GOLFer: Generated Documents Hallucination Filter & Combiner for Smaller LLMs-driven Query Expansion in Information Retrieval

Anonymous ACL submission

Abstract

Large Language Models (LLMs)-driven query expansion for information retrieval augments queries with generated hypothetical documents using LLMs. However, the performance of this method heavily depends on the scale of the LLMs, necessitating the use of larger, more advanced LLMs. It can be costly, computationally intensive, and often have limited accessibility. These constraints hinder the practical application of such methods. To address these limitations, we introduce **GOLFer** - Generated Documents Hallucination Filter & Combiner - a novel method designed to leverage smaller open-source LLMs for query expansion in information retrieval. GOLFer effectively detects and filters out non-factual and inconsistent sentences in generated hypothetical documents, a common issue with smaller LLMs, and combines the filtered content with the original query by a weight vector to enhance information retrieval. We evaluate GOLFer alongside dominant LLMs-driven query expansion methods on TREC DL and five low-resource datasets. Experimental results demonstrate that GOLFer consistently outperforms other LLMs-driven query expansion methods based on hypothetical documents generated by a smaller LLM (i.e., LLaMa3-8B) across all datasets, and also maintains competitive performance even against other methods using a advanced LLM (i.e., GPT-3.5), thereby proving the efficacy of our approach.

1 Introduction

Information retrieval (IR) has become essential in modern search engines, efficiently retrieving relevant information from large repositories (Wang et al., 2019; Karpukhin et al., 2020). Users submit textual queries that are matched and ranked against an indexed database to find the most pertinent results. The field has developed two main paradigms: lexical-based sparse retrieval, such as BM25 (Robertson et al., 2009), and embedding-based dense retrieval, such as Contriever (Izacard et al., 2021). Sparse retrieval methods account for term frequency and document length variations, while dense retrieval uses neural networks to capture contextual cues and semantic nuances. Despite dense retrievers' superior performance with large labeled datasets, BM25 remains competitive on low-resource datasets (Thakur et al., 2021). Some studies have highlighted the role of query ambiguity in shaping outcomes (Zhu et al., 2023).

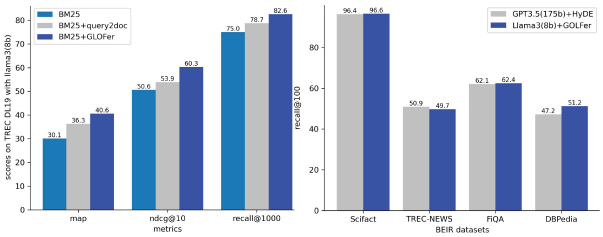

Figure 1: Experimental results comparing GOLFer, other query expansion methods, and a baseline method using LLaMa3-8B for TREC DL19 (left). And Experimental results of GOLFer with LLaMa3-8B and the other query expansion method utilizing GPT-3.5 for low-resource tasks. Detailed results are presented in Tabs. 1 to 5.

To address query ambiguity, query expansion enhances the precision and expressiveness of user queries (Azad and Deepak, 2019). Traditionally, it involves refining the original query using hand-built knowledge resources like WordNet and Thesaurus (Pal et al., 2014; Gong et al., 2005), or leveraging external text collections (Roy et al., 2016; Diaz et al., 2016). The goal is to extract terms that augment the user's query, bridging the gap between the query and relevant documents. However, the effectiveness of these methods depends on the quality of the external data sources and has shown limited success on popular datasets (Azad and Deepak, 2019). Hand-built resources improve retrieval for unstructured queries but offer marginal benefits for structured ones (Anand and Kotov, 2015; Guisado-

