Abstract

This paper introduces a simple yet effective query expansion approach, denoted as query2doc, to improve both sparse and dense retrieval systems. The proposed method first generates pseudo-documents by few-shot prompting large language models (LLMs), and then expands the query with generated pseudo-documents. LLMs are trained on web-scale text corpora and are adept at knowledge memorization. The pseudo-documents from LLMs often contain highly relevant information that can aid in query disambiguation and guide the retrievers. Experimental results demonstrate that query2doc boosts the performance of BM25 by 3% to 15% on ad-hoc IR datasets, such as MS-MARCO and TREC DL, without any model fine-tuning. Furthermore, our method also benefits state-of-the-art dense retrievers in terms of both in-domain and out-of-domain results.

1 Introduction

Information retrieval (IR) aims to locate relevant documents from a large corpus given a user issued query. It is a core component in modern search engines and researchers have invested for decades in this field. There are two mainstream paradigms for IR: lexical-based sparse retrieval, such as BM25, and embedding-based dense retrieval (Xiong et al., 2021; Qu et al., 2021). Although dense retrievers perform better when large amounts of labeled data are available (Karpukhin et al., 2020), BM25 remains competitive on out-of-domain datasets (Thakur et al., 2021).

Query expansion (Rocchio, 1971; Lavrenko and Croft, 2001) is a long-standing technique that rewrites the query based on pseudo-relevance feedback or external knowledge sources such as WordNet. For sparse retrieval, it can help bridge the lexical gap between the query and the documents. However, query expansion methods like RM3 (Lavrenko and Croft, 2001; Lv and Zhai, 2009) have only shown limited success on popular datasets (Campos et al., 2016), and most state-of-the-art dense retrievers do not adopt this technique. In the meantime, document expansion methods like doc2query (Nogueira et al., 2019) have proven to be effective for sparse retrieval.

In this paper, we demonstrate the effectiveness of LLMs (Brown et al., 2020) as query expansion models by generating pseudo-documents conditioned on few-shot prompts. Given that search queries are often short, ambiguous, or lack necessary background information, LLMs can provide relevant information to guide retrieval systems, as they memorize an enormous amount of knowledge and language patterns by pre-training on trillions of tokens.

Our proposed method, called query2doc, generates pseudo-documents by few-shot prompting LLMs and concatenates them with the original query to form a new query. This method is simple to implement and does not require any changes in training pipelines or model architectures, making it orthogonal to the progress in the field of LLMs and information retrieval. Future methods can easily build upon our query expansion framework.

For in-domain evaluation, we adopt the MS-MARCO passage ranking (Campos et al., 2016), TREC DL 2019 and 2020 datasets. Pseudo-documents are generated by prompting an improved version of GPT-3 text-davinci-003 from OpenAI (Brown et al., 2020). Results show that query2doc substantially improves the off-the-shelf BM25 algorithm without fine-tuning any model, particularly for hard queries from the TREC DL track. Strong dense retrievers, including DPR (Karpukhin et al., 2020), SimLM (Wang et al., 2023), and E5 (Wang et al., 2022) also benefit from query2doc, although the gains tend to be diminishing when distilling from a strong cross-encoder based re-ranker. Experiments in zero-shot OOD settings demonstrate that
our method outperforms strong baselines on most datasets. Further analysis also reveals the importance of model scales: query2doc works best when combined with the most capable LLMs while small language models only provide marginal improvements over baselines. To aid reproduction, we release all the generations from text-davinci-003 at https://huggingface.co/datasets/intfloat/query2doc_msmarco.

2 Method

Given a query $q$, we employ few-shot prompting to generate a pseudo-document $d'$ as depicted in Figure 1. The prompt comprises a brief instruction “Write a passage that answers the given query.” and $k$ labeled pairs randomly sampled from a training set. We use $k = 4$ throughout this paper. Subsequently, we rewrite $q$ to a new query $q^+$ by concatenating with the pseudo-document $d'$. There are slight differences in the concatenation operation for sparse and dense retrievers, which we elaborate on in the following section. The population is primarily white...

**Sparse Retrieval** Since the query $q$ is typically much shorter than pseudo-documents, to balance the relative weights of the query and the pseudo-document, we boost the query term weights by repeating the query $n$ times before concatenating with the pseudo-document $d'$:

$$q^+ = \text{concat}(q, n, d')$$  (1)

Here, “concat” denotes the string concatenation function. $q^+$ is used as the new query for BM25 retrieval. We find that $n = 5$ is a generally good value and do not tune it on a dataset basis.

