Analyze, Generate and Refine: Query Expansion with LLMs for Zero-Shot Open-Domain QA

Xinran Chen[1,2], **Xuanang Chen**[2]✉, **Ben He**[1,2]✉, **Tengfei Wen**[1], **Le Sun**[2]
[1]School of Computer Science and Technology, University of Chinese Academy of Sciences
[2]Chinese Information Processing Laboratory, Institute of Software, Chinese Academy of Sciences
chenxinran22@mails.ucas.ac.cn,xuanang2020@iscas.ac.cn,benhe@ucas.ac.cn
wentengfei23@mails.ucas.ac.cn,sunle@iscas.ac.cn

Abstract

Query expansion (QE) is a critical component in the open-domain question answering (OpenQA) pipeline, enhancing the retrieval performance by broadening the scope of queries with additional relevant texts. However, existing methods like GAR and EAR rely heavily on supervised training and often struggle to maintain effectiveness across domains and datasets. Meanwhile, although large language models (LLMs) have demonstrated QE capability for information retrieval (IR) tasks, their application in OpenQA is hindered by the inadequate analysis of query's informational needs and the lack of quality control for generated QEs, failing to meet the unique requirements of OpenQA. To bridge this gap, we propose a novel LLM-based QE approach named AGR for the OpenQA task, leveraging a three-step prompting strategy. AGR begins with an analysis of the query, followed by the generation of answer-oriented expansions, and culminates with a refinement process for better query formulation. Extensive experiments on four OpenQA datasets reveal that AGR not only rivals in-domain supervised methods in retrieval accuracy, but also outperforms state-of-the-art baselines in out-of-domain zero-shot scenarios. Moreover, it exhibits enhanced performance in end-to-end QA evaluations, underscoring the superiority of AGR for OpenQA.[1]

1 Introduction

Open-domain question answering (OpenQA) is a key task in Natural Language Processing, aiming to provide accurate answers to a wide range of factual questions across different domains (Chen and Yih, 2020; Kwiatkowski et al., 2019). The challenge in OpenQA is to retrieve relevant information from large text corpora without specific contexts (Zhu et al., 2021). Retrieval methods are therefore essential, with two primary approaches being lexical-based sparse retrieval, like BM25 (Robertson and Zaragoza, 2009), and embedding-based dense retrieval (Karpukhin et al., 2020; Guu et al., 2020). Dense retrieval models (Luan et al., 2021; Xiong et al., 2021; Qu et al., 2021), while effective with ample domain-specific training data, are computationally demanding and risk omitting crucial information due to their reliance on fixed-length embedding that may not capture all the textual nuances, leading to potential exclusion of relevant details. Conversely, sparse retrieval combined with query expansion techniques (Lavrenko and Croft, 2017; Chuang et al., 2023) can address these semantic challenges and achieve competitive performance.

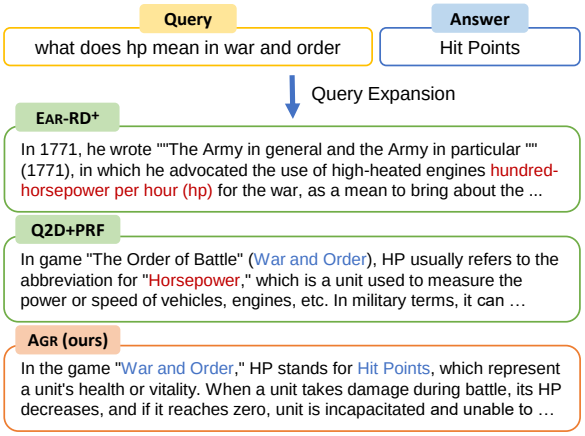

Figure 1: Examples of query expansion generated by EAR-RD$^+$, Q2D+PRF and AGR methods for a query sampled from NQ dataset. EAR-RD$^+$ is a supervised in-domain QE method, it fails to capture the informational needs about "war and order" when transferred from TriviaQA to NQ dataset, resulting in generating erroneous contents, like "hundred-horsepower per hour". Q2D+PRF is a LLM-based QE method, although it knows the game "war and order", its expansion contains irrelevant content about "horsepower" due to the lack of quality control of generation. In contrast, our AGR generates high-quality query expansion contains the correct answer "hit points" for the query.

[1]Our code and data are publicly available at https://github.com/process-cxr/AGR.

Findings of the Association for Computational Linguistics ACL 2024, pages 11908–11922

The recent advent of generation-augmented retrieval has shown promise in providing more precise information for OpenQA, as evidenced by methods such as GAR (Mao et al., 2021) and EAR (Chuang et al., 2023). These approaches utilize seq2seq models like BART (Lewis et al., 2020) to generate contexts that are tailored to the query, incorporating elements such as the answer, the sentence containing the answer, and the title of the passage where the answer is located. Nevertheless, these methods often require extensive supervised training data, which can lead to the generation of subpar QEs, particularly in out-of-domain zero-shot scenarios. For instance, as depicted in Figure 1, the EAR-RD$^+$ model inadequately addresses the information needs about the phrase "war and order," resulting in the generation of incorrect content. Concurrently, the rise of large language models (LLMs) like GPT-3 (Brown et al., 2020) and Flan-T5 (Raffel et al., 2020) has underscored their potential as effective QE tools in information retrieval (IR) tasks (Li et al., 2023), operating without the need for training data or external corpora, such as Q2D (Wang et al., 2023) and Q2D+PRF (Jagerman et al., 2023). However, while these models excel in directly bolstering queries with expansions derived from a LLM, they often lack mechanisms for ensuring the quality of the generated QEs. This issue is exemplified in Figure 1, where the QE produced by Q2D+PRF includes a mix of relevant and irrelevant information, signaling the necessity of effective quality control measures.

To this end, we propose a novel LLM-based QE method specially designed for the OpenQA task, coined as AGR. As shown in Figure 2, AGR employs a three-step progressive prompting strategy, namely **A**nalyze, **G**enerate, and **R**efine, to leverage the extensive capabilities of LLMs, facilitating cross-domain query expansion for OpenQA. Specifically, to address the informational needs of a given query, the Analyze phase utilizes the question understanding capabilities of LLMs to generate an analysis. Subsequently, in Generate phase, AGR utilizes the knowledge retrieval and integration capabilities of LLMs to generate various answer-oriented query expansions as candidates. These expansions are then used to retrieve reference texts from the corpus to filter out erroneous and irrelevant generated information for generating new candidates closer to potential answers. Finally, for the purpose of quality control over QEs, the Refine phase conducts a self-review of all candidates, refining them to achieve a high-quality query expansion. Obviously, AGR is explicitly oriented towards QA tasks with a series of progressively advancing sub-task steps, making it more suitable for high-quality QE generation in out-of-domain zero-shot scenarios.

