CIRAG: Retrieval-Augmented Language Model with Collective Intelligence

Chenxu Cui
Institute of Information Engineering, Chinese Academy of Sciences
School of Cyber Security, University of Chinese Academy of Sciences
State Key Laboratory of Cyberspace Security Defense
Beijing, China
cuichenxu@iie.ac.cn

Haihui Fan*
Institute of Information Engineering, Chinese Academy of Sciences
State Key Laboratory of Cyberspace Security Defense
Beijing, China
fanhaihui@iie.ac.cn

Jinchao Zhang
Institute of Information Engineering, Chinese Academy of Sciences
State Key Laboratory of Cyberspace Security Defense
Beijing, China
zhangjinchao@iie.ac.cn

Lin Shen
Institute of Information Engineering, Chinese Academy of Sciences
School of Cyber Security, University of Chinese Academy of Sciences
State Key Laboratory of Cyberspace Security Defense
Beijing, China
shenlin@iie.ac.cn

Bo Li
Institute of Information Engineering, Chinese Academy of Sciences
State Key Laboratory of Cyberspace Security Defense
Beijing, China
libo@iie.ac.cn

Weiping Wang
Institute of Information Engineering, Chinese Academy of Sciences
Beijing, China
wangweiping@iie.ac.cn

Abstract

Retrieval-augmented generation (RAG) paradigms can integrate external knowledge to enhance and validate the output of Large Language Models (LLMs) thereby mitigating generative hallucinations and broadening the model's knowledge scope. Despite advancements, existing RAG methods still suffer from uncertainty of prediction during the multi-round retrieval-generation process, and a lack of the ability to balance the adequacy and redundancy of retrieved information. To address these challenges, we propose CIRAG, an approach that combines the RAG process with collective intelligence. Inspired by the crowd of wisdom, CIRAG simulates individual independent decision-making and information aggregation within a crowd. Specifically, CIRAG first enhances retrieval diversity by expanding queries based on extracted entities, then combines frequency-based and semantic-based reranking to form a multi granularity fusion reranking thereby assessing better relevance, and integrate multiple information sources for accurate content generation. By undertaking these steps in an integrated manner, CIRAG enables the model to acquire comprehensive and non-redundant information for generating responses. We conduct extensive experiments with HotPotQA and 2WikiMultihopQA datasets, popular benchmark for retrieval-based, multi-step question-answering. Experimental results show that our approach surpasses existing advanced RAG framework while providing high portability in query expansion as well as strong comprehensiveness exhibited in the collective intelligence.

CCS Concepts

• **Information systems** → **Information retrieval**; **Language models**; **Question answering**.

Keywords

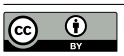
Retrieval-Augmented Generation, Information Retrieval, Natural Language Processing, Query Expansion, Reranking.

ACM Reference Format:
Chenxu Cui, Haihui Fan, Jinchao Zhang, Lin Shen, Bo Li, and Weiping Wang. 2025. CIRAG: Retrieval-Augmented Language Model with Collective Intelligence. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '25), July 13–18, 2025, Padua, Italy*. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3726302.3729921

1 Introduction

The emergence of large language models (LLMs), such as ChatGPT [22] and LLaMa [6], has significantly transformed the field of language understanding and generation [4] and thus has become a foundational component in various natural language processing tasks. Although LLMs have memorized a significant amount of world knowledge during pre-training or subsequent fine-tuning phases, they still tend to struggle with factual errors and hallucinations and create imaginary content. To address this issue, researchers introduce retrieval components, which look up relevant information from external knowledge resources, into LLMs to augment the generation ability of LLMs [17]. This process is called RAG

*Corresponding author.



SIGIR '25, Padua, Italy

ACM ISBN 979-8-4007-1592-1/2025/07
https://doi.org/10.1145/3726302.3729921

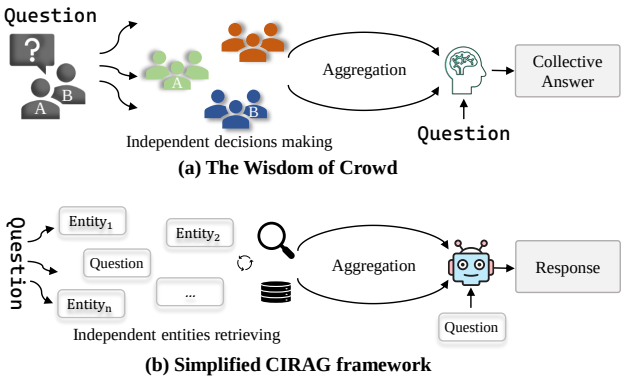


Figure 1: The correspondence between The Wisdom of Crowds and CIRAG framework.

and is a promising direction to address hallucination and expand the boundaries of knowledge of LLMs.

Originally, RAG follows a "retrieve-then-generate" paradigm where they retrieve information snippets based on the user's question, and then generate a complete answer by referencing on the retrieved information snippets. The optimization goals of RAG, which primarily focus on retrieval phase, are to obtain more critical evidence to better support downstream generation tasks. However, **balancing the adequacy and redundancy of the retrieved information remains a significant challenge**. From the retrieval perspective, existing methods mainly include three optimization directions: (1) **Pre-retrieval**, which focuses on mining or enriching the potential information contained in questions. Researchers have found that it is prevalent for users to pose questions with various sub-intents [31]. Thus, some researchers decompose the original query into sub-queries [14, 30, 34] or equip LLMs with tailored search queries across various scenarios [3]. (2) **Post-retrieval**, which filters retrieved results to enable they only contain evidence that supports answering the questions [33]. For example, researchers enhance the self-awareness of source relevance for LLMs, so as to adaptively utilize external knowledge in RAG systems [32] or compress the cluttered retrieved results into a compact set of crucial concepts [8, 28]. And (3) **Retrieving-while-generation**, which performs retrieval tasks when knowledge cannot satisfy the needs during generation. Researchers leverage forward-looking sentences [11] and partially generated contents [25, 38] as dynamic queries during generation or imitate human metacognitive process, iteratively retrieving, criticizing during the generation [40].

Although previous studies have made strides in improving the quality of generated answers, a fundamental limitation exists: Operations other than generation, such as decomposing the original query into sub-queries and criticizing retrieved or generated results, rely heavily on Language Models (LMs), which introduces uncertainty [9, 20, 39] and increases inference overhead [10, 36]. We argue that this limitation causes LLMs to produce more severe hallucinations when uncertainty arises. Humans ordinarily seek insights from others and synthesize all perspectives to reduce uncertainty. This practice demonstrates *the Wisdom of Crowds*, which utilizes collective decision-making to solve complex questions. In this paper, we leverage the underlying idea of the wisdom of crowds to enhance RAG, thereby enabling LLMs to recognize and eliminate uncertainty and generating a more precise answer. As illustrated in Figure 1(a), individuals in the crowd contribute their backgrounds and experiences, which are then integrated into collective decisions through aggregation mechanisms. Consequently, the collective answer, derived from the synthesis of these backgrounds and experiences, frequently demonstrates greater accuracy and efficacy compared to decisions made by individuals independently.

