An Iterative Utility Judgment Framework Inspired by Philosophical Relevance via LLMs

Anonymous ACL submission

Abstract

Relevance and utility are two frequently used measures to evaluate the effectiveness of an information retrieval (IR) system. Relevance emphasizes the aboutness of a result to a query, while utility refers to the result's usefulness or value to an information seeker. In Retrieval-Augmented Generation (RAG), high-utility results should be prioritized to feed to LLMs due to their limited input bandwidth. Re-examining RAG's three core components—relevance ranking derived from retrieval models, utility judgments, and answer generation—aligns with Schutz's philosophical system of relevances, which encompasses three types of relevance representing different levels of human cognition that enhance each other. These three RAG components also reflect three cognitive levels for LLMs in question-answering. Therefore, we propose an Iterative utiliTy judgmEnt fraMework (ITEM) to promote each step in RAG. We conducted extensive experiments on retrieval (TREC DL, WebAP), utility judgment task (GTI-NQ), and factoid question-answering (NQ) datasets. Experimental results demonstrate significant improvements of ITEM in utility judgments, ranking, and answer generation upon representative baselines[1].

1 Introduction

Relevance and utility are two frequently used measures to evaluate Information Retrieval (IR) performance (Saracevic, 1996, 1975; Saracevic et al., 1988). Relevance usually refers to *topical relevance* that measures the degree of fit between the subjects of a query and retrieved items, and the criteria of "aboutness" is used (Saracevic et al., 1988; Schamber and Eisenberg, 1988). In contrast, *utility* refers to the "usefulness" of retrieval items to an information seeker, of which the criterion is their "value" to the user (Saracevic, 1996; Saracevic et al., 1988). As the example from the TREC DL dataset shown on Figure 1, topical relevance does not necessarily mean utility, while utility indicates a higher standard of relevance. Since topical relevance is relatively easy to observe and measure (Schamber et al., 1990), the studies of IR models have been primarily focused on improving relevance for a long time (Bruce, 1994).

Question: Who killed **Nicholas II** of Russia?

[1] Nicholas II was the last emperor, or tsar, of Russia, serving from 1894 to 1917. Nicholas, his wife, and their five children were killed by the Bolsheviks, members of a revolutionary group (led by Lenin) ...(**Perfectly relevant->Utility**)

[2] How was Tsar Nicholas II of Russia killed? Along with his family, Tsar Nicholas II was shot by a firing squad in the year of 1918, in a house in Ekaterinburg. The Romanovs' bodies were buried near the location of their death but have since been exhumed and venerated in the Peter-and-Paul Cathedral in St. Petersburg, Russia as martyred saints. (**Highly relevant**)

[3]In 1881, Nicholas' father, Alexander III, became czar (emperor) of Russia after his father, Alexander II, was killed by an assassin's bomb. Nicholas, at twelve, witnessed his grandfather's death when the czar,horribly maimed, was carried back to the palace.(**Irrelevant**)

Figure 1: An example between utility and relevance from TREC DL dataset.

In the modern LLM era, Retrieval-Augmented Generation (RAG) has become a hot research topic that equips LLMs with external knowledge (Xie et al., 2023; Shi et al., 2023; Izacard et al., 2023; Su et al., 2024; Glass et al., 2022). Given the constrained bandwidth of LLM inputs, it is essential to prioritize high-value results to guide LLMs. Consequently, utility needs to be emphasized more than topical relevance in RAG. More recently, Zhang et al. (2024) highlighted the use of LLMs for utility judgments. In this paper, we aim to further promote the utility judgment performance of LLMs so that RAG can be enhanced by high-utility references.

Schutz's Philosophical Theory of Relevance. Relevance is foundational in information retrieval (IR) and remains widely debated (Mizzaro, 1998). Saracevic (1996) discussed the nature of relevance in the IR system as the effectiveness of interactive exchange on different levels, and they are non-independent interdependencies, which are primarily influenced by Schutz's philosophical theory of relevance. Schutz considered relevance as the property that determines the connections and relations in our lifeworld. He identified three types of basic

[1]Our code and benchmark can be found at https://anonymous.4open.science/r/ITEM-B486/.