**Dense Retrieval** The new query $q^+$ is a simple concatenation of the original query $q$ and the pseudo-document $d'$ separated by [SEP]:

$$q^+ = \text{concat}(q, \text{[SEP]}, d')$$  (2)

For training dense retrievers, several factors can influence the final performance, such as hard negative mining (Xiong et al., 2021), intermediate pretraining (Gao and Callan, 2021), and knowledge distillation from a cross-encoder based re-ranker (Qu et al., 2021). In this paper, we investigate two settings to gain a more comprehensive understanding of our method. The first setting is training DPR (Karpukhin et al., 2020) models initialized from BERT$_{base}$ with BM25 hard negatives only. The optimization objective is a standard contrastive loss:

$$L_{cont} = -\log \frac{e^{h_q \cdot h_d}}{\sum_{d_i \in \mathbb{N}} e^{h_q \cdot h_{d_i}}}$$  (3)

where $h_q$ and $h_d$ represent the embeddings for the query and document, respectively. $\mathbb{N}$ denotes the set of hard negatives.

The second setting is to build upon state-of-the-art dense retrievers and use KL divergence to distill from a cross-encoder teacher model.

$$\min D_{KL}(p_{ce}, p_{stu}) + \alpha L_{cont} \quad (4)$$

$p_{ce}$ and $p_{stu}$ are the probabilities from the cross-encoder and our student model, respectively. $\alpha$ is a coefficient to balance the distillation loss and contrastive loss.

**Comparison with Pseudo-relevance Feedback** Our proposed method is related to the classic method of pseudo-relevance feedback (PRF) (Lavrenko and Croft, 2001; Lv and Zhai, 2009). In conventional PRF, the feedback signals for query expansion come from the top-k documents obtained in the initial retrieval step, while our method prompts LLMs to generate pseudo-documents. Our method does not rely on the quality of the initial retrieval results, which are often noisy or irrelevant. Rather, it exploits cutting-edge LLMs to generate documents that are more likely to contain relevant terms.
Table 1: Main results on the MS-MARCO passage ranking and TREC datasets. The “Fine-tuning” column indicates whether the method requires fine-tuning model on labeled data or not. ∗: our reproduction.

3 Experiments

3.1 Setup

Evaluation Datasets For in-domain evaluation, we utilize the MS-MARCO passage ranking (Campos et al., 2016), TREC DL 2019 (Craswell et al., 2020a) and 2020 (Craswell et al., 2020b) datasets. For zero-shot out-of-domain evaluation, we select five low-resource datasets from the BEIR benchmark (Thakur et al., 2021). The evaluation metrics include MRR@10, R@k (k ∈ {50, 1k}), and nDCG@10.

Hyperparameters For sparse retrieval including BM25 and RM3, we adopt the default implementation from Pyserini (Lin et al., 2021). When training dense retrievers, we use mostly the same hyperparameters as SimLM (Wang et al., 2023), with the exception of increasing the maximum query length to 144 to include pseudo-documents. When prompting LLMs, we include 4 in-context examples and use the default temperature of 1 to sample at most 128 tokens. For further details, please refer to Appendix A.

3.2 Main Results

In Table 1, we list the results on the MS-MARCO passage ranking and TREC DL datasets. For sparse retrieval, “BM25 + query2doc” beats the BM25 baseline with over 15% improvements on TREC DL 2019 and 2020 datasets. Our manual inspection reveals that most queries from the TREC DL track are long-tailed entity-centric queries, which benefit more from the exact lexical match. The traditional query expansion method RM3 only marginally improves the R@1k metric. Although the document expansion method docT5query achieves better numbers on the MS-MARCO dev set, it requires training a T5-based query generator with all the available labeled data, while “BM25 + query2doc” does not require any model fine-tuning.

For dense retrieval, the model variants that combine with query2doc also outperform the corresponding baselines on all metrics. However, the gain brought by query2doc tends to diminish when using intermediate pre-training or knowledge distillation from cross-encoder re-rankers, as shown by the “SimLM + query2doc” and “E5 + query2doc” results.

For zero-shot out-of-domain retrieval, the results are mixed as shown in Table 2. Entity-centric datasets like DBpedia see the largest improvements. On the NFCorpus and Scifact datasets, we observe a minor decrease in ranking quality. This is likely due to the distribution mismatch between training and evaluation.

4 Analysis

Scaling up LLMs is Critical For our proposed method, a question that naturally arises is: how does the model scale affect the quality of query expansion? Table 3 shows that the performance steadily improves as we go from the 1.3B model...
Table 2: Zero-shot out-of-domain results on 5 low-resource datasets from the BEIR benchmark (Thakur et al., 2021). The reported numbers are nDCG@10. For a fair comparison, the in-context examples for prompting LLMs come from the MS-MARCO training set.