Extensive experiments are carried out on four widely-used OpenQA datasets, including Natural Questions (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), WebQuestions (Berant et al., 2013), and CuratedTREC (Baudis and Sedivý, 2015). The results of retrieval evaluation demonstrate that AGR not only achieves comparable performance to SOTA supervised QE methods such as EAR (Chuang et al., 2023) on their trained in-domain datasets but also surpasses them in the context of zero-shot scenarios on out-of-domain datasets, which emphasizes the zero-shot capabilities of our AGR method. Furthermore, in comparison to LLM-based QE methods tailored for IR tasks, AGR exhibits the ability to generate more answer-oriented information so that sparse retrievers leveraging AGR can identify more accurate passages containing the answers, which contributes to the improved end-to-end QA quality.

Our contributions are three-fold: 1) We propose a LLM-based QE method AGR, which adopts a novel reasoning chain generation process suited for OpenQA. 2) Extensive experiments show that AGR achieves SOTA out-of-domain zero-shot performance in boosting the retrieval quality across diverse datasets. 3) End-to-end QA evaluation indicates that AGR enhances the exact match scores of answers, emphasizing its practical utility in real-world applications.

2 Related Work

2.1 Query Expansion

Query expansion (QE) has received widespread attention in the early literature of information retrieval (Efthimiadis, 1996), fundamentally boosts retrieval systems by enriching queries with additional, conceptually similar terms (Carpineto and Romano, 2012). Early QE methods in IR mainly augment queries with additional terms based on user relevance feedback (Rocchio Jr, 1971), which is often unavailable. Then, the Pseudo-Relevance Feedback (PRF) mechanisms (Croft et al., 2009) are developed, wherein the top-ranking results of an original query are utilized for expansions, but this is also constrained by the quality of the top

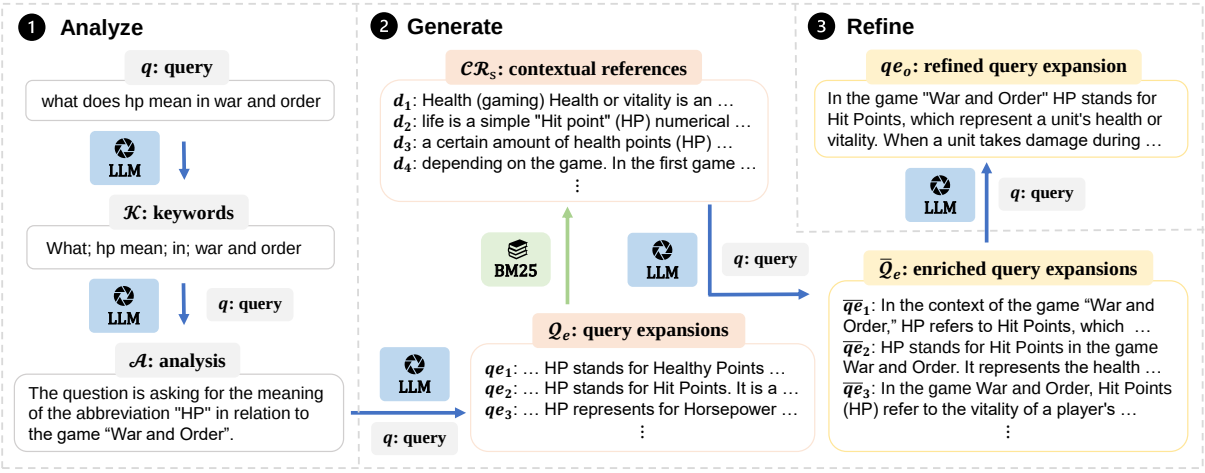

Figure 2: An overview of AGR, where the LLM is prompted to execute a sequence of three sub-steps: Analyze, Generate, and Refine, to produce a refined answer-oriented QE. This approach capitalizes on the concept of multi-step task decomposition, enhancing the effectiveness of QE tailored for OpenQA.

documents (Carpineto and Romano, 2012). More recent QE studies for IR tasks have started to leverage pre-trained language models in the process of query expansion (Zheng et al., 2020; Naseri et al., 2021), commonly achieved through training or fine-tuning models.

The capability of QE in enhancing retrieval has also been utilized in studies on OpenQA, such as GAR(Mao et al., 2021) and EAR(Chuang et al., 2023). GAR utilizes a trained BART model to generate answer-oriented QEs, effectively bridging the information gap between the original query and potential answers. Building upon by GAR, EAR employs the generator from GAR for sampling and generating a various set of QEs, and uses a query ranking model to select the best QE. Although these methods enhance the effectiveness of retrievers, they heavily depend on supervised in-domain training of the generator or reranker, which constrains their adaptability across domains and datasets under zero-shot OpenQA.

2.2 Large Language Models

The rapid advancements in generative modeling have led to the development of large language models (LLMs), like ChatGPT (Ouyang et al., 2022) and GPT-4 (OpenAI, 2023), which demonstrate exceptional multitasking capabilities, particularly exemplifying outstanding zero-shot learning abilities (Liu et al., 2022; Dong et al., 2023). Moreover, LLMs are suitable for a wide array of tasks (Brown et al., 2020; Alayrac et al., 2022), ranging from language translation and question answering to more intricate challenges like sentiment analysis and dialogue generation (Kaddour et al., 2023). Recent studies have highlighted the emergence of prominent open-source LLMs like LLama2 (Touvron et al., 2023) and Mistral (Jiang et al., 2023), which excel in various LLM benchmarks and display robust multitasking capabilities, garnering substantial academic interest due to their accessibility and manageable parameter sizes.

In the domain of query expansion, LLMs have also shown promising capabilities. For instance, WebCPM (Qin et al., 2023) and Q2E (Jagerman et al., 2023) employ LLMs to generate topic-related terms as query expansion, Query2Doc (Wang et al., 2023) focuses on employing LLMs to generate passages related to the potential answers, aiming to alleviate the issue of word mismatch between query and documents. While these LLM-based methods have notably enhanced QE in IR, underscoring the revolutionary impact of LLMs, they are not specifically tailored for OpenQA. As a result, these methods may not fully align with the informational nuances required for queries in OpenQA, and lack quality control for QEs. In contrast, our proposed AGR introduces a multi-step generation framework designed for QA tasks, which is designed to meet the informational needs of query and implements quality control to obtain a refined QE.