Inspired by the wisdom of crowds, we propose **CIRAG**, an approach that equips RAG with Collective Intelligence. As depicted in Figure 1(b), we model independent decision-making in collective wisdom as the separate retrieval of the query and the entities it contains. This targeted approach maximizes access to relevant information, thereby enhancing the effectiveness of generation. However, decisions from individuals are too subjective and one-sided which indicates that not all retrieval results contribute positively to the generation. Hence, we design a multi-granularity reranking mechanism to simulate aggregation, which reranks the context from the perspective of the original query, enabling a more comprehensive and detailed ranking process. This effectively balances global relevance with local specificity, ensuring the reranked top n retrieved contexts are optimally aligned with the query's intent.

Specifically, in this paper, our proposed CIRAG consists of three fundamental steps: (1) **Explicit Query Expanding and Retrieving**, which is modeled from independent decision making, firstly extracts entities and relationships from the query to facilitate query expansion, followed by independent retrieval. Explicit query expansion retrieval ensures the diversity, sufficiency and independence of retrieved information which provides information assurance for subsequent response generation. (2) **Multi-granularity Fusion Reranking**, which simulates independent decisions aggregation, combines frequency-based reranking and semantic-based reranking. The frequency-based approach reranks documents according to their occurrence frequency, under the assumption that higher frequency correlates with greater relevance. Meanwhile, the semantic-based method refines the ranking process by evaluating the interaction between the query and each document through cross-attention mechanisms. By integrating these two strategies, Multi-granularity Fusion Reranking provides a comprehensive perspective on the relevance between the query and the document, making it particularly effective in aggregating retrieved documents. (3) **Collective Generation** is a LLM-centric module, designed to generate more accurate and comprehensive content. At this stage, information from multiple individuals, along with question, is fed into a LLM. With contextual understanding ability, LLM generates content based on the information provided and its own knowledge.

Our contributions in this paper are summarized as:

(1) Inspired by The Wisdom of Crowds, we introduce a simple yet effective approach, **CIRAG**, that integrates LLMs with collective wisdom for QA tasks. CIRAG focuses on optimizing the retrieval phase to obtain information that supports answering user's question and reduce the use of language models that may produce uncertainty.

(2) We propose explicitly expanding queries through entities and introduce a novel multi-granularity fusion reranking method. Together, these approaches provide precise supporting information within the RAG framework to enhance the accuracy of answer generation.
(3) We evaluate the effectiveness of our approach on two publicly available, knowledge-intensive multi-hop question answering datasets. Experimental results show that CIRAG substantially improves the generative performance of LLMs and outperforms existing baselines.

2 Related Work

2.1 Retrieval-Augmented Generation for LLMs

Language models tend to struggle with factual and hallucinations or being constrained by static knowledge. RAG has been regarded as a promising direction to tackle these challenges, offering reliable grounding and the flexibility to access external knowledge bases.

According to the optimization goal of RAG, existing studies primarily can be organized into three categories: (1) Pre-retrieval focuses on mining or enriching user's potential intents contained in questions for retrieving enough information from the retrieval base [31]. For example, DSP [14] expresses high-level programs that bootstrap pipeline-aware demonstrations, search for relevant passages, and generate grounded predictions, systematically breaking down problems into small transformations that the LMs and Retrieval Models (RMs) can handle more reliably. IRCoT [34] interleaves retrieval with steps in a CoT, guiding the retrieval with CoT and in turn using retrieved results to improve CoT. (2) Post-retrieval aims at filtering retrieved results to ensure that they only contain evidence that supports answering the questions [33]. Researchers have demonstrated that adding redundant information can lead to inaccurate generation [13, 24]. Hence, REAR [32] proposes an improved training method based on bi-granularity relevance fusion and noise-resistant training, better utilizing external knowledge by effectively perceiving the relevance of retrieved documents. (3) Retrieval-while-generation refers to performing one or more retrieval tasks when the knowledge inherent in the model or the retrieved knowledge does not satisfy the needs during generation. For example, researchers leverage LMs to generate forward-looking sentences [11] or partially generated contents [25, 38], which are then combined with the original query to form a new query, and this new query is used to retrieved from retrieval base to alleviate insufficient knowledge issue.

2.2 Query Expansion

Query Expansion has been a critical area of research in information retrieval, natural language processing, and question answering systems. It involves modifying or transforming the user's original query into a more effective or precise query to improve retrieval performance or user satisfaction. The primary goal of query expansion is to enhance the quality of the results by overcoming issues like querying suboptimal query formulation.

Recent studies have explored query expansion in RAG. For instance, researchers introduce an extra small language model as query rewriter, which is trained using the feedback of the LLM by reinforcement learning [19] or equip LLMs with tailored search queries across various scenarios [3]. Some researchers decompose the original query into sub-queries [14, 30, 34] by prompting LLMs with Chain-of-Thought (CoT) paradigm [15]. And other researchers leverage LMs to generate forward-looking sentences [11] or partially generated contents [25, 38] to broaden the original query.

2.3 Reranking

Reranking is a critical component in information retrieval systems, where the initial set of retrieved documents is reordered to prioritize the most relevant items before they are used in downstream tasks, such as text generation. In the context of RAG, reranking plays a crucial role in improving the quality of the retrieved information by ensuring that the content fed into the generative model is both relevant and informative.

Early works on reranking in information retrieval primarily focused on improving the basic retrieval results through traditional ranking models. In recent years, reranking has evolved significantly with the adoption of more advanced techniques, including deep learning, attention mechanisms, and reinforcement learning, particularly in the context of RAG [2, 16, 26, 41]. These advancements show how reranking has evolved from traditional methods to more sophisticated techniques, significantly improving the effectiveness of RAG systems. By refining the retrieved information, reranking ensures that the generative model has access to the most relevant and semantically rich content, ultimately enhancing the quality of the generated output.

2.4 The Wisdom of Crowds

In recent years, the concept of the Wisdom of Crowds has garnered significant attention across various fields, including machine learning, economics, and social science. The theory [35] suggests that collective wisdom, when aggregated from a diverse group of individuals, can often lead to more accurate decisions and predictions than those made by individuals or experts. This phenomenon hinges on the diversity, independence, and decentralization of participants, as well as their ability to communicate and aggregate their opinions.

In this paper, we thoughtfully integrate the wisdom of the crowd with the retrieval phase in RAG. Specifically, we conceptualize the original retrieval process, where the query is directly used for retrieval, as the process of retrieving contexts for different entities. Simultaneously, we propose a novel Multi-Granularity Fusion Reranking method that integrates frequency-based and semantic-based reranking approaches to aggregate the retrieved context related to the query from a holistic perspective.

3 RAG with Collective Intelligence

Retrieval augmentation has become a promising method to mitigate the hallucination issues by dynamically retrieving relevant external information during the generation process, ensuring more accurate, fact-based outputs and enabling access to up-to-date knowledge. The key to RAG's effectiveness is its ability to augment the generation process with relevant, external information, which allows the model to overcome many limitations of traditional large language models. Although existing research has made progress, retrieval-augmented language models still face significant challenges in balancing adequacy and redundancy of the retrieved information.