Table 3: Query expansion with different model sizes. Even though GPT-4 performs best, we are unable to apply it in the main experiments due to quota limits.

to 175B models. Empirically, the texts generated by smaller language models tend to be shorter and contain more factual errors. Also, the “davinci-003” model outperforms its earlier version “davinci-001” by using better training data and improved instruction tuning. The recently released GPT-4 (OpenAI, 2023) achieves the best results.

Table 4: Using the concatenation of the original query and the generated pseudo-documents perform substantially better.

**How to Use Pseudo-documents** In this paper, we concatenate the original query and pseudo-documents as the new query. Alternatively, one can solely use the pseudo-documents, as done in the approach of HyDE (Gao et al., 2022). The results presented in Table 4 demonstrate that the original query and pseudo-documents are complementary, and their combination leads to substantially better performance in sparse retrieval.

**Case Analysis** In Table 5, we show two queries along with their respective pseudo-documents and groundtruth. The pseudo-documents, which are generated by LLMs, offer detailed and mostly accurate information, thereby reducing the lexical mismatch between the query and documents. In some cases, the pseudo-documents are sufficient to meet the user’s information needs, rendering the retrieval step unnecessary. However, it is worth noting that the LLM generations may contain factual errors. For instance, in the second query, the theme song "It’s a Jungle Out There" was used as of season two in 2003, not 2002. Although such errors may appear subtle and difficult to verify, they pose a significant challenge to building trustworthy systems using LLMs.

1Refer to https://en.wikipedia.org/wiki/It’s_a_Jungle_Out_There_(song)
Nicholas II of Russia, the last Tsar of Russia, was executed by Bolshevik revolutionaries on July 17th, 1918, along with his wife Alexandra and their five children. The execution was ordered by Vladimir Lenin, the leader of the Bolshevik Party. The actual execution was carried out by a firing squad of Bolshevik troops, and the bodies were later buried in a mass grave.

(1868-1918). Nicholas II was the last emperor, or tsar, of Russia, serving from 1894 to 1917. Nicholas, his wife, and their five children were killed by the Bolsheviks, members of a revolutionary group (led by Lenin) who seized control of the government in Russia during the October Revolution (see Russian Revolution) of 1917.

The theme song for the television show Monk is entitled "It's a Jungle Out There" and is sung by American singer-songwriter Randy Newman. The song was written specifically for the show, and it has been used as the theme song since the series premiered in 2002. It has been praised by critics and fans alike and is often regarded as one of the best theme songs in television history.
Limitations

<table>
<thead>
<tr>
<th>Method</th>
<th>Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BM25</td>
<td>16</td>
</tr>
<tr>
<td>+ query2doc</td>
<td>&gt;2000</td>
</tr>
</tbody>
</table>

Table 6: Latency analysis for retrieval systems with our proposed query2doc. We retrieve the top 100 results for MS-MARCO dev queries with a single thread and then average over all the queries. The latency for LLM API calls depends on server load and is difficult to precisely measure.

An apparent limitation is the efficiency of retrieval. Our method requires running inference with LLMs which can be considerably slower due to the token-by-token autoregressive decoding. Moreover, with query2doc, searching the inverted index also becomes slower as the number of query terms increases after expansion. This is supported by the benchmarking results in Table 6. Real-world deployment of our method should take these factors into consideration.

References


Rodrigo Nogueira and Jimmy Lin. From doc2query to doctttquery.


Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In 9th International Conference on Learning
A Implementation Details

<table>
<thead>
<tr>
<th>Hyperparameter</th>
<th>DPR</th>
<th>w/ distillation</th>
</tr>
</thead>
<tbody>
<tr>
<td>learning rate</td>
<td>$2 \times 10^{-5}$</td>
<td>$3 \times 10^{-5}$</td>
</tr>
<tr>
<td>PLM</td>
<td>BERT\textsubscript{base}</td>
<td>SimLM / E5\textsubscript{base-unsup}</td>
</tr>
<tr>
<td># of GPUs</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>warmup steps</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>batch size</td>
<td>64</td>
<td>64</td>
</tr>
<tr>
<td>epoch</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>n.a.</td>
<td>0.2</td>
</tr>
<tr>
<td>negatives depth</td>
<td>1000</td>
<td>200</td>
</tr>
<tr>
<td>query length</td>
<td>144</td>
<td>144</td>
</tr>
<tr>
<td>passage length</td>
<td>144</td>
<td>144</td>
</tr>
<tr>
<td># of negatives</td>
<td>15</td>
<td>23</td>
</tr>
</tbody>
</table>

Table 7: Hyper-parameters for training dense retrievers on MS-MARCO passage ranking dataset.