3 Methodology

As shown in Figure 2, the proposed AGR leverages the concept of multi-step task decomposition, encompassing a progressive sub-task workflow to analyze, generate, and refine the query. This approach

thoroughly exploits the abilities of LLMs (Zhao et al., 2023), like semantic comprehension, knowledge retrieval, and integration, as well as contextual understanding and reasoning. The result is an expanded query that is anticipated to be more aligned with the answer, thereby improving the performance of sparse retrieval mechanisms within the OpenQA task.

3.1 Overview

Given an original query q, AGR first makes use of an LLM (like Mistral) to analyze this query and generate an analytic text $\mathcal{A}$ on it. Subsequently, building on this analysis, a set of answer-oriented query expansions $\mathcal{Q}_e = \{qe_1, qe_2, ..., qe_n\}$ is generated by the LLM through random sampling strategy, which ensures the diversity of generated expansion content. However, empirical observations indicate that a portion of query expansions align with the correct answer, while others may result from model hallucinations, manifesting as either irrelevant to the correct answer or erroneous. Thus, similar to pseudo relevance feedback (PRF), all top-k documents from BM25 for all query expansions $\mathcal{Q}_e$ can be integrated to rectify and output a series of new query expansion $\overline{\mathcal{Q}}_e = \{\overline{qe}_1, \overline{qe}_2, \ldots, \overline{qe}_m\}$, which combines the knowledge exist in LLM with the knowledge retrieved from the corpus. After that, a final review by the LLM on these all enriched QE candidates $\overline{\mathcal{Q}}_e$ is further conducted to identify relevant answers and erroneous information, ultimately generating the refined query expansion qe_o, which is appended to the original query q to execute the following retrieval.

3.2 Analysis of Original Query

In the initial phase, we first leverage the semantic comprehension abilities of LLM to extract key phrases $\mathcal{K}$ from the original query q, and then conduct an analysis $\mathcal{A}$ based on $\mathcal{K}$.

$$\mathcal{K} = \text{LLM}(q) \tag{1}$$

$$\mathcal{A} = \text{LLM}(q, \mathcal{K}) \tag{2}$$

The generation of key phrases aims to identify the core elements of the original query, which turns back to help the LLM grasp the essential theme and contextual cues within the query, and output a more targeted and relevant analysis. The ablation study conducted on the generation of $\mathcal{K}$ and $\mathcal{A}$ in Section 4.3 highlights the significance of this particular step.

3.3 Generation of Candidate QEs

Building on the insights gained from the initial analysis $\mathcal{A}$ of the query q, the LLM's knowledge retrieval and integration capabilities are employed to produce multiple candidate expansions $\mathcal{Q}_e = \{qe_1, qe_2, \ldots, qe_n\}$.

$$\mathcal{Q}_e = \text{LLM}(q, \mathcal{A}) \tag{3}$$

Subsequently, to enhance the reliability of expansions, we turn to use the relevant documents from the corpus to rectify them. Specifically, the BM25 top-k documents $\mathcal{D}_{qe_i}$ for each candidate expansion $qe_i \in \mathcal{Q}_e$ are all collected to form contextual references $\mathcal{CR}_s$. Then, the texts $\mathcal{CR}_s$ in conjunction with the original query q are re-introduced into the LLM to conduct a second round of sampling generation. This process blends the LLM's intrinsic knowledge with information retrieved from the corpus, yielding more enriched query expansions $\overline{\mathcal{Q}}_e = \{\overline{qe}_1, \overline{qe}_2, ..., \overline{qe}_m\}$.

$$\mathcal{CR}_s = \cup\{\mathcal{D}_{qe_i}\} = \cup\{\text{BM25}(qe_i)\}, qe_i \in \mathcal{Q}_e \tag{4}$$

$$\overline{\mathcal{Q}}_e = \text{LLM}(q, \mathcal{CR}_s) \tag{5}$$

In both the generation steps in Eq. 3 and Eq. 5, a random sampling strategy is adopted, which ensures that LLMs produce a range of potential expansions, each reflecting different facets and interpretations of the initial query q. The breadth of generated expansions provides a comprehensive pool for the following refinement step.

3.4 Refinement for Optimal QE

Due to their stochastic and diverse nature, the documents provided from the corpus to the LLM facilitate the generation of enriched QE candidates, $\overline{\mathcal{Q}}_e$. While many of these candidates are highly relevant or closely aligned with the correct answer for the query q, some may still be irrelevant or erroneous, likely due to the hallucinatory tendencies of LLMs. Consequently, in this phase, we leverage the LLMs' capabilities in contextual understanding and reasoning to conduct a thorough review of all QE candidates:

$$qe_o = \text{LLM}(q, \overline{\mathcal{Q}}_e) \tag{6}$$

This process involves discerning between valuable and superfluous information. In other words, we refine and distill these expansions into an optimized query expansion qe_o, which is then appended to the original query q, enhancing the quality of sparse retrieval outcomes.

Throughout these stages, AGR capitalizes on the LLMs' analytical and generative capabilities, offering a novel paradigm for answer-oriented QE, enhancing the overall performance in OpenQA scenarios. Overall, AGR is a three-step prompting method based on LLMs, and note that the whole process of AGR does not require additional training or fine-tuning of any models, it only relies on the inherent capabilities of LLMs to generate enhanced QEs in a zero-shot context.

4 Experiments

4.1 Experimental Setup

Datasets For the evaluation datasets, we select four diverse datasets pertinent to OpenQA task, including Natural Questions (NQ) (Kwiatkowski et al., 2019), TriviaQA (Trivia) (Joshi et al., 2017), WebQuestions (WebQ) (Berant et al., 2013) and CuratedTREC (TREC) (Baudis and Sedivý, 2015). The former two datasets are utilized for baselines involving supervised approaches and the latter are employed to assess all baseline methods under an unsupervised zero-shot evaluation setting. Detailed introduction of datasets refers to Appendix A. Given that datasets relevant to OpenQA frequently encompass questions annotated with gold-standard answers as benchmarks for evaluation (Rajpurkar et al., 2016), we have consequently established the task paradigm as Retriever-Reader.

Details of AGR We choose Mistral-7B (Jiang et al., 2023) as the backbone model of AGR, and more analysis on different models, including LLama-3B, LLama2-7B and LLama2-13B (Touvron et al., 2023), are available in Appendix B.1. Besides, we utilize the `SamplingParams` class from vLLM (Kwon et al., 2023), allowing for precise and optimized settings across different phases. Specifically, `temperature` is configured at 0.2 for the Analyze and Refine phases, and adjusted to 0.8 for the Generate phase. Considering the constraint of the LLMs' token input capacity limitation, in the Generate phase, the number of generated query expansion n is fixed at 15, and the number of top-ranking documents retrieved for contextual reference k is set at 3. The impacts of these two hyper-parameters are detailed in Section 4.3. In the Refine phase, the number of generated query expansion m is determined to be 10, aiming for a balance between accuracy and practicality. As for inference resources, our experiment employs NVIDIA L40 48GB GPUs. Other detailed settings are provided in the Appendix B.2.