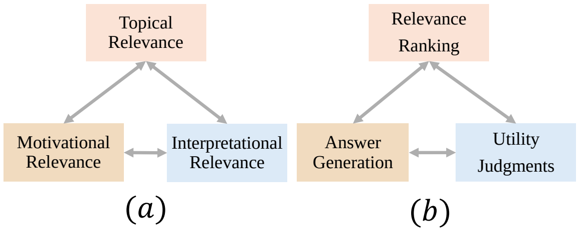

Figure 2: (a): Schutz's "system of relevancies", (b): the relation of each relevance to the components in RAG. The same color in the two frameworks is the corresponding connection.

and interdependent relevance that interact dynamically within a "system of relevancies" (Saracevic, 1996; Schutz, 1970): (i) Topical relevance, which refers to the perception of what is separated from one's experience to form one's present object of concentration; (ii) Interpretational relevance, which involves the past experiences in understanding the currently concerned object; and (iii) Motivational relevance, which refers to the course of action to be adapted based on the interpretations. The motivational relevance, in turn, helps obtain additional materials to become a user's new experience, which further facilitates topical and interpretational relevance. Schutz posited that one's perception of the world may be enhanced by this dynamic interaction, as shown in Figure 2. By incorporating utility judgments into RAG, we can re-examine its three components: topical relevance or relevance ranking derived from retrieval models, utility judgments, and answer generation. Topical relevance is an emerging focus on a topic, utility is the deeper understanding of the topic, and answers indicate the final solution based on the interpretations and will guide users' actions. *Therefore, topic relevance, utility, and derived answer also reflect three cognitive levels for LLMs in question-answering, from low to high, i.e., aboutness, the value for deriving an answer, and the derived answer.*

Iterative utiliTy judgmEnt fraMework (ITEM). Inspired by the philosophical theory of relevance, we believe the dynamic interactions between the three components in RAG can promote the performance of each step. To verify the idea, we leverage LLMs to perform each step in RAG shown in Figure 2, i.e., relevance ranking, utility judgments (classification), and answer generation. We propose an Iterative utiliTy judgmEnt fraMework (ITEM) to enhance the utility judgment and QA performance of LLMs by interactions between the steps. ITEM has two variants depending on whether relevance ranking is involved in the iterations. We are curious to see which option will be better for the tasks: fewer iterations with more components in an iteration, more iterations with fewer components in an iteration, or more iterations with more components.

We experiment on various information-seeking tasks, i.e., multi-grade passage retrieval on TREC DL (Craswell et al., 2020), multi-grade non-factoid answer passage retrieval on WebAP (Yang et al., 2016), utility judgments benchmark on GTI-NQ (Zhang et al., 2024), and factoid QA on NQ (Kwiatkowski et al., 2019). For multi-grade passage retrieval, we consider the ones with the highest grade to be of utility and we focus on the performance of utility judgments and topical relevance ranking. For factoid QA, we emphasize the answer accuracy. Experimental results have demonstrated that ITEM can significantly outperform competitive baselines, including various single-shot judgment approaches in terms of utility judgments, ranking of topical relevance, and answer generation, which confirms the viability of adaptation of Schutz's viewpoint of the relevance system into RAG. We also find that: 1) for difficult tasks (i.e., utility judgments of non-factoid answer passages in WebAP) and complicated candidate passage list (i.e., GTI-NQ), more components in the iteration and multiple iterations are usually more beneficial; 2) for multi-grade relevance ranking tasks, using utility as ranking criterion leads to significantly better multi-grade relevance performance than relevance ranking results even when utility judgment is involved in the iteration; 3) for factoid QA tasks, more iterations with fewer components performs the best, indicating that more components and more iterations are not always needed, especially for simpler tasks.