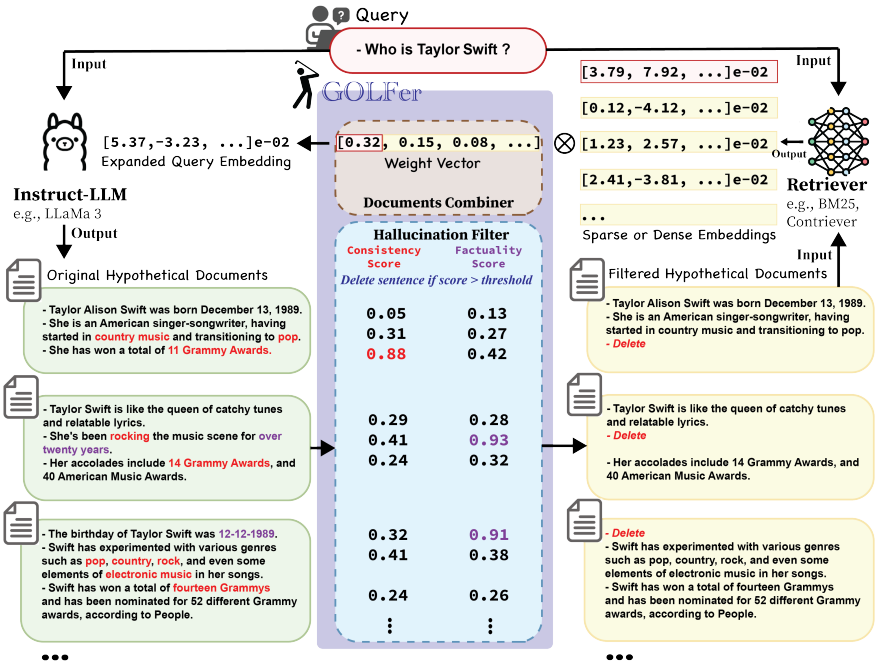

Figure 2: An illustration of our GOLFer framework.

Gámez et al., 2016), while external collections often require preprocessing, which can be problematic without sufficient query logs (Hua et al., 2013). Thus, there is a need for more adaptable query expansion approaches to meet diverse requirements across different contexts.

LLMs like GPT-4 (Ouyang et al., 2022) and LLaMA 3 (Dubey et al., 2024) have shown impressive abilities in generating fluent and realistic responses. Pre-trained on extensive corpora, these models excel in natural language understanding and generation. Ouyang et al. (2022) indicates that LLMs can be fine-tuned with minimal data to align with human intent, enabling them to generalize to diverse instructions in a zero-shot manner. This adaptability has spurred interest in using LLMs for query expansion in IR, where queries are often brief or ambiguous (Mitra and Craswell, 2017; Zhao et al., 2024). LLMs can generate hypothetical documents based on various prompts to enhance query expansion. For example, HyDE uses zero-shot instructions to generate a hypothetical document (Gao et al., 2022), and Query2Doc employs few-shot examples to create hypothetical-documents (Wang et al., 2023). These methods outperform state-of-the-art unsupervised retrievers like Contriever and BM25 and show competitive results against fine-tuned retrievers across tasks like web search, question answering, and fact verification.

However, existing LLMs-driven query expansion methods face several critical challenges. Empirical experiments, such as those involving HyDE, indicate that the performance of LLM-driven query expansion heavily depends on the scale of the LLM employed (Gao et al., 2022). Wang et al. (2023) suggest that smaller LLMs tend to produce shorter outputs with more factual errors, posing a significant obstacle to building trustworthy systems. This tendency to hallucinate facts has been widely observed, where models can confidently generate fictitious information. Consequently, current LLM-driven query expansion methods often advocate for using large-scale, state-of-the-art LLMs like GPT-3.5 (175B) and GPT-4 to mitigate these inaccuracies and enhance query expansion performance in IR. However, in practice, employing such large-scale models is costly, computationally intensive, and regionally restricted. For instance, the API costs for GPT-3.5 and GPT-4 are $1.50 and $2.50 per million tokens, respectively [1], making widespread use financially prohibitive for small and private institutions. Additionally, quota limits often restrict the practical application of these models, and their token-by-token autoregressive decoding can significantly reduce retrieval efficiency, potentially taking over 2000 ms to generate hypothetical documents (Wang et al., 2023). Furthermore, ac-

[1] Refer to https://openai.com/api/pricing/.

cess to large-scale LLMs like GPT is restricted in certain regions, such as China, further limiting their applicability. These factors collectively constrain the practical deployment of existing LLM-driven query expansion methods.

To overcome these limitations, we introduce a new method, GOLFer, i.e., generated documents hallucination filter & combiner for smaller LLMs-driven query expansion in IR (See Fig. 2). Unlike existing LLMs-driven query expansion methods, our method focuses on smaller open-source LLMs, such as LLaMA-3-8B-Instruct, to assist in query expansion, thereby mitigating some of the limitations associated with using large-scale models. However, smaller LLMs often introduce hallucinations, such as non-factual content and inconsistent contexts, when generating hypothetical documents. If these flawed documents are directly utilized by retrievers, they can introduce irrelevant, noisy, or erroneous data, jeopardizing the quality of the expanded query embeddings and thus affecting query expansion performance in IR. To address this issue, GOLFer is designed to detect and filter out non-factual and inconsistent sentences from the hypothetical documents, ensuring that only relevant and accurate information is incorporated. Additionally, to further enhance the impact of the hypothetical documents, our method combines these filtered documents with the original query based on factuality by setting a weight vector. The inner product of the weight vector and the embedding vectors for the original query and filtered documents forms a new, refined query for IR. GOLFer is a lightweight, versatile query expansion method that can be integrated into any Transformer-based LLM without requiring additional training, fine-tuning, or prompt engineering.