For dense retrieval experiments in Table 1, we list the hyperparameters in Table 7. When training dense retrievers with distillation from cross-encoder, we use the same teacher score released by Wang et al.. The SimLM and E5 checkpoints for initialization are publicly available at https://huggingface.co/intfloat/simlm-base-msmarco and https://huggingface.co/intfloat/e5-base-unsupervised. To compute the text embeddings, we utilize the [CLS] vector for SimLM and mean pooling for E5. This makes sure that the pooling mechanisms remain consistent between intermediate pre-training and fine-tuning. The training and evaluation of a dense retriever take less than 10 hours to finish.

When prompting LLMs, we include 4 in-context examples from the MS-MARCO training set. To increase prompt diversity, we randomly select 4 examples for each API call. A complete prompt is shown in Table 11. On the budget side, we make about 550k API calls to OpenAI’s service, which costs nearly 5k dollars. Most API calls are used to generate pseudo-documents for the training queries.

For GPT-4 prompting, we find that it has a tendency to ask for clarification instead of directly generating the pseudo-documents. To mitigate this issue, we set the system message to “You are asked to write a passage that answers the given query. Do not ask the user for further clarification.”.

Regarding out-of-domain evaluations on DBpedia (Hasibi et al., 2017), NFCorpus (Boteva et al., 2016), Scifact (Wadden et al., 2020), Trec-Covid (Voorhees et al., 2021), and Touche2020 (Bondarenko et al., 2022), SimLM’s results are based on the released checkpoint by Wang et al..

For ablation experiments in Figure 2, we fine-tune for 40 epochs or 18k steps, whichever is reached first.

B Exploring Other Prompting Strategies

<table>
<thead>
<tr>
<th>Model</th>
<th>DL 2019</th>
<th>DL 2020</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query2doc</td>
<td>69.2</td>
<td>64.5</td>
</tr>
<tr>
<td>+ iterative prompt</td>
<td>68.6</td>
<td>64.8</td>
</tr>
</tbody>
</table>

Table 8: GPT-4 with iterative prompting to self-verify the generated pseudo-documents.

Instead of generating pseudo-documents in one round, recent work (Press et al., 2022) proposes to iteratively prompt the LLM to improve the generation quality. We explore this intuition by asking GPT-4 to rewrite its own generated pseudo-documents with the following prompt template:

You are asked to rewrite the passage that potentially answers the given query. You should only correct the factual errors in the passage, do not ask for clarification or make unnecessary changes.

Query: {{query}}

# Begin of passage
{{passage}}

# End of passage

Empirically, we find that GPT-4 makes very few changes to the generated pseudo-documents, which suggests that the pseudo-documents are already of high quality or GPT-4 is not capable of correcting its own errors. The results are shown in Table 8.
Hey Good Lookin’ was originally written and recorded by the country music singer Hank Williams in 1951. The song has since been covered by several other artists and bands, including Jimmy Buffett, Conway Twitty, and Willie Nelson.

Hey, Good Lookin’. (1951) Hey, Good Lookin’ is a 1951 song written and recorded by Hank Williams, and his version was inducted into the Grammy Hall of Fame in 2001. Since its original 1951 recording it has been covered by a variety of artists.

The fax number for the Trumbull Marriott Shelton is 203-378-4444.

HRA stands for Health Reimbursement Arrangement. It is a type of benefit plan where an employer reimburses employees for qualifying medical expenses. It also allows employers to set aside money tax-free to reimburse employees for medical expenses that are not covered under the employer’s health insurance plan. HSA stands for Health Savings Account. It is a savings account that is used in conjunction with a qualified High Deductible Health Plan (HDHP). An HSA enables an individual to pay for current health expenses and save for future health expenses on a tax-free basis. Contributions to the account are tax.

Table 9: More examples of LLM generations. The format is the same as in Table 5.