2 Related Work

Multi-dimensional relevance. The concept of "relevance" is central to information retrieval theory. Researchers have extensively debated its definition and measurement (Mizzaro, 1997). Early approaches primarily defined and assessed relevance through exact term matching (Vickery, 1959) or logical entailment (Hillman, 1964). However, subsequent empirical studies revealed the limitations of system-oriented relevance analysis, prompting diverse perspectives on relevance (Saracevic, 1975; Swanson, 1986; Saracevic, 1996; Lancaster, 1968; Goffman and Newill, 1964; Kemp, 1974; Bruce, 1994). For example, Cooper (1971) introduced logical relevance and utility. Saracevic (1996) summarized five frameworks for information science: systems, communication, situational, psychological, and interaction frameworks, and categorized five distinct types of relevance, i.e., 1) system or algorithmic relevance; 2) topical or subject relevance; 3) cognitive relevance or pertinence; 4) situational relevance or utility; and 5) motivational or affective relevance. Bruce (1994) explored cognitive dimensions of relevance. Over time, scholarly consensus has coalesced around two primary perspectives: the system view and user view, with topical relevance and utility serving as their respective representative frameworks.

Utility-Focused Information Retrieval. Utility is a distinct measure of relevance compared to topical relevance (Zhao et al., 2024; Saracevic et al., 1988; Saracevic, 1975, 1996; Ji et al., 2024; Zhang et al., 2023), and more recently, Zhang et al. (2024) highlighted the use of LLMs for utility judgments. However, Zhang et al. (2024) only conducted a preliminary exploration of LLMs in utility judgments. Our work aims to further explore how to improve the performance of utility judgments for LLMs.

Retrieval-Augmented Generation (RAG). RAG approaches are widely employed to mitigate the hallucination issues in large language models (LLMs) (Xie et al., 2023; Zhou et al., 2024; Su et al., 2024). The current RAG approaches are categorized as follows: (i) single-round retrieval (Borgeaud et al., 2022; Lewis et al., 2020; Glass et al., 2022; Izacard et al., 2023; Shi et al., 2023), which involves using the initial input as a query to retrieve information from an external corpus and then the information is incorporated as part of the input for the model; and (ii) multi-round retrieval (Su et al., 2024; Jiang et al., 2023b; Ram et al., 2023; Khandelwal et al., 2020; Trivedi et al., 2023), which need multi-round retrieval based on feedback from LLMs.

Iterative Relevance Feedback via LLMs. Recent works (Li et al., 2023; Shao et al., 2023) have achieved great success in using LLMs to obtain the information needs of the question as pseudo-relevance feedback for iterative retrieval. They posit that a single retrieval may not yield comprehensive information, thus requiring multiple retrievals. In contrast, our methodology involves making iterative utility judgments on the results obtained from a single retrieval. Given the substantial operational costs associated with retrieval systems, the expense incurred from conducting multiple retrievals for a single query becomes even more prohibitive.

3 Utility Judgments (UJ) via LLMs

Schutz's philosophy of relevances encompasses three types of relevance: topical, interpretational, and motivational relevance, representing different levels of human cognition, and their dynamic interactions can enhance each other. By incorporating utility judgments into RAG and re-examining its three components, i.e., relevance ranking derived from the retrieval models, utility judgments, and answer generation, we realize they closely correspond to Schutz's philosophical system of relevance. Topic relevance, utility, and derived answer also reflect three cognitive levels for LLMs in question-answering, from low to high, i.e., aboutness, the value for deriving an answer, and the derived answer. Inspired by Schutz's theory, we presume they can also interact with each other and enhance each other. Therefore, we propose an Iterative utiliTy judgmEnt fraMework (ITEM) for utility judgments.

3.1 Notations and Definitions

Given a question q and a list of retrieved passages $\mathcal{D} = [p_1, p_2, ..., p_n]$ based on topical relevance, the goal of utility judgments for LLMs is to identify a set of passages $U = \{p_1, ..., p_m\}$, m is the number of passage with utility judged by LLMs. There are two typical input approaches for LLMs to construct the set U: pointwise and listwise: The pointwise approach independently evaluates the utility of individual passages, whereas the listwise method assesses the utility of multiple passages with the list input.