We evaluate GOLFer using LLaMA-3-8B-Instruct as the smaller LLM, alongside mainstream dense and sparse retrievers such as BM25, Contriever (Izacard et al., 2021), and ANCE (Xiong et al., 2020). Under the same experimental conditions, we compare GOLFer with existing LLM-driven query expansion methods, including HyDE (Gao et al., 2022) and query2doc (Wang et al., 2023), across two TREC DL datasets and five low-resource datasets from the BEIR benchmark (Thakur et al., 2021). These datasets cover a diverse range of tasks, including web search, question answering, and fact verification. Experimental results demonstrate that GOLFer outperforms existing LLMs-driven query expansion methods with smaller LLMs generated documents on all datasets across almost all evaluation metrics, highlighting the effectiveness of our method. Notably, GOLFer remains competitive even when compared to dominant LLMs-driven query expansion methods which utilize large-scale advanced LLMs like GPT-3.5(175B). Fig. 1 presents some representative results, underscoring the improvement of query expansion performance by GOLFer.

In summary, the contributions of our paper are as follows:

- We propose a novel smaller LLMs-driven query expansion method: GOLFer. It can i) detect and filter out non-factual and inconsistent sentences in the original hypothetical documents generated by smaller LLMs, and ii) combine these filtered hypothetical documents with the original query by a weight vector to form a expanded query for IR, enhancing factuality.

- We evaluate GOLFer alongside dominant query expansion methods on TREC DL19 and DL20 and five low-resource datasets form BEIR. Experimental results demonstrate the efficacy of our method.

2 Related Work

2.1 LLMs-Driven Query Expansion

Query expansion improves retrieval systems by broadening query terms to include synonyms or related concepts, enhancing document matching. Leveraging LLMs' generative capabilities, some studies generate hypothetical documents for query expansion. For instance, HyDE uses an instruction-following LLM to create a hypothetical document (Gao et al., 2022), which is then encoded into an embedding vector for retrieval. Similarly, Query2Doc generates hypothetical documents through few-shot prompting of LLMs (Wang et al., 2023), combining them with the original query to boost retrieval performance. Additionally, various prompting strategies to generate hypothetical documents for query expansion, including zero-shot, few-shot, and Chain-of-Thought, have been explored (Jagerman et al., 2023), with Chain-of-Thought particularly effective. These advances suggest that knowledge distillation from LLMs can transfer their capabilities to smaller models, advancing LLM-driven query expansion.

2.2 Hallucination Detection in LLM-Generated Documents

LLM-generated documents often suffer from hallucinations, producing nonsensical or inaccurate text (Raunak et al., 2021), which degrades system performance (Welleck et al., 2019). Recent research has focused on identifying these hallucinations through three primary approaches: white-box, grey-box, and black-box methods. White-box methods leverage LLM internal states to assess response factuality (Azaria and Mitchell, 2023), requiring labeled data for supervised training. Grey-box methods evaluate factuality using output distributions, employing intrinsic uncertainty metrics to identify uncertain segments (Yuan et al., 2021; Fu et al., 2023). Black-box methods, like SelfCheck-GPT (Manakul et al., 2023), fact-check responses by comparing multiple sampled outputs for consistency. Each approach addresses hallucinations in different contexts, enhancing the reliability of LLM-generated content.

3 Methodology

In this section, we provide a detailed explanation of the GOLFer method. GOLFer is composed of two main components: the hallucination filter and the documents combiner, as shown in Fig. 2. Within the GOLFer method, n passages generated by an Instruct-smaller LLM from a given user query are treated as original hypothetical documents for query expansion. Initially, these documents are processed through the hallucination filter to produce filtered hypothetical documents. Subsequently, the expanded query embedding for query expansion is obtained by combining embedding vectors with a weight vector, which is determined by the documents combiner module in the GOLFer method. The configuration of the weight vector varies between sparse and dense retrieval methods. The hallucination filter is detailed in section 3.1, while the documents combiner is discussed in section 3.2.

Notation Suppose n passages for query expansion are generated by an instruction-following smaller LM, such as LLaMa-3-8B-Instruct, given a user query q. Let d^i refer to the i-th smaller LM-generated document, where $i \in \{1, 2, ..., n\}$. Each document d^i contains m_i sentences, denoted by s^i_j for the j-th sentence in document d^i, where $j \in \{1, 2, ..., m_i\}$. Furthermore, each sentence s^i_j consists of o^i_j tokens, with $t^i_j(l)$ representing the l-th token in the j-th sentence of the i-th smaller LM-generated document, where $l \in \{1, 2, ..., o^i_j\}$.

3.1 Hallucination Filter

The hallucination filter module can evaluate the degree of hallucination for each sentence in smaller LM-generated passages based on consistency and factuality, and then, can filter these sentences out based on their hallucination degree. In this module, we introduce two scores—consistency score, $\mathscr{C}$, and factuality score, $\mathscr{F}$—to comprehensively assess the hallucination degree. The following subsections will detail the methods used to calculate these consistency and factuality scores for each sentence in LLM-generated passages.