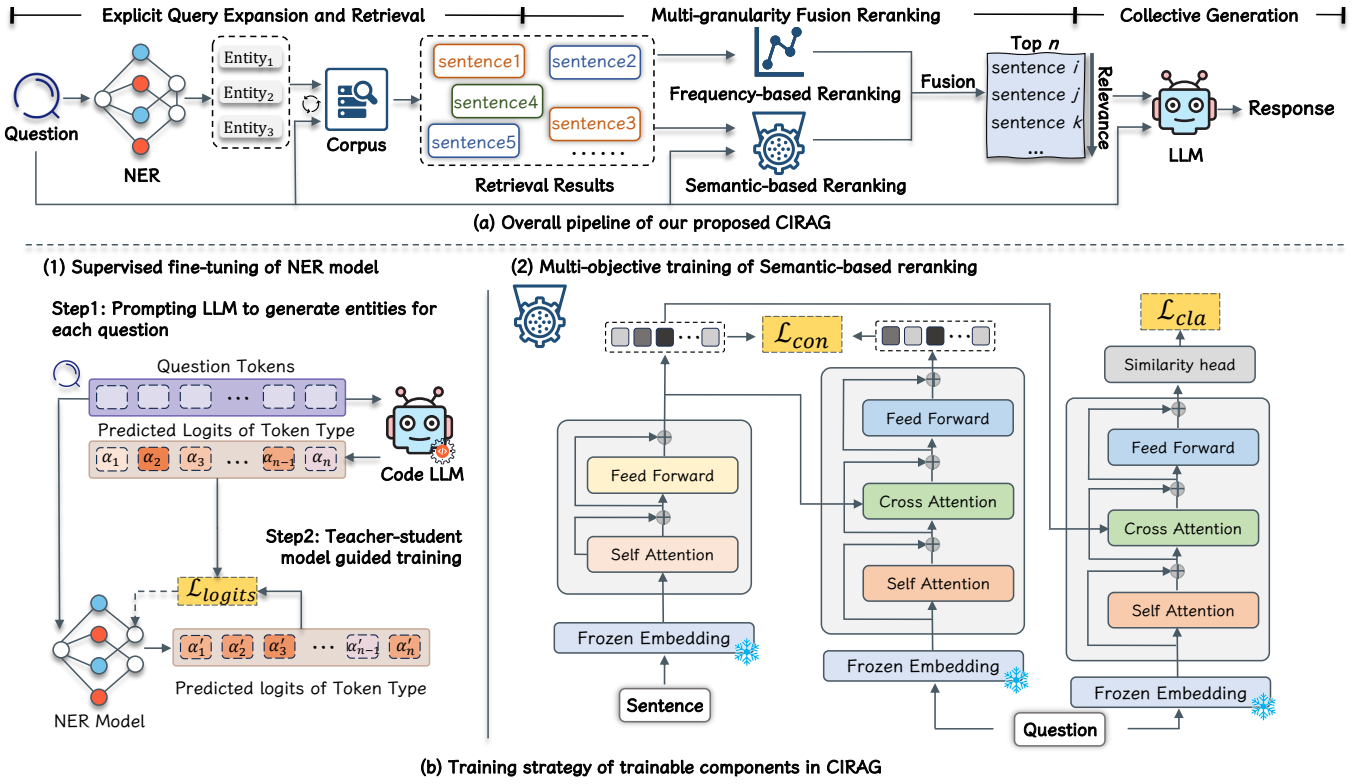


Figure 2: (a) is overall pipeline of our proposed CIRAG. (b) contains how we fine-tune our NER model (left) and our multi-objective training strategy of semantic-based Reranking (right). Model layers with the same color share the same parameters.

Motivated by this observation, in this section, we introduce a RAG with collective intelligence. This approach leverages collective intelligence, enabling the model to make more accurate decisions during the retrieval process. As shown in Figure 2(a), CIRAG comprises three main phases: (1) **Explicit Query Expanding and Retrieving**; (2) **Multi-granularity Fusion Reranking**; (3) **Collective Generating**. The following sections first define the problem and then introduce the details of these three steps.

3.1 Problem Definition

The vanilla RAG settings usually contains a knowledge corpus C, a retriever $\mathcal{R}$, and a frozen LLM serving as the generator $\mathcal{G}$. In this paper, we aim to integrate the wisdom of the crowd with the retrieval phase in RAG. Therefore, for a question q, we explicitly extend q with entities and their relationships it contains and regard the entities as individuals, denoted as

$$Inds = \{entity_1,\ entity_2,\ ...,\ entity_n\}.$$

These individuals have their own corresponding backgrounds and experiences, denoted as

$$Knowledge = \{DocSet_1,\ DocSet_2,\ ...,\ DocSet_n\},$$

where $DocSet_i$ is a document set derived by performing retrieval for each individual entity $entity_i$. However, not all backgrounds and experiences contribute positively to answer the question. We design a Multi-granularity Fusion Reranking module to identify content in *Knowledge* that supports answering the question at sentence-level, the result is denoted as:

$$Knowledge = MFR(\{sentence_1,\ sentence_2,\ ...,\ sentence_N\}),$$

where the sentence set combines all sentence segmented from every $DocSet_i$ in *Knowledge*. Finally, we combine the question q with the top n sentences in *Knowledge* and input them into the frozen language model generator $\mathcal{G}$ and get the final collective answer.

3.2 Explicit Query Expanding and Retrieving

In the wisdom of crowd, when a question arises, a diverse collection of independently deciding individuals is likely to be more extensive than single individual or even experts. Explicit query expanding and retrieving, which models from independent decision making, guarantees the diversity, sufficiency and independence of retrieved information which provides information assurance for subsequent response generation.

3.2.1 Explicit Query Expanding. Existing query rewriting or expansion methods decompose the original query into sub-queries by prompting LLMs with the CoT paradigm. However, these approach carry the risk of amplifying any errors contained in the decomposed sub-queries [11], as the LLMs are not always reliable [9, 20, 39].

To overcome this issue, we start from the original query and conduct a thoughtful investigation. Through self-attention analysis

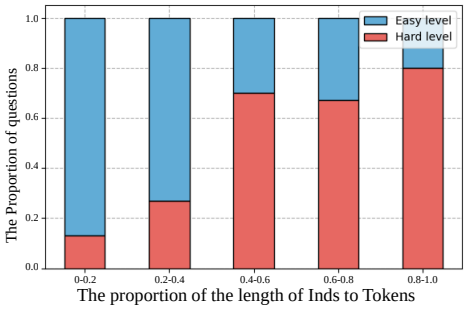


Figure 3: Statistics of entity proportion in questions at different difficulty levels.

of the query, we observe that certain tokens, such as persons and organizations, carry significantly higher importance in contributing to the query's overall meaning. In light of the limitations of existing methods and our observations, we posit that understanding the background contexts of entities within the query is essential for accurately answering it. Therefore, we enhance the query by explicitly incorporating its constituent entities and treating them independently. This process can be denoted as:

$$Inds = Model_{NER}(query),$$

where $Model_{NER}$ represents a Named Entity Recognition (NER) model, *Inds* stands for the extracted entity set.