3.2 Single-Shot Utility Judgments

The most common approach to judge utility for the LLM is to perform a single-shot utility judgment, i.e., $U = LLM(q, \mathcal{D}, I)$, where I is the instruction. Zhang et al. (2024) proposed to generate a pseudo-answer a while conducting the utility judgment task, which can help LLMs to judge utility better, i.e., $a, U = LLM(q, \mathcal{D}, I)$.

3.3 Iterative utiliTy judgmEnt fraMework (ITEM)

Inspired by Schutz's theory of relevance in philosophy, we propose an Iterative Utility Judgment Framework (ITEM) for RAG. As illustrated in Figure 3, our framework dynamically iterates among

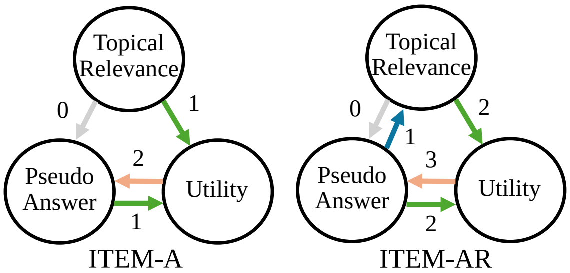

Figure 3: Flowchart illustrating the first iteration of ITEM. For ITEM-A, the process involves Step 1 (pseudo-answer generation) followed by Step 2 (utility judgments). For ITEM-AR, the process includes Step 1 (pseudo-answer generation), Step 2 (relevance ranking), and Step 3 (utility judgments). Subsequent iterations alternate between these steps.

Answer generation instruction

Implicit answer: To answer the question, output what information is necessary to answer the question based on the references.
Explicit answer: Answer the following question based on the given information with one or few words/sentences.

Figure 4: I_a instruction contains the *implicit answer* and *explicit answer*.

topical relevance ranking, pseudo-answer generation, and utility judgments. We propose two types of loops where two or three components in RAG interact with each other iteratively (ITEM-A and ITEM-AR).

ITEM with Answering in the Loop (ITEM-A). Formally, at each iteration t ($t \geq 1$), given the pseudo answer a_t generated based on the utility judgment result U_{t-1} from the previous iteration, we perform utility judgments on the candidate passages list $\mathcal{D}$ to obtain a set of passages with utility U_t:

$$a_t = LLM(q, U_{t-1}, I_a), \quad (1)$$

$$U_t = LLM(q, \mathcal{D}, a_t, I_u), \quad (2)$$

where I_a represents the answer prompts for LLMs (as detailed in Figure 4), a_t can be in two forms (details are shown in the Figure 4): (i) *explicit answer* to the question q; (ii) *implicit answer* that specifies the necessary information to answer the question q. I_u denotes the utility judgment prompts for LLMs (as detailed in Figure 5). We consider $U_0 = \mathcal{D}$ as the initial candidate set, where $\mathcal{D}$ represents the initial results ranked by a retriever such as BM25 (Robertson et al., 2009).

ITEM with both Answering and Ranking of Topical Relevance in the Loop (ITEM-AR). In the ITEM-A framework, topical relevance is not updated during the iteration process. To incorporate dynamic updating of topical relevance, we integrate

Utility judgments instruction

Listwise: Directly output the passages you selected that have utility in generating the reference answer to the question.
Pointwise: Directly output whether the passage has utility in generating the reference answer to the question or not.
The **requirements** for judging whether a passage has utility in answering the question are: The passage has utility in answering the question, meaning that the passage not only be relevant to the question, but also be useful in generating a correct, reasonable and perfect answer to the question. Directly output the passages you selected that have utility in generating the reference answer to the question.

Figure 5: I_u instruction contains listwise and pointwise approaches.

Relevance ranking instruction

Reference answer: {answer}
Rank the {num} passages above based on their relevance to the query. The passages should be listed in descending order using identifiers.