Consistency score The fundamental idea behind detecting hallucination in terms of consistency is rooted in the premise that if a smaller LLM possesses genuine knowledge of a concept, its sampled responses will likely be similar and factually consistent. Conversely, hallucinated facts often lead to divergent and contradictory responses when multiple outputs are drawn from the same query. By comparing multiple responses generated from the same query, we can assess information consistency and determine the factuality of statements (Manakul et al., 2023). Natural Language Inference (NLI) has been utilized to measure faithfulness in hallucination detection (Manakul et al., 2023), demonstrating competitive performance. Inspired by this approach, we employ a fine-tuned NLI classifier to compute the NLI contradiction score for each sentence across different documents. Specifically, we utilize DeBERTa-v3-large (He et al., 2021) fine-tuned on the MNLI dataset Williams et al. (2018) as our NLI classifier. Only the logits associated with the entailment and contradiction classes are considered, and the NLI contradiction score $\mathscr{C}$ for the sentence s^i_j in document $d^{k \neq i}$ is computed as follows:

$$\mathscr{C}(s^i_j, d^{k \neq i}) = \frac{\exp(\omega_c)}{\exp(\omega_c) + \exp(\omega_e)}, \quad (1)$$

where $\exp(\omega_c)$ and $\exp(\omega_e)$ are the logits of the contradiction and entailment classes, respectively. In GOLFer, the consistency score for a sentence s^i_j is the mean value of its NLI contradiction scores across all documents that do not contain s^i_j. This can be formulated as:

$$\mathscr{C}(s^i_j) = \frac{\sum_{k \in \{1,2,...,n\} \setminus \{i\}} \mathscr{C}(s^i_j, d^k)}{n-1}. \quad (2)$$

Factuality score Factuality score is based on the idea that factual sentences generated by a smaller LLM tend to contain tokens with higher likelihood and lower entropy, whereas hallucinated sentences are characterized by tokens with flat probability distributions and high uncertainty. What's more, each token has a different influence on the subsequent context. Thus, we define the factuality score of the hypothesis documents by evaluates the uncertainty of tokens and their impact on subsequent tokens. This dual consideration is inspired by Su et al. (2024). Our method begins by quantifying the uncertainty of each token, $t_j^i(l)$. This is achieved by recording the entropy of the token's probability distribution across the vocabulary. For any token $t_j^i(l)$, the entropy $\mathcal{H}_{j_l}^i$ is computed as follows:

$$\mathcal{H}_{j_l}^i = -\sum_{\tilde{v} \in \mathcal{V}} p_{j_l}^i(\tilde{v}) \log p_{j_l}^i(\tilde{v}), \tag{3}$$

where $p_{j_l}^i(\tilde{v})$ denotes the probability of generating the token $\tilde{v}$ over all tokens in the vocabulary $\mathcal{V}$ at position l of the j-th sentence in document i.

In addition to uncertainty, GOLFer leverages the self-attention mechanism inherent in Transformer-based LLMs to assign weights to tokens, reflecting their impact on the subsequent context. Specifically, for any given token $t_j^i(l)$, we quantify its influence by recording the average attention value $Avg(\mathcal{A}_{j_l}^i)$, which captures the average attention from all following tokens. The attention scores are taken from the last Transformer layer of the smaller LM. The attention value $\mathcal{A}_{j_{l,v}}^i$ between two tokens $t_j^i(l)$ and $t_j^i(v)$ for any $l < v$ is computed as follows:

$$\mathcal{A}_{j_{l,v}}^i = \text{softmax}\left(\frac{Q_{j_l}^i {K_{j_v}^i}^\top}{\sqrt{d_k}}\right), \tag{4}$$

where $Q_{j_l}^i$ represents the query vector of token $t_j^i(l)$, $K_{j_v}^i$ is the key vector of token $t_j^i(v)$, and d_k denotes the dimensionality of the key vector. The softmax function is applied to the dot product of $Q_{j_l}^i$ and $K_{j_v}^i$, normalized by the square root of d_k. The average attention value $Avg(\mathcal{A}_{j_l}^i)$ for token $t_j^i(l)$ is then identified by averaging $\mathcal{A}_{j_{l,v}}^i$ for all $v > l$:

$$Avg(\mathcal{A}_{j_l}^i) = \frac{\sum_{v=l+1}^{o_j^i} \mathcal{A}_{j_{l,v}}^i}{o_j^i - l}. \tag{5}$$

Combining uncertainty and significance, GOLFer computes a comprehensive factuality score for each token $t_j^i(l)$. Specifically, the factuality score $\mathscr{F}_{j_l}^i$ is calculated by multiplying the entropy $\mathcal{H}_{j_l}^i$ of the token by its average attention value $Avg(\mathcal{A}_{j_l}^i)$:

$$\mathscr{F}_{j_l}^i = \mathcal{H}_{j_l}^i \cdot Avg(\mathcal{A}_{j_l}^i). \tag{6}$$