Baselines We compare AGR with four types of approaches:

1) *Direct retrieval without QE*: BM25 (Robertson and Zaragoza, 2009) is a standard term-matching sparse retriever, and DPR (Karpukhin et al., 2020) is a standard dense retriever based on BERT model.

2) *Traditional QE method*: BM25+RM3 (Roy et al., 2019) uses a traditional QE method based on extracting tokens from retrieval documents without supervised training or LLMs.

3) *Supervised QE models*: GAR (Mao et al., 2021) adopts three types of query expansion generators based on trained seq2seq models, EAR further uses trained query rankers to reorganize the QEs, achieving SOTA performance on NQ and TriviaQA datasets as shown in Chuang et al. (2023).

4) *LLM-based QE for IR tasks*: Query2doc (Q2D) (Wang et al., 2023) uses LLMs to generate answer-oriented passages as QEs, and Q2E (query2keywords) (Jagerman et al., 2023) generates topic-oriented keyword information as QEs.

To ensure a unified basis for comparison, all QE methods employ BM25 for retrieval, and all LLM-based QE baseline methods utilize Mistral-7B as the backbone model. Moreover, for the specific experimental settings during the generation and PRF retrieval processes of the baseline Q2D+PRF and Q2E+PRF, they utilize the same hyperparameters as AGR, involving sampling generation of 10 QEs, retrieving 3 PRF documents for each QE, and a second generation for 10 new QEs based on the retrieved documents. The distinction arises with that AGR implements a third-step refinement for the newly generated QE candidates, whereas Q2D+PRF and Q2E+PRF randomly choose one QE candidate from the candidates.

Metrics Akin to prior research (Brown et al., 2020; Mao et al., 2021), we adopt the Retriever-Reader task paradigm and introduce two principal metrics. *Hit@k* for retrieval accuracy, is utilized for assessing the effectiveness of the retriever, and it is defined as the proportion of questions/queries for which at least one answer span is contained within the top-*k* retrieved passages. *EM@k* for exact match score, is employed to evaluate the results of the Reader, serving as an end-to-end performance indicator. The score is the proportion

Method	Natural Questions		TriviaQA		WebQuestions		CuratedTREC	
	Hit@5	Hit@100	Hit@5	Hit@100	Hit@5	Hit@100	Hit@5	Hit@100
In-domain Supervised Settings								
DPR*	68.3	86.1	72.7	84.8	62.8	82.2	66.6	89.9
GAR*	60.8	84.7	71.8	85.3	-	-	-	-
EAR-RI*	63.2	85.9	73.4	85.9	-	-	-	-
EAR-RD*	**69.3**	**86.5**	**77.6**	**86.4**	-	-	-	-
Out-of-domain Zero-shot Settings								
BM25	43.8	78.3	67.7	83.9	41.8	75.5	64.3	89.9
BM25+RM3	43.23	75.46	64.07	82.16	-	-	-	-
DPR+	-	-	-	-	52.7/56.8	78.3/81.2	74.1/78.8	92.1/93.7
GAR+	-/40.00	-/75.01	61.54/-	81.17/-	50.0/45.5	79.0/76.7	70.9/71.5	92.4/91.5
EAR-RI+	-/45.87	-/79.09	67.80/-	83.99/-	53.7/49.6	81.3/79.6	73.5/74.2	92.9/92.5
EAR-RD+	-/ 50.58	-/78.92	70.77/-	84.05/-	59.5/54.5	81.3/79.7	80.0/79.8	93.7/93.1
Q2D	59.71	84.32	72.89	85.04	64.22	83.26	84.26	94.38
Q2D+PRF	61.41	82.99	73.69	84.98	63.43	82.18	84.58	93.80
Q2E	51.72	79.63	68.46	82.81	57.08	78.54	76.65	91.78
Q2E+PRF	50.72	76.90	67.06	80.95	55.26	76.03	74.92	90.34
AGR (ours)	**68.47**	**85.76**	**77.47**	**86.01**	**67.07**	**83.51**	**88.62**	**94.96**

Table 1: Hit@k retrieval accuracy (%) on test sets across four datasets. Methods marked with * denote supervised in-domain settings, while those with a plus + indicate cross-domain settings (transferred from NQ/TriviaQA).

of instances where the predicted answer span exactly matches one of the true answers after string normalization.

4.2 Results

Retrieval evaluations As presented in Table 1, our principal findings from the retrieval evaluations can be summarized as follows:

1) Our zero-shot AGR is on par with the strong supervised baseline EAR-RD*. This equivalence is demonstrated in Table 1. The AGR method, closely matching the performance of SOTA supervised method EAR-RD* within a 1% margin, showcases its effectiveness. Particularly noteworthy is AGR's improved performance over the supervised approach DPR* on TriviaQA, highlighting its capability to understand search intent without relying on supervised training.

2) AGR showcases superior generalization abilities over all baseline methods. As substantiated in Table 1, on the former two datasets our experiment conducts a cross-domain evaluation of GAR and EAR methods, and for the subsequent two datasets we incorporate cross-domain results of DPR, GAR, and EAR from (Chuang et al., 2023) (transferred from NQ/TriviaQA). These results are considered representative of zero-shot out-of-domain settings for supervised methods. Evidently, AGR demonstrates consistently superior performance over all baseline methods in zero-shot out-of-domain settings across various datasets, manifesting its generalization ability beyond domain constraints.

3) AGR outperforms other LLM-based QE methods in IR tasks due to its QA-specific design. This conclusion is drawn from the comparison to LLM-based QE methods, where AGR consistently leads in performance against counterparts like Query2Doc, Q2E, and their variants. Surpassing the average Hit@5/100 retrieval accuracy of these methods by significant margins, AGR confirms its design specifically tailored for QA tasks, as opposed to general IR objectives.