Figure 6: I_r instruction

a relevance ranking task into the ITEM framework, ensuring that all three tasks are executed in a loop. Formally, at iteration t ($t \geq 1$), the answer a_t is generated based on the judging result U_{t-1} from the previous iteration. Subsequently, given a_t, the passage list R_{t-1} from the previous iteration is ranked based on the relevance to the question, yielding a new ranked list R_t. Finally, the judging result U_t is derived using the ranked list R_t and the answer a_t:

$$a_t = LLM(q, U_{t-1}, I_a), \quad (3)$$

$$R_t = LLM(q, R_{t-1}, a_t, I_r), \quad (4)$$

$$U_t = LLM(q, R_t, a_t, I_u), \quad (5)$$

where I_r is the relevance ranking prompt for LLMs (as detailed in Figure 6), respectively.

Overall. At iteration t, we have two ways to produce the set U_t: (i) `Set-based approach`: Asking LLMs to identify the set of passages that have utility using listwise and pointwise input forms, which called ITEM-A$_s$ or ITEM-AR$_s$ variants; (ii) `Rank-based approach`: Requesting LLMs to provide a ranked passage list based on utility (utility ranking prompt is shown in Appendix H.2) using the listwise input approach and considering the top-k passages in the list to build U_t, which called ITEM-A$_r$ or ITEM-AR$_r$ variants. We set $k = 5$ and more details of k are shown in Appendix A.3. We find that ITEM-AR$_r$ does not improve ranking performance as well as ITEM-A$_r$ (see Appendix A.4 for experimental analysis), so we do not employ ITEM-AR$_r$ in the ranking experiment. The rank-based approach has poor performance on the utility judgment task (details can be found in Appendix A.4), so we only show the performance of the set-based approach on the utility judgment task.

We stop the iteration when at most m (m=3 in our paper) iterations are reached or the set of selected passages does not change, i.e., $t = m$ or $U_t = U_{t-1}$. Full details of all prompts can be found in Appendix H.

4 Experimental Setup

4.1 Datasets

Our experiments are conducted on four benchmark datasets, including two retrieval datasets, i.e., TREC DL (Craswell et al., 2020) and WebAP (Yang et al., 2016), a utility judgment dataset, i.e., GTI-NQ (Zhang et al., 2024), and an open-domain question answer (ODQA) dataset, i.e., NQ (Kwiatkowski et al., 2019). Detailed statistics of the experimental datasets are shown in Table Appendix C. We use two representative retrievers to gather candidate passages in $\mathcal{D}$ for utility judgments on TREC DL, WebAP, and NQ datasets. Construction details can be found in Appendix F.

TREC DL. We use the TREC-DL19 and TREC-DL20 datasets (Craswell et al., 2020). Judgments of TREC DL are on a four-point scale, i.e., "perfectly relevant", "highly relevant", "related", and "irrelevant". We consider the passages that are "perfectly relevant" to have utility. We filter questions of two datasets that contain the passages labeled "perfectly relevant' and combine them to form a whole dataset, i.e., the TREC DL.

WebAP. WebAP (Yang et al., 2016) is a non-factoid answer passage collection built on Gov2. Non-factoid questions usually require longer answers, such as sentence-level or passage-level (Keikha et al., 2014a; Yang et al., 2016; Keikha et al., 2014b). Relevant passages are annotated and categorized as "perfect", "excel", "good", and "fair". Similarly to TREC DL, we considered the "perfect" passages to have utility.

NQ. Natural Questions (NQ) consist of factoid questions issued to the Google search engine (Kwiatkowski et al., 2019). Each question is annotated with a long answer (typically a paragraph) and a short answer (one or more entities). Following Zhang et al. (2024), we use the questions that have long answers in our experiments.

GTI-NQ. Ground-truth inclusion (GTI) benchmark is constructed by Zhang et al. (2024) for utility judgment task. The GTI-NQ constructs a candidate passage set of 10 passages for each query sourced from the NQ dataset, comprising the long answer (designated as the utility passage), highly relevant noisy passages, weakly relevant noisy passages, and counterfactual passages.

4.2 Evaluation metrics

For the utility judgments task, we evaluate the results of judgments using **P**recision, **R**ecall, and micro **F1**. For the ranking task, we use the normalized discounted cumulative gain (**NDCG**) (Järvelin and Kekäläinen, 2017) to evaluate the ranking performance. For the answer generation task, we use the standard exact match (EM) metric and F1.