This token-level factuality score serves as the basis for evaluating the overall factuality of a sentence. The sentence-level factuality score $\mathscr{F}(s_j^i)$ is then derived by averaging the factuality scores of all tokens within the sentence:

$$\mathscr{F}(s_j^i) = \frac{\sum_{l=1}^{o_j^i} \mathscr{F}_{j_l}^i}{o_j^i}. \tag{7}$$

We then set a set of threshold value (θ, δ) to filter out sentences in the documents. Only the sentences satisfied that $(\mathscr{C}(\cdot), \mathscr{F}(\cdot)) < (\theta, \delta)$ are qualified for query expansion. We have empirically found that $(\theta = 0.85, \delta = 0.85)$ is a generally good value and do not tune it on a dataset basis.

3.2 Documents Combiner

The documents combiner module can merge the original query with filtered hypothetical documents to form an expanded query for IR. This combination process is tailored based on the type of retrieval—sparse or dense—and the generation confidence of the documents. The subsequent sections elaborate on the specific operations for sparse and dense retrievers, respectively.

Sparse Retrieval In the case of sparse retrieval, we enhance the query term weights by repeating the query $k \cdot \alpha \cdot n$ times. This repetition aims to balance the relative weights of the query and the hypothetical documents before merging them. Here, k compensates for the typically shorter length of the query q compared to the hypothetical documents, with $k = 5$ being an empirically effective value (Wang et al., 2023). The parameter α is positively correlated with the number of deleted sentences, and we find $\alpha = 0.75$ to be generally effective for $(\theta, \delta) = (0.85, 0.85)$ without requiring dataset-specific tuning. The term n represents the number of hypothetical documents. The expanded query q^+ for sparse retrieval is then formulated as follows:

$$q^+ = \text{int}(k \cdot \alpha \cdot n) \cdot q + \sum_{i=1}^{n} d^i. \tag{8}$$

This formulation ensures an effective balance between the original query and the augmented content for improved retrieval performance.

Dense Retrieval For dense retrieval, the document combiner module takes into account the generation confidence of smaller LM-generated hypothetical documents. These documents, with varying degrees of generation confidence, exhibit different levels of factual information and relevance patterns concerning the real documents we aim to retrieve. Consequently, we posit that hypothetical documents with higher generation confidence should contribute more significantly to the query expansion process for IR.

Specifically, we estimate the generation confidence, $\varpi(d^i)$ of a filtered document by averaging the generation probabilities, p^i_{jl}, of each token within the filtered documents. This can be expressed as follows:

$$\varpi(d^i) = \frac{\sum_{j=1}^{m_i} \sum_{l=1}^{o^i_j} p^i_{jl}}{\sum_{j=1}^{m_i} o^i_j}. \quad (9)$$

Subsequently, we encode both the generated documents and the original query into embedding vectors using the dense retriever, denoted as $f(\cdot)$. The expanded query embedding for IR, V_{q^+}, is then formulated by combining the embedding vectors for the filtered hypothetical documents and the original query with their corresponding weights. This can be expressed as follows:

$$V_{q^+} = \beta \cdot f(q) + \frac{1-\beta}{\sum_{i=1}^{n} \varpi(d^i)} \cdot \sum_{i=1}^{n} \varpi(d^i) f(d^i), \quad (10)$$

where β represents the contribution rate of the original query in forming the expanded query for IR. It varies according to the max tokens adopted in documents generation:

$$\beta = a_1 + b_1 * \mu, \quad (11)$$

where μ refers to the max tokens for a hypothesis documents. And we find that $a_1 = 0.25$ and $b_1 = 0.075$ is a set of effective values, which we do not tune on a dataset-specific basis. For dense retrieval, the inner product is computed between V_{q+} and the set of all document vectors, and the most similar documents are subsequently retrieved.

4 Experiments

4.1 Setup

Implementation We implement Meta-LLaMA-3-8B-Instruct (Dubey et al., 2024), a smaller open-source LLM, to generate hypothetical documents for given queries. We sample documents with a temperature setting of 0.6, top-p of 0.9 and max tokens of 128 for open-ended generation. Retrieval experiments are conducted using the Pyserini toolkit (Lin et al., 2021), and performance is evaluated using metrics MAP, nDCG@10 and R@k ($k \in \{100, 1000\}$).

Datasets We evaluate our method using TREC DL19 (Craswell et al., 2020) and DL20 (Craswell et al., 2021), both derived from the MS-MARCO dataset (Bajaj et al., 2016). Additionally, we utilize a diverse collection of five low-resource datasets from the BEIR benchmark (Thakur et al., 2021), including Scifact, TREC-NEWS, FiQA, DBPedia, and Touche2020.