End-to-end QA evaluations To further compare AGR method with all baseline approaches across the complete end-to-end OpenQA task, we uniformly employ the Fusion-in-Decoder (FiD) model (Izacard and Grave, 2021) pre-trained on NQ/TriviaQA datasets as the reader component. As illustrated in Table 2, we can obtain observations from the results as follows:

1) AGR secures a dominant position in end-to-end EM scores across various datasets. This view is supported by the end-to-end evaluation results conducted on the NQ/TriviaQA datasets. The results indicate that our AGR method almost consistently outperforms all other baseline approaches in terms of end-to-end effectiveness. This includes

Method	Natural Questions			TriviaQA		
	EM@1	EM@10	EM@100	EM@1	EM@10	EM@100
GAR*	28.14	42.00	49.61	40.38	62.51	69.8
EAR-RI*	28.84	44.54	51.25	48.26	65.61	71.16
EAR-RD*	36.20	46.73	**51.94**	57.14	67.59	71.46
BM25+RM3	13.55	31.72	44.53	40.36	58.64	67.55
GAR$^+$	13.13	29.78	42.47	38.70	55.78	65.83
EAR-RI$^+$	14.99	32.58	45.70	41.96	61.96	69.46
EAR-RD$^+$	20.42	47.01	45.71	48.84	63.53	69.63
Q2D	28.14	42.05	50.33	55.56	66.59	71.40
Q2D+PRF	31.52	42.16	49.61	58.07	66.07	70.49
Q2E	21.99	36.26	46.73	46.95	61.91	68.15
Q2E+PRF	23.21	34.46	44.40	48.82	60.02	66.29
AGR (ours)	**37.73**	**47.01**	51.50	**61.64**	**69.29**	**72.24**

Table 2: Exact-match scores for end-to-end QA on NQ and TriviaQA (TQ) test sets. EM@1/10/100 denote the scores when top-1/10/100 documents are input to FiD.

surpassing the SOTA in-domain supervised method EAR-RD*, which is notable given that its retrieval accuracy marginally exceeds AGR. This comparison underscores AGR's comprehensive proficiency, especially in demonstrating a performance advantage in end-to-end OpenQA task.

2) With fewer input documents, AGR's end-to-end EM score advantage grows more distinct. This conclusion is derived through conducting multiple comparative tests by varying the input parameters of the FiD reader. In these experiments, we used top-1/10/100 retrieved passages – standard quantity parameters for FiD – as inputs to compute the EM scores for each QE method accordingly. Significantly, with the reduction in the number of input passages from 100 to 1, the AGR method's superiority becomes progressively evident. This suggests that the passages retrieved through AGR not only demonstrate exceptional retrieval accuracy but also rank higher in delivering gold passages rich with answer content, thereby facilitating FiD in extracting more accurate responses.

4.3 Analysis

Ablation study To better comprehend the utility of AGR, we conduct various experiments on the NQ dataset analyzing the impact and effectiveness of each component within this architecture as follow:

1) Necessity of individual components: In this experiment, we establish three variants to investigate the necessity of each component: **a) w/o Analyze:** Our proposed framework without the analysis step, replaced by the generate-refine pipeline; **b) w/o CRs:** Our proposed framework without integrating contextual references (CRs) in second round sampling generation; **c) w/o Refine:** Our proposed framework without the refinement step, directly samples a QE from candidates.

Method	Hit@5	EM@10
AGR	**68.47**	**47.01**
w/o Analyze	67.85	46.56
w/o CRs	64.07	44.72
w/o Refine	67.64	46.42

Table 3: Ablation results of AGR on NQ datasets.

From Table 3, we can draw the following conclusions: a) The performance of AGR on the NQ datasets is better than these variants lacking components of this method, affirming the effectiveness of our proposed Analyze, Generate, and Refine framework. A plausible explanation involves decomposing the problem into sub-questions and employing multi-step progressive prompting. This can amplify LLMs' multi-task reasoning capabilities, thereby facilitating the generation of improved answer-oriented QEs specific to OpenQA. b) When comparing the performance among different variants, we observe that the variant w/o CRs performs the worst, highlighting the critical role of contextual references in enhancing the quality of subsequent sampling generation. By further incorporating the Refine step, it can reduce irrelevant or erroneous information from the initial round of sampling generation, thus enhancing the overall performance of the AGR framework.

2) Quantitative assessment of generation and refinement: At this assessment stage, we statisti-

		Query: what is the main mineral in lithium batteries? **Answer**: lithium, Lithium
EAR-RD*	QE	The mineral lithium chloride is the most abundant component of lithium ion; as such , it is often considered ...
	Retriever	Title: Lithium as an investment Text: Increased tendency for costlier components to be targeted for replacement by new technologies. Current projections of the global market for lithium-iron batteries range from $26 billion in 2023 (Navigant Research). It's most frequently found in deposits such as spodumene and pegmatite...
	Reader	Answer: spodumene
AGR	QE	The main minerals used in the production of lithium for lithium batteries are spodumene, petalite, lepidolite ...
	Retriever	Title: Lithium iron phosphate Text: Lithium iron phosphate, also known as LFP, is an inorganic compound with the formula. It is a gray, red-grey, brown, or black solid that is insoluble in water. The material has attracted attention as a candidate component of lithium iron phosphate batteries. It's targeted for use in power...
	Reader	Answer: Lithium

Table 4: An example of end-to-end QA via EAR-RD* and AGR on NQ dataset. Although both top-1 retrieved are gold passages, the answer obtained by EAR-RD* is wrong due to the quality of the gold passage.

cally analyze the QEs generated across three distinct stages, to quantitatively assess the impact of integrating contextual references in the sampling generation process between $\mathcal{Q}_e$ and $\overline{\mathcal{Q}}_e$, as well as the effect of the refinement process between $\overline{\mathcal{Q}}_e$ and qe_o. A QE is considered high-quality if the retrieval results include the gold passage within the top-5 documents; otherwise, it is categorized as low-quality. The proportion of high-quality QEs per query instance then acts as our fundamental unit of statistical measurement.

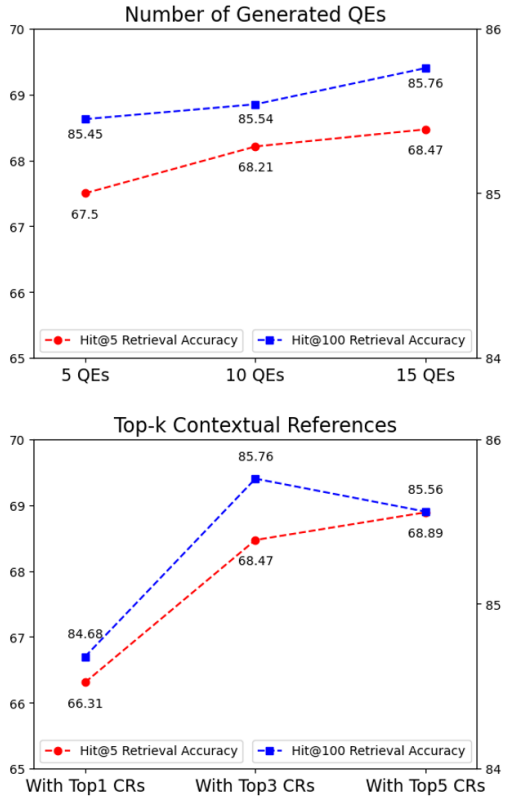

Figure 3: The impacts of hyper-parameter on NQ dataset. In experiments concerning *"n"*, *"k"* is fixed to 3, and for the assessment of *"k"*, *"n"* is fixed to 15.