3.2.2 Retrieving. In retrieval-augmented language model, retrieval helps enhance the capabilities of the generative model by fetching relevant external information from a knowledge base. However, retrieval is only needed when the problem becomes so complex that language model fails to provide proper answer.

To decide whether a retrieval process should be initiated, we randomly sampled 500 questions labeled as easy and 500 questions labeled as hard from four public datasets and extract the entities they contained, respectively. As shown in Figure 3, results indicate that in questions with easy level, entities accounts for a small proportion of total tokens. Conversely, a greater number of entities are present in questions that are hard to answer. With this mind, we establish a static difficulty classifier to decide whether retrieval should be triggered. The classifier can be defined as:

$$\text{Next step} = \begin{cases} \text{activate retrieving} & \text{if} \langle \frac{len(Inds)}{len(tokens)} \rangle > k, \\ \text{generate directly} & \text{otherwise.} \end{cases}$$

where *tokens* and $len(tokens)$ represent tokens in a question and the length of these tokens, respectively. $len(Inds)$ stands for the length of Inds. k serves as a threshold value governing the model's behavior. A higher value of k implies that the question may be harder for model to answer. In case where the proportion of the length of *Inds* to *tokens* falls below a certain threshold, the retrieval process and reranking process are triggered.

3.3 Multi-granularity Fusion Reranking

Explicit query expanding and retrieving phase provides sufficient background knowledge for generating the correct answer to the question. However, the background knowledge from individuals carries a certain degree of subjectivity in relation to the original question, indicating that not all of this knowledge is necessary. If all the background knowledge is input into the model indiscriminately, it will not only fail to improve the accuracy of response, but may even degrade the answer quality due to the noise [13, 21, 23, 24].

To reduce the disturbance of subjective information on the model's answer to the question, we propose a Multi-granularity Fusion Reranking (MFR) method that simulates the aggregation of independent decisions to eliminate subjective information. MFR encompasses frequency-based and semantic-based reranking, which will be elaborated upon below.

3.3.1 Frequency-based Reranking. In crowd of wisdom, most background and experience are varying among individuals. However, for a problem, they may exhibit a consensus which are crucial to solve this problem. The purpose of frequency-based reranking is to emulate this phenomenon and identify the consensus within the retrieval results.

Specifically, we receive the retrieval results from all *Inds* in the retrieval phase. In light of the variability in the retrieved documents, we divide each document into sentences to ensure that identical situations can appear in retrieval results. Following this division, we apply Biased Voting and Borda Count to the resulting $sentences = \{s_1, s_2, ..., s_n\}$. The Biased Voting is formulated as:

$$f_{weighted}(s_i) = f(s_i) \cdot w(s_i),$$

$$w(s_i) = \begin{cases} 2 & \text{if } s_i \in \mathcal{R}(\text{question}), \\ 1 & \text{otherwise.} \end{cases}$$

where $f(s_i)$ denotes the frequency of sentence s_i, $w(s_i)$ represents its corresponding weight. $\mathcal{R}(question)$ is the set of retrieval results for question. Here, to mitigate the retrieval drift issue associated with entity, we assign a weight of 2 to the retrieval results derived from the original question, while assigning a weight of 1 to the retrieval results obtained from all entities. Then, we obtain $sentences_{sorted} = \{s_1, s_2, ..., s_m\}$ where $m < n$ and $\forall i \in [0, m-1], f_{weighted}(s_i) \geq f_{weighted}(s_{i+1})$.

In simple majority system above, only the most-preferred sentences receives recognition, potentially disregarding the preferences for certain sentences in specific contexts. Building upon the situation, we adopt the Borda Count method to account for the overall ranking of all sentences, thereby avoiding potential biases caused by the frequent appearance of a few sentences in the final results. The Borda Count is represented as:

$$g(s_i) = m - rank(s_i),$$

where m is the length of $sentences_{sorted}$, $rank(s_i)$ is the rank of i-th sentence in $sentences_{sorted}$, $g(s_i)$ is the score of s_i given by Borda Count based on the rank.

Based on the weighted frequency and Borda count, the top k sentences can be selected by integrating these two factors. The combined score $G(s_i)$ is defined as:

$$G(s_i) = a \cdot f_{weighted}(s_i) + (1 - a) \cdot g(s_i),$$

where a is the weight coefficient that controls the relative importance of the weighted frequency and the Borda Count score.

3.3.2 Semantic-based Reranking. Frequency-based reranking helps mitigate the interference in generation caused by subjective information but lacks a comprehensive assessment of the relevance of retrieved information to the complete question. Therefore, we design a semantic-based reranking method, which takes into account the semantic similarity between the retrieved information and the entire context of the question. Specifically, we leverage the embedding layer of the pre-trained BERT model to map question and sentences into a high-dimensional vector space, generating their initial semantic representations, denoted as:

$$E_Q = BERT_{embed}(Q) \in \mathbb{R}^{l_Q \times d}, \quad E_{s_i} = BERT_{embed}(s_i) \in \mathbb{R}^{l_{s_i} \times d},$$

where Q stands for question, s_i is a single sentence in the retrieval results based on Q and the entities in Q. E_Q, E_{s_i} are semantic representations of Q and s_i, respectively. l_Q and l_{s_i} corresponds to the total number of tokens in Q and s_i. d is the embedding dimension. $BERT_{embed}$ denotes the embedding layer of BERT.

Then, following transformer structure, we employ an encoder to obtain global semantic information of sentence s_i, denoted as:

$$H^e_{s_i} = Encoder(E_{s_i}, E_{s_i}, E_{s_i}),$$

where *Encoder* represents a transformer encoder which each layer includes multi-head attention and feed-forward network, *Encoder* takes E_{s_i} as input and output $H^e_{s_i}$. And the encoder interacts with the E_Q and $H^e_{s_i}$ to determine the relevant feature representations between the question and the sentence. This are presented as:

$$H = Decoder(E_Q, H^e_{s_i}, H^e_{s_i}),$$

where H is representation that combines the semantic information of question and sentence. *Decoder* includes self-attention and cross-attention for interacting the feature representations of question and sentence.

The ultimate goal is to calculate the relevance score between question and sentence, we employ a fully connected network as similarity score head to derive the relevance score, statemented as:

$$H_{avg} = \frac{1}{l_Q} \sum_{i=1}^{l_Q} h^d_i,$$

$$R(s_i) = \text{sigmoid}(WH_{avg} + b),$$

where H_{avg} is the result of average pooling, h^d_i is output of the last hidden layer in H. $R(\cdot)$ calculates the relevance score between a question and a sentence by applying fully connected network to H_{avg} followed by a sigmoid activation. W and b are parameters of fully connected network.