Compared Systems For our experiments, we set the classical heuristic-based lexical retriever BM25 (Robertson et al., 2009) as our baseline for sparse retrieval. For dense retrieval, we use the Contriever (Izacard et al., 2021) as our baseline without fine-tuning. It is trained using unsupervised contrastive learning. And ANCE (Xiong et al., 2020), which involves fine-tuning on massive relevance data and involves some sophisticated hard negative techniques serve as our dense retrieval baseline with fine-tuning. Retrievers within the GOLFer share the same embedding spaces as these baselines; the primary difference lies in how the query vector is constructed. This setup allows us to effectively assess the impact of GOLFer. Additionally, we compare our method with two other LLMs-driven query expansion approaches: HyDE (Gao et al., 2022), designed for dense retrieval systems, and query2doc (Wang et al., 2023), applicable to both sparse and dense retrieval systems. In this section, HyDE (Gao et al., 2022) and query2doc (Wang et al., 2023) use the same original hypothesis documents as we do.

4.2 Web Search

The results, summarized in Tab. 1, show the performances of different query expansion methods based on hypothesis documents generated by LLaMA-3-8B-Instruct. For sparse retrieval, GOLFer outperforms another query expansion approach, query2doc, across all metrics, highlighting its superior effectiveness in enhancing retrieval performance. In the context of dense retrieval, ANCE combined with GOLFer outperforms other LLMs-driven query expansion approaches, except for a marginal underperformance in one metric, where

	DL19			DL20		
	MAP	nDCG@10	Recall@1000	MAP	nDCG@10	Recall@1000
w/o relevance judgement						
BM25	30.1	50.6	75.0	28.6	48.0	78.6
+query2doc	36.3	53.9	78.7	38.4	56.1	83.6
+GOLFer	**40.6**$^{+10.4}$	**60.3**$^{+9.7}$	**82.6**$^{+7.6}$	**42.3**$^{+13.8}$	**59.7**$^{+11.7}$	**85.6**$^{+6.9}$
w/ relevance judgement						
ANCE	37.1	64.5	75.5	40.8	64.6	77.6
+HyDE	45.3	68.2	80.5	44.2	67.8	81.6
+query2doc	45.0	69.9	77.8	44.6	67.5	**82.0**
+GOLFer	**46.0**$^{+8.9}$	**71.4**$^{+6.9}$	**80.6**$^{+5.0}$	**45.7**$^{+4.9}$	**69.2**$^{+4.6}$	81.2$^{+3.6}$

Table 1: Results for web search on DL19/20. Best performing systems are marked **bold**. ANCE is one of the in-domain supervised models that are fine-tuned on MS MARCO training data.

	Scifact	TREC-NEWS	FiQA	DBPedia	Touche2020
			nDCG@10		
w/o relevance judgement					
BM25	67.9	39.5	23.6	31.8	44.2
+query2doc	70.3	49.1	23.6	32.8	45.3
+GOLFer	**70.8**$^{+2.9}$	**50.5**$^{+11.0}$	**23.9**$^{+0.3}$	**34.1**$^{+2.3}$	**45.8**$^{+1.6}$
w/ relevance judgement					
Contriever	64.9	34.8	24.5	29.2	16.7
+HyDE	67.9	40.1	24.5	30.5	17.5
+query2doc	66.2	38.7	21.3	31.1	18.3
+GOLFer	**69.4**$^{+4.5}$	**41.1**$^{+6.3}$	**28.1**$^{+3.6}$	**35.7**$^{+6.5}$	**21.1**$^{+4.5}$
			Recall@100		
w/o relevance judgement					
BM25	92.5	44.7	54.0	46.8	**58.2**
+query2doc	95.1	51.5	56.0	44.3	56.8
+GOLFer	**95.4**$^{+2.9}$	**52.8**$^{+8.1}$	**56.9**$^{+3.0}$	**47.0**$^{+0.2}$	56.7
w/ relevance judgement					
Contriever	92.6	42.3	56.2	45.3	37.4
+HyDE	96.1	46.1	59.3	44.1	39.3
+query2doc	94.1	43.2	56.8	42.9	38.7
+GOLFer	**96.6**$^{+4.0}$	**49.7**$^{+7.4}$	**62.4**$^{+6.2}$	**51.2**$^{+5.9}$	**41.8**$^{+4.4}$

Table 2: Results for Low resource tasks from BEIR. Best performing systems are marked **bold.**

the difference is only 0.8. This consistent improvement across both sparse and dense retrieval models underscores the robustness and reliability of GOLFer in enhancing smaller LLMs-driven query expansion.

4.3 Low Resource Retrieval

The performance for low resource tasks are summarized in Tab. 2. For sparse retrieval using BM25 as the retriever, GOLFer generally shows better performance than query2doc across almost all low-resource datasets. The only exception is a slight underperformance in Recall@100 on the Touche2020 dataset, where the margin is minimal (0.1 difference). This consistent performance underscores the robustness of GOLFer in enhancing sparse retrieval tasks. For dense retrieval using Contriever as the retriever, GOLFer also surpasses other query expansion approaches, across all low-resource datasets and metrics.