Specific statistical results are depicted in Figure 4. We observe an increasing trend where most intermediate data shifted towards higher proportion rates of high-quality QEs after undergoing the generation and refinement processes. This trend suggests that the processes are effective at filtering out some irrelevant or erroneous information produced during the initial sampling generation, thereby improving the quality of the QE. Conversely, less intermediate data moved towards lower proportion rates, indicating that an overabundance of irrelevant or erroneous information can negatively impact subsequent generation attempts.

To further quantify the process, we track the change in the proportion of high-quality QEs across three stages for each query, estimating the probability of transformation: With a threshold set at 0.6, data above this threshold had a 95.1% probability of resulting in high-quality QEs after processing, whereas data below had only a 31.03% probability. These findings highlight the limitations of the effectiveness in the Generate and Refine phase, indicating the need for quality control at each stage to obtain the final high-quality QE.

Hyper-parameter sensitivity We proceed to explore the sensitivity of our method's performance to hyper-parameters, primarily focusing on two pivotal sets in the Generate phase: a) the number of generated query expansion, denoted as *"n"*, and b) the number of top-ranking documents retrieved from contextual references, labeled as *"k"*. Due to constraint of the LLMs' token input capacity limitation, the range for both is capped at their maximum values, with *"n"* up to 15 and *"k"* up to 5.

The experiments presented in Figure 3 demonstrate that within the constraints of the model's token input capacity, increasing *"n"* enhances re-

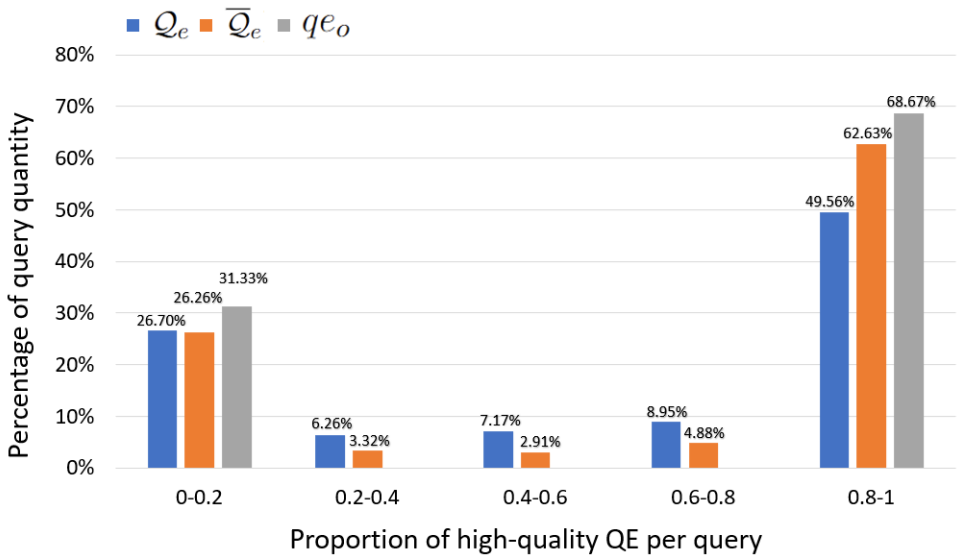

Figure 4: Trend analysis of the ratio of high-quality QEs during the generation and refinement of AGR.

trieval accuracy. This suggests that employing a more powerful model with a larger token input capacity could yield even better results. On the other hand, marginal gains diminish as k increases, suggesting a point of diminishing returns in terms of retrieval accuracy enhancement. These findings clearly illustrate the AGR method's performance sensitivity to these two hyper-parameters.

Case illustration As demonstrated by the results in Tables 1 & 2, although AGR marginally lags behind SOTA in-domain supervised method EAR-RD* in terms of retrieval accuracy, it exhibits better end-to-end EM scores, nearly universally surpassing EAR-RD* across various setting. This phenomenon underscores the adaptability of AGR's design for end-to-end OpenQA. As illustrated by an example in Table 4, it can be observed that the top-1 documents retrieved by both methods contain the question's answer, marked as the gold passage, while the gold passage retrieved by AGR exhibits a stronger relevance to the answer and subject matter compared to EAR-RD*, allowing the reader to have a better chance to extract the correct answer.

Time overhead Regarding time overhead, since both employ sample generation, Q2D+PRF and Q2E+PRF involve 2 generative inference steps throughout the process, while AGR involves 5, with the retrieval overhead being identical for both. Consequently, AGR exhibits a time overhead in generative inference which is 2 to 3 times higher than that of Q2D+PRF. In the future we plan to emloy more feasible multi-GPU distributed model loading strategies for the deployment of larger-parameters LLMs under limited computational resources, and then we can merge some LLMs generation steps (combining Equations 1 and 2), without compromising effectiveness, to reduce the number of inferences and lower latency. Additionally, we will also utilize more powerful parallel inference strategies to increase the speed of LLM inference within the pipeline, thereby reducing latency.

5 Conclusion

In this paper, we introduce a multi-step generative framework AGR with analyze, expand and refine phases, which deeply mines the inherent knowledge of LLMs for answer-oriented query expansion. AGR eschews the need for additional in-domain data for supervised training and demonstrates significantly higher retrieval accuracy compared to state-of-the-art QE methods under out-of-domain zero-shot scenes. Moreover, when integrated with the Fusion-in-Decoder as the reader component, AGR achieves a nearly comprehensive lead in end-to-end performance, showcasing its effectiveness on OpenQA task. In future work, we plan to explore a more comprehensive framework, further integrating the requirements of the reader component to achieve a unified architecture and enhance the end-to-end performance for OpenQA task.

Limitations

Firstly, as shown in our experimental analysis, the effectiveness of our proposed AGR, an LLMs-based QE generation framework, is intrinsically linked to the quality of the underlying backbone model. It is needful to find out a suitable LLM to fully demonstrate the AGR's capacity, and carry out further investigations with more types of LLMs, like GPT-4. Secondly, although AGR avoids additional supervised training data and conserves computational resources, its reliance on multiple inference generations with LLMs introduces certain latency. Therefore, further exploration of strategies to accelerate LLM inference and reduce overall latency are left for future work. Finally, the generations from LLMs are associated with the input prompts. Although we have tested multiple configurations to achieve the current performance, the possibility of better configurations exists.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (62272439), and the Fundamental Research Funds for the Central Universities.