<table>
<thead>
<tr>
<th>query</th>
<th>who sings hey good looking</th>
</tr>
</thead>
<tbody>
<tr>
<td>LLM generation</td>
<td>Hey Good Lookin’ was originally written and recorded by the country music singer Hank Williams in 1951. The song has since been covered by several other artists and bands, including Jimmy Buffett, Conway Twitty, and Willie Nelson.</td>
</tr>
<tr>
<td>Groundtruth</td>
<td>Hey, Good Lookin’. (1951) Hey, Good Lookin’ is a 1951 song written and recorded by Hank Williams, and his version was inducted into the Grammy Hall of Fame in 2001. Since its original 1951 recording it has been covered by a variety of artists.</td>
</tr>
<tr>
<td>query</td>
<td>trumbull marriott fax number</td>
</tr>
<tr>
<td>LLM generation</td>
<td>The fax number for the Trumbull Marriott Shelton is 203-378-4444.</td>
</tr>
<tr>
<td>Groundtruth</td>
<td>Business name: Trumbull Marriott Merritt Parkway; Address: 180 Hawley Lane Trumbull, Connecticut 06611; Phone number: 203-378-4958; Fax number: 203-378-1400; Business hours: 24; Credit cards accepted: Yes; Number of employees: 10-19; Map:</td>
</tr>
<tr>
<td>query</td>
<td>what is hra and hsa</td>
</tr>
<tr>
<td>LLM generation</td>
<td>HRA stands for Health Reimbursement Arrangement. It is a type of benefit plan where an employer reimburses employees for qualifying medical expenses. It also allows employers to set aside money tax-free to reimburse employees for medical expenses that are not covered under the employer’s health insurance plan. HSA stands for Health Savings Account. It is a savings account that is used in conjunction with a qualified High Deductible Health Plan (HDHP). An HSA enables an individual to pay for current health expenses and save for future health expenses on a tax-free basis. Contributions to the account are tax.</td>
</tr>
<tr>
<td>Groundtruth</td>
<td>The key difference between HRA and HSA is that HRA is an employer funded health benefit plan that reimburses for medical expenses including personal health insurance policy premiums of employees whereas HSA is also a tax-advantaged health benefit plan exclusively available to taxpayers in the United States who are enrolled in a High-Deductible Health Plan (HDHP).</td>
</tr>
</tbody>
</table>

Table 10: Sparse retrieval results of query2doc across 3 random runs. The randomness comes from the selection of few-shot examples and the auto-regressive sampling of LLMs.

<table>
<thead>
<tr>
<th></th>
<th>DL 2019</th>
<th>DL 2020</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>64.8</td>
<td>60.9</td>
</tr>
<tr>
<td>Std dev.</td>
<td>±1.14</td>
<td>±1.63</td>
</tr>
</tbody>
</table>

C Results Across Multiple Runs

In our method, there are two sources of randomness: the selection of few-shot examples and the auto-regressive top-p sampling of LLMs. To quantify the variance of our method, we report the average and standard deviation of sparse retrieval results across 3 random runs in Table 10. One possible improvement is to select few-shot examples based on semantic similarity to the query. We leave this for future work.
Write a passage that answers the given query:

**Query:** what state is this zip code 85282  
**Passage:** Welcome to TEMPE, AZ 85282. 85282 is a rural zip code in Tempe, Arizona. The population is primarily white, and mostly single. At $200,200 the average home value here is a bit higher than average for the Phoenix-Mesa-Scottsdale metro area, so this probably isn’t the place to look for housing bargains. 5282 Zip code is located in the Mountain time zone at 33 degrees latitude (Fun Fact: this is the same latitude as Damascus, Syria!) and -112 degrees longitude.

**Query:** why is gibbs model of reflection good  
**Passage:** In this reflection, I am going to use Gibbs (1988) Reflective Cycle. This model is a recognised framework for my reflection. Gibbs (1988) consists of six stages to complete one cycle which is able to improve my nursing practice continuously and learning from the experience for better practice in the future. In conclusion of my reflective assignment, I mention the model that I chose, Gibbs (1988) Reflective Cycle as my framework of my reflective. I state the reasons why I am choosing the model as well as some discussion on the important of doing reflection in nursing practice.

**Query:** what does a thousand pardons means  
**Passage:** Oh, that’s all right, that’s all right, give us a rest; never mind about the direction, hang the direction - I beg pardon, I beg a thousand pardons, I am not well to-day; pay no attention when I soliloquize, it is an old habit, an old, bad habit, and hard to get rid of when one’s digestion is all disordered with eating food that was raised forever and ever before he was born; good land! a man can’t keep his functions regular on spring chickens thirteen hundred years old.

**Query:** what is a macro warning  
**Passage:** Macro virus warning appears when no macros exist in the file in Word. When you open a Microsoft Word 2002 document or template, you may receive the following macro virus warning, even though the document or template does not contain macros: C:\<path>\<file name>contains macros. Macros may contain viruses.

**Query:** when was pokemon green released  
**Passage:** Pokemon Green was released in Japan on February 27th, 1996. It was the first in the Pokemon series of games and served as the basis for Pokemon Red and Blue, which were released in the US in 1998. The original Pokemon Green remains a beloved classic among fans of the series.

Table 11: The full prompt used for the example in Figure 1.