These results demonstrate the effectiveness of GOLFer in filtering and combining smaller LLMs-generated texts contributes significantly to the improvement of query expansion performance.

5 Analysis

In this section, we explore the potential of GOLFer using a smaller LLM by comparing it with existing dominant LLMs-driven query expansion methods with advanced LLMs. Previous studies have shown that the scale of the generative LLM significantly impacts the quality of query expansion. For instance, Wang et al. (2023) demonstrated that query2doc with GPT-3.5 (175B) achieves over 11% improvements in nDCG@10 on TREC DL19 compared to query2doc with Curie (6.7B). Similarly,

	GPT-3.5 (175B)	LLaMA-3 (8B)
	w/ BM25	
	w/ query2doc(few-shot)	w/ GOLFer
Scifact	68.6	**70.8**
DBPedia	**37**	34.1
touche2020	39.8	**45.8**

Table 3: Results for effect of different combination of instruction LLMs and query expansion approaches for sparse retrieval. nDCG@10 on BEIR. Best systems are marked **bold.**

	GPT-3.5 (175B)	LLaMA-3 (8B)
	w/ Contriever	
	w/ HyDE	w/ GOLFer
Scifact	69.1	**69.4**
TREC-NEWS	**44**	41.1
FiQA	27.3	**28.1**
DBPedia	**36.8**	35.7

Table 4: Results for effect of different combination of instruction LLMs and query expansion approaches for dense retrieval. nDCG@10 on BEIR. Best systems are marked **bold.**

Gao et al. (2022) found that HyDE with GPT-3.5 (175B) outperforms Flan-T5 (11B) by 12.4% in nDCG@10 on the same dataset with Contriever. To assess GOLFer's potential, we compared our performance to Wang et al. (2023) and Gao et al. (2022) in BEIR datasets. It is important to note that the hypothesis documents for GOLFer are generated using LLaMA-3-8B-Instruct, while those for other methods are generated by GPT-3.5 (text-davinci-003, (Wang et al., 2022)), which comprises 175B parameters. Theoretically, larger LLMs generate higher-quality hypothetical documents with fewer factual errors, which is crucial for query expansion.

	GPT-3.5 (175B)	LLaMA-3 (8B)
	w/ Contriever	
	w/ HyDE	w/ GOLFer
Scifact	96.4	**96.6**
TREC-NEWS	**50.9**	49.7
FiQA	62.1	**62.4**
DBPedia	47.2	**51.2**

Table 5: Results for effect of different combination of instruction LLMs and query expansion approaches for dense retrieval. Recall@100 on BEIR. Best systems are marked **bold.**

Compare to Large size Generative Models In Tab 3, we compare the performance of GOLFer using a smaller LLM with the query2doc method employing an advanced LLM on sparse retrieval in BEIR datasets. The results indicate that GOLFer with LLaMA-3-8B surpasses query2doc with GPT-3.5 on the SciFact and Touche2020 datasets, showing a notable 15% improvement on the latter. However, on the DBPedia dataset, query2doc with GPT-3.5 outperforms GOLFer with LLaMA-3-8B. It is worth noting that while query2doc generates hypothesis documents using few-shot prompting, our approach employs zero-shot prompting.

In Tabs. 4 and 5, we compare the performance of dense retrieval between GOLFer with LLaMA-3-8B and HyDE with GPT-3.5 on low-resource datasets from BEIR. GOLFer with LLaMA-3-8B consistently outperforms HyDE with GPT-3.5 on the SciFact and FiQA datasets across all metrics. Additionally, GOLFer surpasses HyDE in terms of Recall@100 on the DBPedia dataset, although it falls slightly behind in nDCG@10. Conversely, HyDE with GPT-3.5 shows superior performance on the TREC-NEWS dataset across all metrics.

In summary, we found that GOLFer with LLaMA-3-8B is competitive with, and sometimes outperforms, other LLMs-driven query expansion methods such as query2doc and HyDE with GPT-3.5 (175B) across various low-resource retrieval tasks from the BEIR benchmark. GOLFer is potential as a viable alternative to advanced LLMs-driven query expansion methods in information retrieval.

6 Conclusion

In this work, we introduce GOLFer, a novel method designed to leverage smaller open-source LLMs for query expansion, aiming to enhance both sparse and dense retrieval systems. The core idea is to distill the smaller LLMs outputs through effective hallucination detection and mitigation techniques. GOLFer identifies and filters out non-factual and inconsistent sentences in smaller LLMs-generated documents, ensuring that only reliable documents are used as hypothetical documents for query expansion. The expanded query embeddings for information retrieval are then obtained by computing the dot product of the embedding vectors of the filtered hypothetical documents and the original query with a weight vector. Experimental evaluations demonstrate that the effectiveness of GOLFer in filtering and combining smaller LLMs-generated texts contributes significantly to the improvement of query expansion performance in information retrieval.