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L. Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karén Simonyan. 2022. Flamingo: a visual language model for few-shot learning. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Petr Baudis and Jan Sedivý. 2015. Modeling of the question answering task in the yodaqa system. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction - 6th International Conference of the CLEF Association, CLEF 2015, Toulouse, France, September 8-11, 2015, Proceedings*, volume 9283 of *Lecture Notes in Computer Science*, pages 222–228. Springer.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1533–1544. ACL.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Claudio Carpineto and Giovanni Romano. 2012. A survey of automatic query expansion in information retrieval. *ACM Comput. Surv.*, 44(1):1:1–1:50.

Danqi Chen and Wen-tau Yih. 2020. Open-domain question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 34–37, Online. Association for Computational Linguistics.

Yung-Sung Chuang, Wei Fang, Shang-Wen Li, Wen-tau Yih, and James Glass. 2023. Expand, rerank, and retrieve: Query reranking for open-domain question answering. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12131–12147, Toronto, Canada. Association for Computational Linguistics.

W. Bruce Croft, Donald Metzler, and Trevor Strohman. 2009. *Search Engines - Information Retrieval in Practice*. Pearson Education.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, Lei Li, and Zhifang Sui. 2023. A survey for in-context learning. *CoRR*, abs/2301.00234.

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

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Retrieval augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 3929–3938. PMLR.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume,*

EACL 2021, Online, April 19 - 23, 2021, pages 874–880. Association for Computational Linguistics.

Rolf Jagerman, Honglei Zhuang, Zhen Qin, Xuanhui Wang, and Michael Bendersky. 2023. Query expansion by prompting large language models. *CoRR*, abs/2305.03653.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *CoRR*, abs/2310.06825.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Jean Kaddour, Joshua Harris, Maximilian Mozes, Herbie Bradley, Roberta Raileanu, and Robert McHardy. 2023. Challenges and applications of large language models. *CoRR*, abs/2307.10169.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick S. H. Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6769–6781. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur P. Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: a benchmark for question answering research. *Trans. Assoc. Comput. Linguistics*, 7:452–466.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles, SOSP 2023, Koblenz, Germany, October 23-26, 2023*, pages 611–626. ACM.

Victor Lavrenko and W. Bruce Croft. 2017. Relevance-based language models. *SIGIR Forum*, 51(2):260–267.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Minghan Li, Honglei Zhuang, Kai Hui, Zhen Qin, Jimmy Lin, Rolf Jagerman, Xuanhui Wang, and Michael Bendersky. 2023. Generate, filter, and fuse: Query expansion via multi-step keyword generation for zero-shot neural rankers. *CoRR*, abs/2311.09175.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics.

Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. 2021. Sparse, dense, and attentional representations for text retrieval. *Trans. Assoc. Comput. Linguistics*, 9:329–345.

Yuning Mao, Pengcheng He, Xiaodong Liu, Yelong Shen, Jianfeng Gao, Jiawei Han, and Weizhu Chen. 2021. Generation-augmented retrieval for open-domain question answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4089–4100, Online. Association for Computational Linguistics.

Shahrzad Naseri, Jeff Dalton, Andrew Yates, and James Allan. 2021. CEQE: contextualized embeddings for query expansion. In *Advances in Information Retrieval - 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28 - April 1, 2021, Proceedings, Part I*, volume 12656 of *Lecture Notes in Computer Science*, pages 467–482. Springer.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Yujia Qin, Zihan Cai, Dian Jin, Lan Yan, Shihao Liang, Kunlun Zhu, Yankai Lin, Xu Han, Ning Ding, Huadong Wang, Ruobing Xie, Fanchao Qi, Zhiyuan Liu, Maosong Sun, and Jie Zhou. 2023. WebCPM: Interactive web search for Chinese long-form question answering. In *Proceedings of the 61st Annual*

Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 8968–8988, Toronto, Canada. Association for Computational Linguistics.

Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. 2021. Rocketqa: An optimized training approach to dense passage retrieval for open-domain question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 5835–5847. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Stephen E. Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: BM25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

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

Dwaipayan Roy, Sumit Bhatia, and Mandar Mitra. 2019. Selecting discriminative terms for relevance model. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1253–1256.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Liang Wang, Nan Yang, and Furu Wei. 2023. Query2doc: Query expansion with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9414–9423, Singapore. Association for Computational Linguistics.

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 Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A survey of large language models. *CoRR*, abs/2303.18223.

Zhi Zheng, Kai Hui, Ben He, Xianpei Han, Le Sun, and Andrew Yates. 2020. BERT-QE: contextualized query expansion for document re-ranking. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 4718–4728. Association for Computational Linguistics.

Fengbin Zhu, Wenqiang Lei, Chao Wang, Jianming Zheng, Soujanya Poria, and Tat-Seng Chua. 2021. Retrieving and reading: A comprehensive survey on open-domain question answering. *CoRR*, abs/2101.00774.

A Dataset Information

Natural Questions (NQ) (Kwiatkowski et al., 2019) introduced as a question answering dataset, is comprised of real, anonymized, aggregated queries submitted to the Google search engine. The dataset includes 79,168 training examples, 8,757 development examples, and 3,610 test examples.

TriviaQA (Trivia) (Joshi et al., 2017) is a substantial and realistic text-based question answering dataset, which encompasses 950,000 question-answer pairs derived from 662,000 documents sourced from Wikipedia and the web. The dataset is composed of 60,413 training examples, 8,377 development examples, and 11,313 test examples, providing a diverse range of scenarios that test the depth and adaptability of QA systems.

WebQuestions (WebQ) (Berant et al., 2013) designed for question answering tasks, utilizes Freebase as its underlying knowledge base and consists of 6,642 question-answer pairs. This dataset was developed by sourcing questions through the Google Suggest API, followed by obtaining corresponding answers via Amazon Mechanical Turk. It is structured with an original split of 3,778 training examples and 2,032 testing examples. All answers are defined as Freebase entities.

CuratedTREC (TREC) (Baudis and Sedivý, 2015) is a reference question dataset for benchmarking Question Answering systems created from the TREC-8 (1999) to TREC-13 (2004). It comprises a concise yet focused collection of 694 annotated data entries, making it an ideal resource for evaluating the precision and effectiveness of QA systems under test conditions.

