78	24.38	36.06	53.44	53.82	53.83
	LLM-based expansion methods												
	Query2Term	25.79	24.94	33.82	8.3	10.89	12.04	12.29	27.27	58.82	44.79	45.63	45.68
	Query2Term-FS	31.92	30.66	38.57	11.24	14.63	15.91	15.38	**32.83**	61.72	52.99	53.68	53.71
	Query2Term-PRF	32.14	29.92	38.21	11.92	15.01	16.29	16.78	31.63	60.55	49.27	49.83	49.87
	Query2Doc	33.47	30.41	38.76	12.54	15.31	16.54	16.68	30.96	61.09	54.61	55.19	55.23
	Query2Doc-FS	33.41	30.1	38.09	12.59	15.32	16.45	16.27	30.22	59.55	54.08	54.64	54.7
	Query2Doc-PRF	33.82	31.23	39.41	12.64	16.17	17.44	16.97	32.7	62.5	51.26	51.72	51.77
	CoT	34.52	30.68	38.88	12.95	15.78	16.93	16.88	29.53	60.63	56.23	56.64	56.69
	CoT-PRF	35.76	31.93	39.84	**13.95**	16.9	18.09	18.13	31.76	59.87	55.65	56.05	56.09
	MILL	**36.79**	**33.02**	**41.75**	13.81	**17.18**	**18.56**	**18.21**	32.42	**64.95**	**58.35**	**58.86**	**58.91**

Table 12: Overall experimental results on DBPEDIA.

Metrics		NDCG			AP			Recall			MRR		
		@10	@100	@1000	@10	@100	@1000	@10	@100	@1000	@10	@100	@1000
DBPEDIA	No expansion	26.59	32.45	38.7	11.59	17.71	18.89	17.2	42.15	63.61	51.7	52.37	52.39
	Traditional expansion methods												
	Bo1	26.59	32.59	39.05	11.65	18.03	19.24	17.32	42.67	64.9	50.47	51.17	51.2
	KL	26.42	32.44	38.87	11.52	17.89	19.09	17.27	42.62	64.71	50.01	50.84	50.86
	RM3	25.47	31.81	38.11	10.88	17.4	18.6	17.05	42.92	64.37	46.6	47.28	47.31
	AxiomaticQE	26.59	32.45	38.7	11.59	17.71	18.89	17.2	42.15	63.61	51.7	52.37	52.39
	LLM-based expansion methods												
	Query2Term	22.1	26.59	33.51	9.16	13.54	14.54	14.11	34.63	58.9	46.54	47.16	47.2
	Query2Term-FS	26.46	31.9	39.36	11.87	17.04	18.29	17.67	41.59	65.67	53.5	54.16	54.19
	Query2Term-PRF	23.39	27.85	34.83	9.98	14.79	15.96	16.1	37.15	61.11	45.37	46.03	46.07
	Query2Doc	32.31	37.72	44.79	14.27	20.65	21.97	20.13	46.37	70.29	61.82	62.32	62.34
	Query2Doc-FS	32.87	37.99	45.11	14.65	20.86	22.16	19.65	45.85	70.04	63.35	63.82	63.84
	Query2Doc-PRF	27.43	33.22	39.85	11.53	18.11	19.34	18.74	44.23	66.41	52.58	53.26	53.28
	CoT	29.96	36.01	43.05	13.29	19.42	20.7	19.22	45.76	69.24	57.68	58.3	58.32
	CoT-PRF	28.17	33.66	40.43	12.26	18.49	19.75	18.15	43.43	66.06	52.95	53.59	53.6
	MILL	**34.33**	**39.71**	**46.39**	**15.65**	**22.89**	**24.28**	**21.32**	**48.86**	**71.13**	**64.09**	**64.53**	**64.55**

Table 13: Overall experimental results on FIQA-2018.

Metrics		NDCG			AP			Recall			MRR		
		@10	@100	@1000	@10	@100	@1000	@10	@100	@1000	@10	@100	@1000
FIQA-2018	No expansion	25.26	31.74	35.28	19.40	20.86	21.04	30.97	55.92	77.42	31.03	32.11	32.18
	Traditional expansion methods												
	Bo1	24.36	31.21	34.97	18.71	20.30	20.49	30.21	56.25	79.18	29.37	30.51	30.58
	KL	24.75	31.40	35.12	18.99	20.52	20.72	30.88	56.21	78.84	29.77	30.84	30.92
	RM3	22.8	29.23	33.14	16.85	18.32	18.51	30.37	54.82	78.82	26.47	27.55	27.63
	AxiomaticQE	25.26	31.76	35.28	19.40	20.87	21.04	30.97	56.00	77.42	31.03	32.11	32.18
	LLM-based expansion methods												
	Query2Term	21.72	28.1	32.12	16.15	17.45	17.65	28.42	54.12	78.22	25.82	26.83	26.91
	Query2Term-FS	24.83	31.95	35.78	18.90	20.49	20.68	30.50	58.45	81.84	30.57	31.61	31.68
	Query2Term-PRF	21.56	27.43	31.50	16.29	17.55	17.73	27.47	50.78	76.31	25.32	26.21	26.29
	Query2Doc	27.00	33.92	37.63	20.46	22.15	22.34	34.26	60.11	82.72	32.64	33.73	33.78
	Query2Doc-FS	27.23	34.46	37.96	20.37	22.15	22.33	34.80	61.94	83.46	33.14	34.23	34.29
	Query2Doc-PRF	23.51	30.26	34.09	17.91	19.39	19.57	28.99	55.33	79.14	29.18	30.19	30.27
	CoT	26.69	33.78	37.28	19.8	21.48	21.65	**34.88**	62.34	83.56	32.12	33.16	33.22
	CoT-PRF	27.78	34.30	38.04	21.45	23.06	23.24	34.50	59.26	82.14	33.25	34.21	34.29
	MILL	**28.42**	**35.63**	**39.23**	**21.89**	**23.61**	**23.80**	34.63	**62.46**	**84.23**	**34.94**	**35.99**	**36.05**

Table 14: Overall experimental results on SCIDOCS.