Overall, the final relevance score between question and sentence depends on the frequency-based reranking score $G(s_i)$ and the semantic-based reranking score $R(s_i)$, denoted as:

$$R_{final} = \lambda G(s_i) + (1 - \lambda) R(s_i).$$

3.4 Collective Generating

After the aggregation of individual background knowledge is completed, the original question proposer reflects on the aggregation results to derive the final answer. Therefore, in our framework, once the next step provided by 3.2.2 *Retrieving* is to activate retrieving based on the input question, the retrieval model (e.g. BM25) is triggered to retrieve relevant information from external knowledge base. Suppose the retrieved and reranked top k sentences are denoted as $sentences_{top_k}$. Upon successful retrieval, the next step of CIRAG is to integrate this external knowledge into the LLM's generation process, as shown below:

$$answer = LLM(question, sentences_{top_k}, Prompt),$$

where LLM stands for a specific large language model, $Prompt$ is used to combine $question$ and its corresponding relevant sentences.

4 Experiment Setup

In this section, we introduce the task, data and evaluation metrics in the experiment, the implementation and training details of trainable part in our framework, the baseline approaches we compared with.

4.1 Task, Data and Evaluation Metrics

CIRAG targets the multi-hop retrieval question-answering task. In such task, the knowledge required to solve the problem is usually scattered in multiple passages from a given document corpus. For the experiment, we test CIRAG in HotPotQA [37] and 2WikiMultihopQA [7], which are widely used in open-domain multi-hop QA dataset. Following previous studies [11, 40], we sub-sample 1000 questions from the validation set of each dataset for experiments.

In retrieval phase, we utilize sentence-level recall (*R@n*) to evaluate our Explicit Query Expanding and Retrieving. In generation phase, for the entire response, we prompt another LLM(i.e. gpt-4o-2024-11-20) to determine whether the generated response and gold answer are semantically equivalent to achieve exact match (EM). For the tokens in response, following [40], we adopt token-level F1, precision (Prec.) and recall (Rec.) for comprehensive evaluation.

4.2 Implementation Details

We begin by outlining the implementation details of the RAG infrastructure, including the retriever, corpus, and generator. Following this, we provide the training details for the trainable components of our proposed method.

4.2.1 Details of RAG infrastructure. Since both HotPotQA and 2WikiMultihopQA predominantly depend on Wikipedia, we utilize the Wikipedia [12] to serve as corpus C, where articles are segmented into passages of 10 sentences. And we employ the BM25 algorithm [27] to implement our retriever. We choose Qwen2.5-7B-Instruct and Llama-3.1-8B-Instruct as our backbone generator.

4.2.2 Details of trainable components. Following previous study [5], we use pre-trained bert-base-NER[1] as our backbone NER model. We employ Qwen2.5-Coder-32B-Instruct [2] to process these tokens in quesion, and generate predict token logits, which correspond to the entity type predictions for each token in the input question. The logits generated by the LLM serve as soft targets, functioning as a teacher signal. These logits are then used to fine-tune the NER model, which acts as the student. We leverage the pre-trained LLM to generate high-quality supervision signals, which enhance the effectiveness of the fine-tuning process for the NER model.

[1] https://huggingface.co/dslim/bert-base-NER
[2] https://huggingface.co/Qwen/Qwen2.5-Coder-32B-Instruct

Table 1: Results on two multi-hop question answering datas. *Gen.* denotes different generators. R@5 refers to the accuracy of key sentences contained in the top five retrieved relevant documents. The best results are in bold and the second best results are underlined.

Method	*Gen.*	HotPotQA					2WikiMultihopQA				
		R@5	EM	F1	Prec.	Rec.	R@5	EM	F1	Prec.	Rec.
Without Retrieval											
vanilla model	llama3.1-8b	-	16.8	19.2	18.4	20.1	-	17.2	21.2	21.6	20.9
	Qwen-7b-Instruct		17.5	21.1	20.0	22.3		16.9	19.6	18.6	20.7
Chain-of-Thought	llama3.1-8b	-	21.7	25.6	24.7	26.5	-	21.2	25.0	24.3	25.7
	Qwen-7b-Instruct		20.9	27.5	26.6	28.4		20.2	26.2	25.1	27.3
With Retrieval											
vanilla RAG	llama3.1-8b	32.7	24.2	29.2	27.0	31.8	31.6	25.7	30.9	30.2	31.7
	Qwen-7b-Instruct		23.5	28.8	27.4	30.4		22.7	30.1	28.7	31.6
ReAct	llama3.1-8b	49.7	26.3	40.2	40.7	39.8	47.2	25.6	38.1	38.4	37.8
	Qwen-7b-Instruct	51.3	27.1	40.2	40.5	39.9	48.5	26.9	38.9	39.7	38.2
IR-Cot	llama3.1-8b	61.3	33.8	44.9	42.8	47.2	60.4	32.9	45.2	42.3	48.5
	Qwen-7b-Instruct	63.2	34.2	48.0	47.7	48.3	62.8	33.8	47.1	46.5	47.8
Flare	llama3.1-8b	53.5	31.2	40.3	40.0	40.6	50.3	30.6	41.2	39.8	42.7
	Qwen-7b-Instruct	51.8	30.6	42.1	42.3	41.8	52.6	31.2	41.2	41.9	40.6
Self-RAG	llama2-7b	58.2	32.5	43.1	43.0	43.2	56.8	31.4	43.1	42.1	44.2
CIRAG(ours)	llama2-7b	**71.8**	37.1	49.3	49.6	49.2	**70.3**	36.9	47.6	47.0	48.3
	llama3.1-8b		40.6	51.4	51.7	51.1		**41.8**	52.9	**53.1**	52.8
	Qwen-7b-Instruct		**42.9**	**54.4**	**52.8**	**56.1**		41.7	**53.7**	51.9	**55.7**

In 3.3.2 , we adopt the transformer architecture and modify it to develop our semantic similarity discrimination model to implement semantic-based reranking. We follow the settings outlined in the original paper and configure the number of layers in both the encoder and decoder to 6. Unlike training from scratch, we employ the pre-trained BERT embedding layer to initialize the embeddings in our model, thereby enhancing the semantic representation. Additionally, we incorporate a fully connected layer following the final decoder to map the output representation to a similarity score. We adopt multi-objective optimization to train the semantic similarity model. Specifically, we frame semantic similarity as a binary classification task, where a question-sentence pair is labeled as 1 if the sentence supports answering the question, and 0 otherwise. We select BCEWithLogitsLoss as the loss function, denoted as:

$$\mathcal{L}_{\text{classification}} = -\left[y \log(\sigma(\hat{y})) + (1-y)\log(1-\sigma(\hat{y}))\right],$$

where y is the gold label, $\hat{y}$ is the raw logit, $\sigma(\cdot)$ is the Sigmoid function. Furthermore, we employ contrastive learning to enhance the model's ability to discriminate between similarities, denoted as:

$$\mathcal{L}_{\text{contrastive}} = y\|H_e - H\|^2 + (1-y)\max(0, m - \|H_e - H\|)^2,$$

where y is the gold label, H_e and H are the output of encoder and decoder, respectively. $\|\cdot\|$ is Euclidean distance function. Therefore, the multi-objective training loss of our semantic similarity model is as follows:

$$\mathcal{L}_{\text{multi-objective}} = \mathcal{L}_{\text{classification}} + \mathcal{L}_{\text{contrastive}}.$$

4.3 Baselines

We first consider two closebook models: **Qwen2.5-7B-Instruction** [29], an instruction-tuned model designed to better understand and generate natural language responses tailored to specific task instructions, enhancing its ability to follow and execute user prompts and **Llama3.1-8B-Instruction** [6], an instruction-tuned variant of the LLaMA model, optimized to follow and respond to task-specific prompts with improved accuracy and relevance. We employ a zero-shot prompting method, where the model is instructed to generate answers directly based on the given questions, without retrieving any information beyond parametric knowledge in the model.