B Experimental Details

In this appendix section, we elaborate on the more details of the model parameter settings employed during the inference process of the AGR method, along with the precise prompts used in all LLM-based QE methods throughout our experiments.

B.1 Backbone Model

We conducted a preliminary experiment to identify the most appropriate model foundation, evaluating the performance of prominent LLMs including Llama-3B, Llama2-7B, Llama2-13B, and Mistral-7B. The Base method was designed for direct answer-oriented text generation, whereas the AGR method was simplified, excluding contextual

Method	Prompt
Q2E	"""Write a list of keywords for the given query: Query: {query} Keywords:"""
Q2E+PRF	"""Write a list of keywords for the given query based on the context: Context: {Q2K_PRF_Doc_1} {Q2K_PRF_Doc_2} {Q2K_PRF_Doc_3} Query: {query} Keywords:"""
Q2D	"""Write a passage that answers the given query: Query: {query} Passage:"""
Q2D+PRF	"""Write a passage that answers the given query based on the context: Context: {Q2D_PRF_Doc_1} {Q2D_PRF_Doc_2} {Q2D_PRF_Doc_3} Query: {query} Passage:"""

Table 5: Prompts for Q2D, Q2E and their variants.

references integration, to expedite evaluation. Notably, prompt adaptability varied across models to ensure compatibility, maintaining a consistent overall framework.

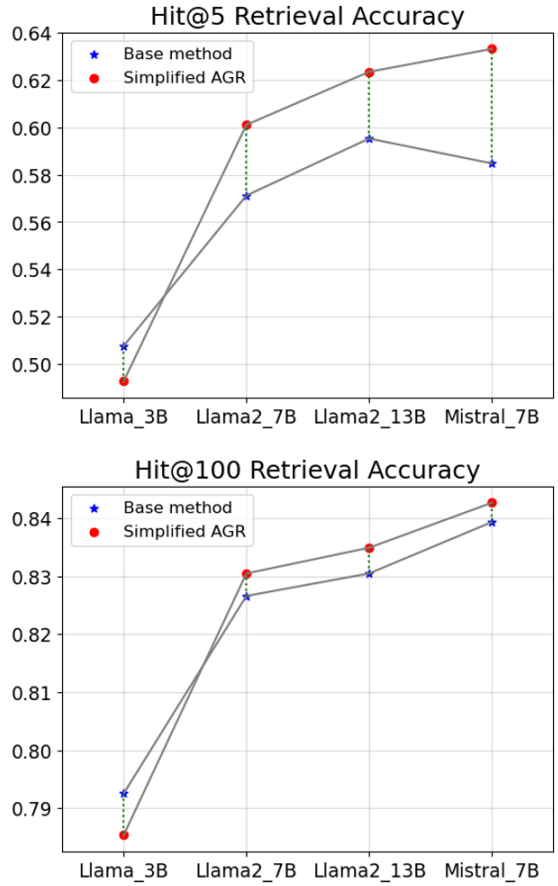

Figure 5: Accuracy of backbone models on NQ.

Experimental results, depicted in Figure 5 show that the effectiveness of the AGR method is intrinsically linked to the size and capabilities of underlying model. Under the Llama-3B setup, AGR underperformed due to the model's limited grasp of complex prompts. While it consistently achieves improvements on larger models, with enhancements scaling alongside model capabilities. This highlights that models with greater robustness and advanced semantic understanding better complement the AGR approach. Specifically, the Mistral-7B model outshines Llama2-7B and Llama2-13B in reasoning tasks, in line with the findings of Jiang et al. (2023) research.

B.2 Additional Parameter Details

For the detailed configuration of model parameters within the AGR framework, we utilize the SamplingParams class from vLLM (Kwon et al., 2023) to facilitate precise and optimized settings across different stages. Key parameters include *"temperature"* set at 0.2 during the Analysis and Refinement phases and adjusted to 0.8 for the Sampling Generation phase. *"max_tokens"* is configured to 150 for Analysis, 100 per item in Sampling Generation, and 300 for Refinement. *"repetition_penalty"* is consistently set at 1.1 across all phases, and *"top_p"* is maintained at 1.0.

The configuration of these model parameters is established through testing with a very limited set of sample data from NQ validation dataset. However, given that the initial parameter choices were based on empirical estimates, there remains the potential for undiscovered, more effective parameter combinations.

B.3 Prompts

In this subsection, we provide a thorough description of the prompts utilized in all LLM-based QE methods employed in our experiment. Table 5 details the prompts used in Q2D, Q2E, and their variants, while Table 6 displays the prompts applied in each sub-module of AGR. The objective of this subsection is to clarify the precise nature of the prompts, facilitating a deeper understanding of our study. It is important to note that these prompts are configured for the Mistral-7B model, and slight adaptability adjustments may be necessary when applying them to other models.

B.4 Contextual References

In the experiment of AGR, we explore the impact of deduplicating the retrieved contextual references before proceeding with subsequent steps. The final outcomes reveal that the Top-5 recall rate stood at 68.37%, marginally different from the non-deduplicated result of 68.47%. However, the end-to-end Top-1 EM score was 36.52, significantly lower than the non-deduplicated result of 37.73. A possible explanation is that, without deduplication, the gold passages, being repeatedly retrieved across different QE candidates. This makes gold passages more significant informational weight, thereby benefiting the overall end-to-end answer generation.

Sub-Modules	Prompt
AGR-Analyze	"""Question: {query} Key Phrases: {answer_kp_analysis} Do not attempt to explain or answer the question, just provide the Question Analysis. Expected Output: "Question Analysis": Question Analysis based on Question and Key Phrases Output:"""
AGR-Generate	"""Question: {query} Question analysis: {answer_analysis} Based on the analysis and your available knowledge, create a possibly correct and concise answer that directly answers the question "{query}". Expected Output: "Answer": answer with a detailed context Output:"""
AGR-Generate with contextual references	"""Question: {query} Retrieval Context: {AGR_Retrieval_Docs} Based on the retrieval context and your available knowledge, create a possibly correct and concise answer that directly answers the question "{query}". Expected Output: "Answer": answer with a detailed context Output:"""
AGR-Refine	"""Question: {query} Candidate answer list: {Candidate_Answer_List} Based on the candidate answers and your available knowledge, please evaluate the accuracy and reliability of each candidate answer. Identify any misinformation or incorrect facts in the answers. Then, generate a correct and concise response that best answer the question, refer to the information from the candidate answers that you have verified as accurate. Expected Output: "Best Answer": a concise answer for the question "{query}" Output:"""

Table 6: Prompts for AGR sub-modules.