4.3 LLMs

We conduct our experiments using several representative LLMs, i.e., (i) ChatGPT (OpenAI, 2022) (we use the gpt-3.5-turbo-1106 version), (ii) Mistral (Jiang et al., 2023a) (the Mistral-7B-Instruct-v0.2 version), and (iii) Llama 3 (Meta, 2024) (the Meta-Llama-3-8B-Instruct version). To ensure the reproducibility of the experiments, the temperature for all experiments is set to 0.

4.4 Baselines

We utilize the following baselines on the utility judgments task and question answering performance based on the utility judgment results:

Single-shot utility judgments. (i) **Vanilla**: Ask LLMs to provide utility judgments based on the instruction directly. (ii) **UJ-ExpA**: Utility judgments and provide explicit answers simultaneously through a single output, which is shown to be effective in Zhang et al. (2024). (iii) **UJ-ImpA**: Utility judgments and provide implicit answers that are necessary to answer the question through a single output.

k-sampling. (Zhang et al., 2024) proposed k-sampling to alleviate the sensitivity of LLMs to input order. Specifically, the k-sampling method randomizes the order of the input passage list k times in addition to the original input and aggregates the $k+1$ utility judgment results through voting. For fair comparison, we use the $k = 5$, more details are on Appendix E.

To evaluate the effectiveness of the proposed ITEM framework in ranking tasks, we are using a verberlized ranking. Therefore, we also employ another verberlized ranking method, i.e., **RankGPT** (Sun et al., 2023) as our main baseline, which uses the LLMs to directly rank input passages based on their relevance to the query.

5 Experimental Results of LLMs

This section will present the performance of each task within our ITEM framework. By default, the

Method	WebAP						TREC DL					
	Listwise			Pointwise			Listwise			Pointwise		
	M	L	C	M	L	C	M	L	C	M	L	C
Vanilla	20.79	21.79	28.43	23.05	25.09	26.85	45.67	49.39	55.19	45.11	47.64	49.84
UJ-ExpA	27.94	26.99	30.50	25.27	29.25	27.44	54.10	52.83	57.49	43.53	53.73	48.09
UJ-ImpA	25.06	26.22	29.89	28.35	25.29	26.32	48.29	48.22	56.18	48.31	50.20	48.83
5-sampling	30.16	28.97	31.49	-	-	-	52.31	52.68	60.49	-	-	-
ITEM-A$_s$ $w.$ ExpA (1)	29.76	27.50	36.89	29.10	31.08	32.02	53.78	53.66	62.52	49.44	52.09	53.61
ITEM-A$_s$ $w.$ ImpA (1)	26.06	25.59	34.97	28.28	30.53	29.34	49.39	53.73	58.11	46.01	53.68	54.61
ITEM-AR$_s$ $w.$ ExpA(1)	35.50	**31.44**	36.58	-	-	-	52.34	48.97	62.00	-	-	-
ITEM-A$_s$ $w.$ ExpA (3)	31.65	29.32	39.57	**30.50**	**32.67**	31.43	54.86	**56.03**	**63.18**	**51.74**	52.46	**55.74**
ITEM-A$_s$ $w.$ ImpA (3)	28.36	26.10	**40.78**	30.13	29.64	**32.54**	52.05	55.14	60.56	46.59	**53.76**	54.90
ITEM-AR$_s$ $w.$ ExpA(3)	**37.06**	29.08	38.58	-	-	-	**56.27**	52.10	61.37	-	-	-

Table 1: The micro-F1 performance (%) of utility judgments with different LLMs on the different datasets (the numbers in parentheses represent m-values). "-" indicates no experiments are performed under the pointwise approach because of that the k-sampling method and our ITEM-AR$_s$ require listwise input. **bold** indicates the best performance. Underline means the best performance among all variants of our ITEM with the same m value. "M", "L", and "C" mean "Mistral", "Llama 3" and "ChatgGPT", respectively.