Additionally, **Chain-of-Thought** [34] provides LLM with examples that include the reasoning process, allowing the model to better understand the potential logic and sequential steps. **Vanilla RAG** [17] employs the query to retrieve multiple documents, and feed them into LLM for deriving answers.

Table 2: Comparision between Fine-tuned NER models and LLMs with different parameter size (Param.) in HotPotQA.

NER Model	Param.	EM	F1	Prec.	Rec.
Fine-tuned NER Models					
bert-base	110M	40.6	51.4	51.7	51.1
T5-large	770M	40.9	50.9	51.9	52.4
Large Language Models					
Qwen2.5-Coder	7B	40.8	51.1	51.4	53.1
CodeLLama	13B	41.1	52.1	52.3	53.6

We also reproduce several advanced, representative RAG methods: **IR-CoT** [34] combines information retrieval with chain-of-thought reasoning, where relevant information is retrieved and

used in a step-by-step reasoning process to generate more accurate answers. **Self-RAG** [1] leverages a model's own generated outputs as a source of information for further retrieval and reasoning, enhancing its ability to generate more coherent and contextually relevant answers. **ReAct** [38] enables agents to alternate between reasoning and interacting with external environments, using both reasoning and action to solve complex tasks more effectively. **Flare** [11] enhances language generation by actively retrieving relevant information throughout the generation process, using predictions of upcoming content to guide the retrieval and regeneration of low-confidence sentences.

5 Experiment

5.1 Main Experiment Results

The performance of various RAG methods are reported in Table 1, we summarize it as:

(1) Our proposed CIRAG, deploying different LLMs as generator, consistently surpasses all baseline methods across two datasets. Specifically, compared to the direct prompting-based approach, i.e., Vanilla model and Chain-of-Thought, The Chain-of-Thought approach enables model to engage in deeper reasoning to answer complex questions, yielding better performance compared to vanilla model. However, due to the static nature of the model's knowledge, the performance improvement achieved through the Chain-of-Thought approach is prone to reaching a bottleneck. CIRAG gains significant improvement by leveraging collective wisdom through collective generation. By obtaining more facts supporting answering questions through explicit query expanding and retrieving, CIRAG also makes significant progress compared to vanilla RAG while introducing little resource-intensive.

(2) When compared to IR-CoT, which achieves the second best performance, CIRAG demonstrates substantial improvements across all metrics. IR-CoT interleaves Chain-of-Thought generation with knowledge retrieval to guide the retrieval process through CoT, but its effectiveness is heavily dependent on the model's CoT generation capabilities. Under the same generation model setup, CIRAG significantly outperforms IR-CoT by incorporating explicit query expanding and retrieving. This highlights that our method is decoupled from the generation model and can be scaled to any model without causing performance degradation.

(3) Compared to those fine-tuned retrieval-augmented language model, which generate specific tokens to determine when to trigger the retrieval phase and discriminate the retrieval result it self, our proposed CIRAG shows inspiring performance. This indicates that our single-turn multiple retrieval through explicit query expanding and subsequent fusion reranking help retrieval-augmented language model to gain more sufficient information.

(4) The effectiveness of RAG lies in its ability to enhance generation by integrating relevant retrieved information with the model's generative capabilities. Therefore, we also evaluated the ability of different methods to recall key information. Our proposed method shows varying degrees of improvement compared to the baseline method. This indicates that using the collective retrieval method can obtain richer information.

5.2 Ablation study of Overall CIRAG

As shown in Table 3, we verify the function of CIRAG's infrastructure. By introducing explicit query expanding and retrieving, the performance of the retrieval-augmented language model has significantly improved, indicating that explicit query expanding and retrieving enables the model to access more relevant information related to the query, thereby guiding the model to provide more accurate answers. Next, we introduce a multi-granularity fusion reranking method to reorder the retrieved results and select the top n at sentence-level. The model performance has further improved, suggesting that while explicit query expanding and retrieving provides more relevant information, it may also introduce additional noise. Reranking the retrieved results helps mitigate this impact.

5.3 Ablation study of Trainable Components

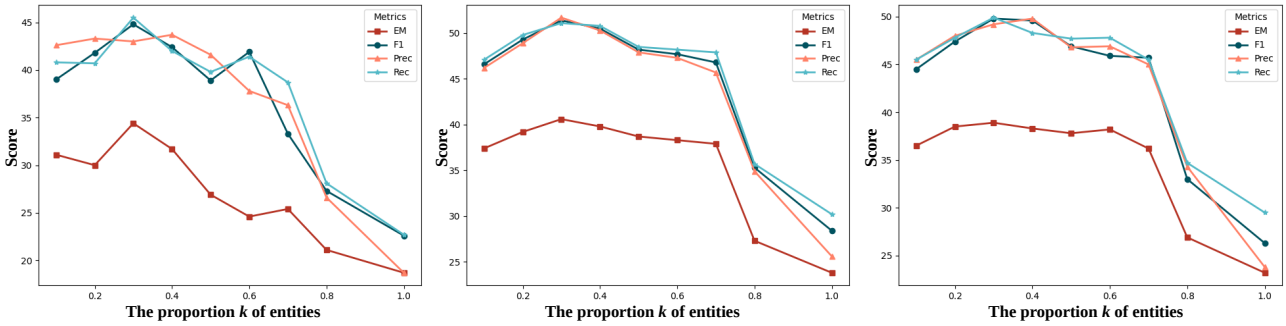
In this section, we study the trainable functional modules in Collective Intelligence and investigate the role of each module within the overall system.

5.3.1 The study of NER Model. LLMs have significantly advanced the field of Natural Language Processing (NLP), offering a unified framework for addressing a wide array of tasks. In this context, we explore the effect of employing LLMs as NER models on our proposed method. Previous studies have demonstrated that large code generation models can effectively perform NER tasks [18], therefore, we select two code language model, i.e., Qwen2.5-Coder and CodeLLama as NER model. Meanwhile, we fine-tune T5-large using the same strategy applied to our backbone NER model.