	Metrics	NDCG			AP			Recall			MRR		
		@10	@100	@1000	@10	@100	@1000	@10	@100	@1000	@10	@100	@1000
SCIDOCS	No expansion	14.71	20.91	25.14	8.36	9.73	9.94	15.84	34.48	55.04	25.37	26.41	26.48
	Traditional expansion methods												
	Bo1	15.10	21.82	26.14	8.73	10.29	10.51	16.43	36.39	57.47	25.31	26.41	26.48
	KL	15.10	21.81	26.15	8.75	10.31	10.54	16.37	36.24	57.38	25.43	26.54	26.61
	RM3	14.56	21.49	25.91	8.41	10.05	10.28	15.79	36.24	57.88	24.46	25.63	25.70
	AxiomaticQE	14.71	20.91	25.14	8.36	9.73	9.94	15.84	34.48	55.04	25.37	26.41	26.48
	LLM-based expansion methods												
	Query2Term	13.04	20.02	25.11	7.32	8.84	9.10	14.30	35.08	60.00	22.34	23.66	23.73
	Query2Term-FS	14.16	21.25	26.18	8.07	9.68	9.94	15.26	36.21	60.15	24.31	25.54	25.62
	Query2Term-PRF	13.10	20.13	24.97	7.49	9.12	9.37	14.84	35.56	59.25	20.54	21.84	21.91
	Query2Doc	15.09	22.63	27.40	8.57	10.34	10.59	16.13	38.31	61.63	26.21	27.49	27.55
	Query2Doc-FS	15.06	22.35	27.18	8.43	10.16	10.43	16.49	37.94	61.33	25.83	27.01	27.08
	Query2Doc-PRF	14.30	21.50	26.16	8.21	9.96	10.21	15.70	36.78	59.50	23.84	25.03	25.11
	CoT	15.54	22.77	27.50	8.90	10.58	10.84	16.65	37.96	60.90	26.81	28.07	28.13
	CoT-PRF	14.71	21.66	26.23	8.44	10.10	10.34	16.05	36.50	58.72	24.77	25.91	25.98
	MILL	**16.38**	**23.73**	**28.36**	**9.50**	**11.23**	**11.48**	**17.49**	**39.28**	**61.86**	**28.10**	**29.25**	**29.31**

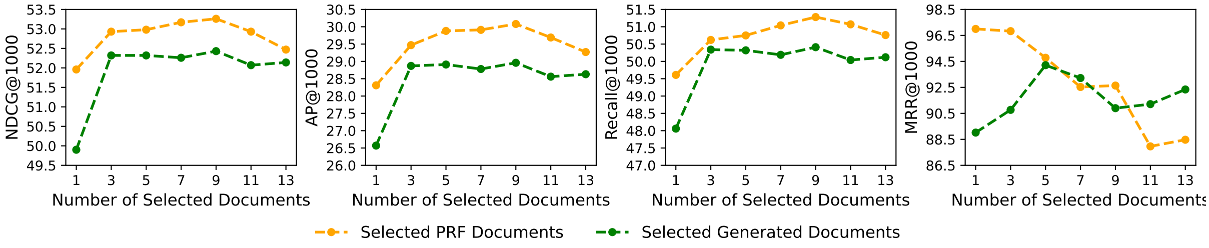


Figure 5: Hyperparameter analysis on the number of document selections. The x-axis denotes the number of document selected, and the y-axis represents the metrics values (NDCG@1000, AP@1000, Recall@1000, and MRR@1000).

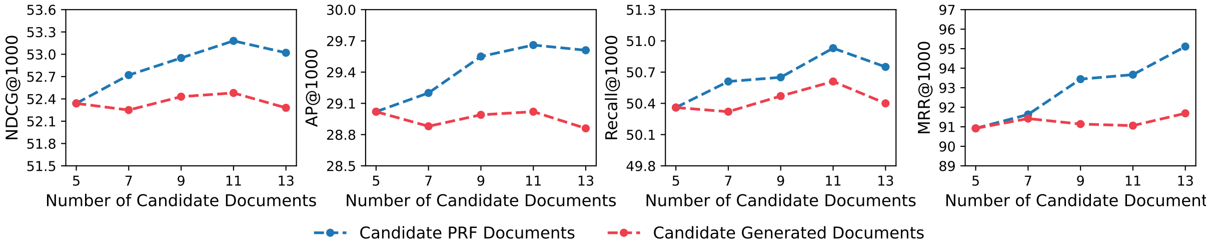


Figure 6: Hyperparameter analysis on the number of document candidates. The x-axis denotes the number of document candidates, and the y-axis represents the metrics values (NDCG@1000, AP@1000, Recall@1000, and MRR@1000).