7 Limitations

We acknowledge several limitations in this paper. One significant limitation is the dependency on the self-attention mechanism of Transformer-based LLMs for evaluating factuality scores within the hallucination detection module. Although self-attention scores are available for all open-source LLMs, our method cannot be applied directly to certain APIs that do not offer access to these scores. Consequently, our future work will focus on developing alternative approaches to address this limitation.

References

Rajul Anand and Alexander Kotov. 2015. An empirical comparison of statistical term association graphs with dbpedia and conceptnet for query expansion. In *Proceedings of the 7th Annual Meeting of the Forum for Information Retrieval Evaluation*, pages 27–30.

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

Amos Azaria and Tom Mitchell. 2023. The internal state of an llm knows when it's lying. *arXiv preprint arXiv:2304.13734*.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. 2016. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*.

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

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M Voorhees. 2020. Overview of the trec 2019 deep learning track. *arXiv preprint arXiv:2003.07820*.

Fernando Diaz, Bhaskar Mitra, and Nick Craswell. 2016. Query expansion with locally-trained word embeddings. *arXiv preprint arXiv:1605.07891*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2023. Gptscore: Evaluate as you desire. *arXiv preprint arXiv:2302.04166*.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2022. Precise zero-shot dense retrieval without relevance labels. *arXiv preprint arXiv:2212.10496*.

Zhiguo Gong, Chan Wa Cheang, and U Leong Hou. 2005. Web query expansion by wordnet. In *Database and Expert Systems Applications: 16th International Conference, DEXA 2005, Copenhagen, Denmark, August 22-26, 2005. Proceedings 16*, pages 166–175. Springer.

Joan Guisado-Gámez, Arnau Prat-Pérez, and Josep Lluis Larriba-Pey. 2016. Query expansion via structural motifs in wikipedia graph. *arXiv preprint arXiv:1602.07217*.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*.

Xian-Sheng Hua, Linjun Yang, Jingdong Wang, Jing Wang, Ming Ye, Kuansan Wang, Yong Rui, and Jin Li. 2013. Clickage: Towards bridging semantic and intent gaps via mining click logs of search engines. In *Proceedings of the 21st ACM international conference on Multimedia*, pages 243–252.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Towards unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*, 2(3).

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

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.

Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. Pyserini: A python toolkit for reproducible information retrieval research with sparse and dense representations. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2356–2362.

Potsawee Manakul, Adian Liusie, and Mark JF Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. *arXiv preprint arXiv:2303.08896*.

Bhaskar Mitra and Nick Craswell. 2017. Neural models for information retrieval. *arXiv preprint arXiv:1705.01509*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Dipasree Pal, Mandar Mitra, and Kalyankumar Datta. 2014. Improving query expansion using wordnet. *Journal of the Association for Information Science and Technology*, 65(12):2469–2478.

Vikas Raunak, Arul Menezes, and Marcin Junczys-Dowmunt. 2021. The curious case of hallucinations in neural machine translation. *arXiv preprint arXiv:2104.06683*.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

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

Weihang Su, Yichen Tang, Qingyao Ai, Zhijing Wu, and Yiqun Liu. 2024. Dragin: Dynamic retrieval augmented generation based on the real-time information needs of large language models. *arXiv preprint arXiv:2403.10081*.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *arXiv preprint arXiv:2104.08663*.

Heyuan Wang, Ziyi Wu, and Junyu Chen. 2019. Multi-turn response selection in retrieval-based chatbots with iterated attentive convolution matching network. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1081–1090.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Simlm: Pre-training with representation bottleneck for dense passage retrieval. *arXiv preprint arXiv:2207.02578*.

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

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2019. Neural text generation with unlikelihood training. *arXiv preprint arXiv:1908.04319*.

Adina Williams, Nikita Nangia, and Samuel R Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2018*, pages 1112–1122. Association for Computational Linguistics (ACL).

Yijun Xiao and William Yang Wang. 2021. On hallucination and predictive uncertainty in conditional language generation. *arXiv preprint arXiv:2103.15025*.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *arXiv preprint arXiv:2007.00808*.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. *Advances in Neural Information Processing Systems*, 34:27263–27277.

Wayne Xin Zhao, Jing Liu, Ruiyang Ren, and Ji-Rong Wen. 2024. Dense text retrieval based on pretrained language models: A survey. *ACM Transactions on Information Systems*, 42(4):1–60.

Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Haonan Chen, Zhicheng Dou, and Ji-Rong Wen. 2023. Large language models for information retrieval: A survey. *arXiv preprint arXiv:2308.07107*.