As illustrated in Table 2, utilizing fine-tuned models as the NER model in explicit query expanding and retrieving exhibit comparable performance compared to large language models. This suggests that fine-tuned NER models can offer entities as accurate as Code LLMs while with fewer paramters, meeting the efficiency and effectiveness requirements of NER. The performance of Qwen2.5-Coder and CodeLLama indicates that using LLMs to extract entities in query is a feasible approach, setting higher standards for model capabilities. Aside from the above, the performance of the fine-tuned T5-large model slightly surpasses that of our backbone NER model. This may be because our backbone model has not yet achieved optimal NER performance, and the larger number of parameters in T5-large helps to compensate for this limitation.

Table 3: The overall study of CIRAG, EQER stands for explicit query expanding and retrieving. *fre* and *sem* represent frequency-based reranking and semantic-based reranking, respectively.

Vanilla	EQER	FR		EM	F1	Prec.	Rec.
		fre	*sem*				
✓	-	-	-	24.2	29.2	27.0	31.8
✓	✓	-	-	30.8	42.0	41.6	42.5
✓	✓	✓	-	31.5	42.9	43.1	42.7
✓	✓	✓	✓	40.6	51.4	51.7	51.1



(a) Top n = 1 (b) Top n = 5 (c) Top n = 10

Figure 4: Experiment results in different proportion k of entities which determines whether to trigger the retrieval, and n which determines how many sentences selected to feed into LLM.

Table 4: Comparison of different training strategies, the $\mathcal{L}_{\text{multi-objective}}$ stands for combining the first two losses.

Training loss	EM	F1	Prec.	Rec.
$\mathcal{L}_{\text{classification}}$	38.8	45.7	45.3	46.2
$\mathcal{L}_{\text{contrastive}}$	37.5	44.9	44.7	45.2
$\mathcal{L}_{\text{multi-objective}}$	40.6	51.4	51.7	51.1

5.3.2 The study of Semantic-based reranking training strategy. The semantic-based reranking model aims to identify the most relevant sentences from the candidate sentences based on semantic similarity to the question. We adopt a multi-objective training strategy, including semantic classification and contrastive learning, to achieve better semantic judgment performance. To explore the necessity of these two objectives in the training strategy, we conduct ablation studies by comparing each single-objective training with multi-objective training.

The findings, depicted in Table 4, highlight that stripping away any single-objective straining strategy detrimentally impacts performance across all evaluation metrics. Among these, the omission of semantic classification results in the most pronounced decline in model efficiency. This suggests that using supervised learning for judging the semantic similarity between questions and sentences is reliable. The combination of contrastive learning and classification further improves performance, suggesting that contrastive learning enhances the model's ability to distinguish features and alleviate overfitting during the training of the classification task.

5.4 Ablation study of Key Hyperparameters

In this section, we focus on the impact of some key hyperparameters. We analyzed their effects through experiments, as detailed below.

5.4.1 The study of the threshold **k** *for triggering retrieval.* Simple questions can usually be answered directly using the model's existing parameters without the need for external information. In contrast, complex or open-ended questions, especially those requiring reasoning or involving multiple knowledge points, necessitate the retrieval stage. These questions typically rely on external knowledge bases to provide support, ensuring that the generated answers are sufficiently rich and accurate. In CIRAG, we assess the complexity of the question based on the proportion of entities it contains, and this guides the decision of whether to trigger the retrieval process. To validate the effectiveness of this, we conduct experiments to measure performance of the model under different k.

As presented in Figure 4 (b), where **Top n=5** represents that the top 5 relevant documents are returned during each retrieval, this is also the default setting in our experiments. Experimental results indicates that when the threshold k is set to 0.3, CIRAG achieves the best performance, indicating that this setting strikes a good balance in deciding whether to trigger retrieval. As k gradually increases, the proportion of entities required to trigger retrieval also increases, and when k reaches 1, it becomes vanilla model. When k is set below 0.3, simple questions also trigger retrieval which may introduce noisy information. However, the performance of CIRAG still improves, indicating that our fusion reranking module effectively retrieves highly relevant information from the already retrieved information, regardless of the complexity of the query.

6 Conclusion

In this paper, we propose CIRAG, a novel approach inspired by the wisdom of crowds, which connects LLM and information retrieval to accomplish complex reasoning tasks through collective intelligence. In CIRAG, entities in the query are considered independently and contribute their own retrieved documents. These documents are reranked by multi-granularity fusion reranking module, to better match the needs of LLM. Experimental results on two public datasets show that CIRAG outperforms baseline methods.

Acknowledgments

This work was supported by Special Task Project of the Ministry of Industry and Information Technology of China under Grant ZTZB-23-990-024.

References

[1] Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection. In *The Twelfth International Conference on Learning Representations*. https://openreview.net/forum?id=hSyW5go0v8

[2] Arian Askari, Mohammad Aliannejadi, Chuan Meng, Evangelos Kanoulas, and Suzan Verberne. 2023. Expand, Highlight, Generate: RL-driven Document Generation for Passage Reranking. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Singapore, 10087–10099. doi:10.18653/v1/2023.emnlp-main.623

[3] Chi-Min Chan, Chunpu Xu, Ruibin Yuan, Hongyin Luo, Wei Xue, Yike Guo, and Jie Fu. 2024. RQ-RAG: Learning to Refine Queries for Retrieval Augmented Generation. arXiv:2404.00610 [cs.CL] https://arxiv.org/abs/2404.00610

[4] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, and et al. 2024. PaLM: scaling language modeling with pathways. *J. Mach. Learn. Res.* 24, 1, Article 240 (mar 2024), 113 pages.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. doi:10.18653/v1/N19-1423

[6] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, and et al. 2024. The Llama 3 Herd of Models. arXiv:2407.21783 [cs.AI] https://arxiv.org/abs/2407.21783

[7] Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. *arXiv preprint arXiv:2011.01060* (2020).

[8] Sebastian Hofstätter, Jiecao Chen, Karthik Raman, and Hamed Zamani. 2022. FiD-Light: Efficient and Effective Retrieval-Augmented Text Generation. arXiv:2209.14290 [cs.CL] https://arxiv.org/abs/2209.14290

[9] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of Hallucination in Natural Language Generation. *ACM Comput. Surv.* 55, 12, Article 248 (mar 2023), 38 pages. doi:10.1145/3571730

[10] Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2023. LLMLingua: Compressing Prompts for Accelerated Inference of Large Language Models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Singapore, 13358–13376. doi:10.18653/v1/2023.emnlp-main.825

[11] Zhengbao Jiang, Frank Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active Retrieval Augmented Generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Singapore, 7969–7992. doi:10.18653/v1/2023.emnlp-main.495

[12] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick 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)*. Association for Computational Linguistics, Online, 6769–6781. doi:10.18653/v1/2020.emnlp-main.550

[13] Nora Kassner and Hinrich Schütze. 2020. Negated and Misprimed Probes for Pretrained Language Models: Birds Can Talk, But Cannot Fly. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 7811–7818. doi:10.18653/v1/2020.acl-main.698

[14] Omar Khattab, Keshav Santhanam, Xiang Lisa Li, David Hall, Percy Liang, Christopher Potts, and Matei Zaharia. 2023. Demonstrate-Search-Predict: Composing retrieval and language models for knowledge-intensive NLP. arXiv:2212.14024 [cs.CL] https://arxiv.org/abs/2212.14024

[15] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2024. Large language models are zero-shot reasoners. In *Proceedings of the 36th International Conference on Neural Information Processing Systems* (New Orleans, LA, USA) *(NIPS '22)*. Article 1613, 15 pages.

[16] Sarawoot Kongyoung, Craig Macdonald, and Iadh Ounis. 2022. monoQA: Multi-Task Learning of Reranking and Answer Extraction for Open-Retrieval Conversational Question Answering. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 7207–7218. doi:10.18653/v1/2022.emnlp-main.485

[17] Patrick Lewis, Ethan Perez, Aleksandra Piktus, and et al. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 9459–9474.

[18] Peng Li, Tianxiang Sun, Qiong Tang, Hang Yan, Yuanbin Wu, Xuanjing Huang, and Xipeng Qiu. 2023. CodeIE: Large Code Generation Models are Better Few-Shot Information Extractors. Association for Computational Linguistics, Toronto, Canada, 15339–15353. doi:10.18653/v1/2023.acl-long.855

[19] Xinbei Ma, Yeyun Gong, Pengcheng He, hai zhao, and Nan Duan. 2023. Query Rewriting in Retrieval-Augmented Large Language Models. In *The 2023 Conference on Empirical Methods in Natural Language Processing*. https://openreview.net/forum?id=gXq1cwkUZc

[20] Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When Not to Trust Language Models: Investigating Effectiveness of Parametric and Non-Parametric Memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Toronto, Canada, 9802–9822. doi:10.18653/v1/2023.acl-long.546

[21] Kanishka Misra, Julia Rayz, and Allyson Ettinger. 2023. COMPS: Conceptual Minimal Pair Sentences for testing Robust Property Knowledge and its Inheritance in Pre-trained Language Models. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, Dubrovnik, Croatia, 2928–2949. doi:10.18653/v1/2023.eacl-main.213

[22] OpenAI. 2024. GPT-4 Technical Report. arXiv:2303.08774 [cs.CL] https://arxiv.org/abs/2303.08774

[23] Ruotong Pan, Boxi Cao, Hongyu Lin, Xianpei Han, Jia Zheng, Sirui Wang, Xunliang Cai, and Le Sun. 2024. Not All Contexts Are Equal: Teaching LLMs Credibility-aware Generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Miami, Florida, USA, 19844–19863. doi:10.18653/v1/2024.emnlp-main.1109

[24] Fabio Petroni, Patrick Lewis, Aleksandra Piktus, Tim Rocktäschel, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. 2020. How Context Affects Language Models' Factual Predictions. In *Automated Knowledge Base Construction*. https://openreview.net/forum?id=025X0zPfn

[25] Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah Smith, and Mike Lewis. 2023. Measuring and Narrowing the Compositionality Gap in Language Models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*. Association for Computational Linguistics, Singapore, 5687–5711. doi:10.18653/v1/2023.findings-emnlp.378

[26] Leigang Qu, Meng Liu, Wenjie Wang, Zhedong Zheng, Liqiang Nie, and Tat-Seng Chua. 2023. Learnable Pillar-based Re-ranking for Image-Text Retrieval. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Taipei, Taiwan) *(SIGIR '23)*. New York, NY, USA, 1252–1261. doi:10.1145/3539618.3591712

[27] Stephen Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Found. Trends Inf. Retr.* 3, 4 (April 2009), 333–389. doi:10.1561/1500000019

[28] Kaize Shi, Xueyao Sun, Qing Li, and Guandong Xu. 2024. Compressing Long Context for Enhancing RAG with AMR-based Concept Distillation. arXiv:2405.03085 [cs.CL] https://arxiv.org/abs/2405.03085

[29] Qwen Team. 2024. Qwen2.5: A Party of Foundation Models. https://qwenlm.github.io/blog/qwen2.5/

[30] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. Interleaving Retrieval with Chain-of-Thought Reasoning for Knowledge-Intensive Multi-Step Questions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Toronto, Canada, 10014–10037. doi:10.18653/v1/2023.acl-long.557

[31] Shuting Wang, Xin Yu, Mang Wang, Weipeng Chen, Yutao Zhu, and Zhicheng Dou. 2024. RichRAG: Crafting Rich Responses for Multi-faceted Queries in Retrieval-Augmented Generation. *CoRR* abs/2406.12566 (2024).

[32] Yuhao Wang, Ruiyang Ren, Junyi Li, Wayne Xin Zhao, Jing Liu, and Ji-Rong Wen. 2024. REAR: A Relevance-Aware Retrieval-Augmented Framework for Open-Domain Question Answering. arXiv:2402.17497 [cs.CL] https://arxiv.org/abs/2402.17497

[33] Zhiruo Wang, Jun Araki, Zhengbao Jiang, Md Rizwan Parvez, and Graham Neubig. 2023. Learning to Filter Context for Retrieval-Augmented Generation. arXiv:2311.08377 [cs.CL] https://arxiv.org/abs/2311.08377

[34] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2024. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems* (New Orleans, LA, USA) *(NIPS '22)*. Curran Associates Inc., Red Hook, NY, USA, Article 1800, 14 pages.

[35] Wikipedia contributors. 2024. The Wisdom of Crowds — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=The_Wisdom_of_Crowds&oldid=1255403153. [Online; accessed 25-December-2024].

[36] Fangyuan Xu, Weijia Shi, and Eunsol Choi. 2024. RECOMP: Improving Retrieval-Augmented LMs with Context Compression and Selective Augmentation. In *The Twelfth International Conference on Learning Representations*. https://openreview.net/forum?id=mlJLVigNHp

[37] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600* (2018).

[38] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing Reasoning and Acting in Language Models. In *The Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=WE_vluYUL-X

[39] Gal Yona, Roee Aharoni, and Mor Geva. 2024. Can Large Language Models Faithfully Express Their Intrinsic Uncertainty in Words? arXiv:2405.16908 [cs.CL] https://arxiv.org/abs/2405.16908

[40] Yujia Zhou, Zheng Liu, Jiajie Jin, Jian-Yun Nie, and Zhicheng Dou. 2024. Metacognitive Retrieval-Augmented Large Language Models. In *Proceedings of the ACM*

Web Conference 2024 (Singapore, Singapore) *(WWW '24)*. Association for Computing Machinery, New York, NY, USA, 1453–1463. doi:10.1145/3589334.3645481

[41] Xiaozhi Zhu, Tianyong Hao, Sijie Cheng, Fu Lee Wang, and Hai Liu. 2022. A Self-supervised Joint Training Framework for Document Reranking. In *Findings of the Association for Computational Linguistics: NAACL 2022*. Association for Computational Linguistics, Seattle, United States, 1056–1065. doi:10.18653/v1/2022.findings-naacl.79