Method	llama3.1-8B		ChatGPT	
	Listwise	Pointwise	Listwise	Pointwise
Vanilla	43.38	28.55	59.37	35.31
UJ-ExapA	47.07	39.32	66.13	37.17
UJ-ImpA	43.31	38.72	57.40	37.29
k-sampling	49.20	-	71.17	-
ITEM-As-ExpA(1)	49.26	47.52	72.44	54.89
ITEM-As-Imp(1)	47.47	37.98	68.92	43.17
ITEM-ARs-ExpA(1)	50.77	-	74.43	-
ITEM-As-ExpA(3)	49.73	**48.90**	73.55	**55.45**
ITEM-As-Imp(3)	48.03	38.34	69.68	43.58
ITEM-ARs-ExpA(3)	**51.22**	-	**76.34**	-

Table 2: The micro-F1 performance (%) of utility judgments with different LLMs on the GTI-NQ dataset. **Bold** and Underline are defined in Table 1.

pseudo answer is the *explicit answer* in all experiments, if not specified otherwise.

5.1 Utility Judgments Results

Table 1 shows the micro F1 performance on the TREC DL and WebAP datasets using three LLMs. Further, we utilize a better-performing open-source LLM, i.e., Llama-3.1-8B, and a closed LLM, i.e., ChatGPT, to conduct experiments on GTI-NQ, as shown in Table 2. Since ITEM-A$_r$ and ITEM-AR$_r$ have poor F1 performance in utility judgments (refer to Table 12 in Appendix A.4 for details), we restrict our experiments to ITEM-A$_s$ and ITEM-AR$_s$ in this section.

ITEM with a Single Iteration vs. Baselines. All LLMs using our ITEM with a single iteration generally outperform the single-shot utility judgments on three datasets and may even surpass the k-sampling method. For example, ChatGPT on the TREC DL dataset using our ITEM-A$_s$ $w.$ ExpA and ImpA in the listwise approach improve the F1 performance by 8.7% and 3.4% over UJ-ExpA and UJ-ImpA, respectively. Explicit generation of pseudo-answers by LLMs enhances their performance in utility judgment tasks, highlighting the importance of task interaction. Moreover, concurrent execution of answer generation and utility judgment within a single inference cycle yields inferior performance compared to sequential task execution through separate reasoning phases.

ITEM with Multiple Iterations vs. ITEM with Single Iteration. All LLMs using our ITEM-A and ITEM-AR generally demonstrate improved performance with multiple iterations compared to single iterations on all three datasets. For instance, on the WebAP dataset, Mistral, Llama 3, and ChatGPT (using our ITEM-A $w.$ ExpA) improved their F1 scores in the listwise approach by 6.4%, 6.6%, and 7.3%, respectively, after multiple iterations. Moreover, our method achieves state-of-the-art performance compared to all baselines by leveraging the iterative framework. The performance improvement from multiple iterations underscores the significance of iterative interaction and further supports Schutz's interactive framework. However, in some specific cases, multiple iterations may not outperform single iterations, likely due to the unpredictable nature of zero-shot settings. ChatGPT outperforms other LLMs on all datasets using both input approaches.

ITEM-A$_s$ vs. ITEM-AR$_s$. In our utility-emphasized iterative RAG framework, ITEM-A$_s$ and ITEM-AR$_s$ are the two major methods we propose. From Table 1, we find that ITEM-AR$_s$ works better than ITEM-A$_s$ most of the time for complex questions (WebAP, all the questions are non-factoid) and the complex candidate passage list (GTI-NQ, containing different types of passage), indicating complicated question or passage list need more components in the loop. For TREC

Method	Mistral		Llama 3		ChatGPT	
	TREC	WebAP	TREC	WebAP	TREC	WebAP
D	58.69	21.89	58.69	21.89	58.69	21.89
RankGPT	69.81	29.34	75.61	41.73	80.56	42.49
ITEM-A$_r$ (1)	70.57	37.11	73.95	40.89	80.79	<u>50.30</u>
ITEM-AR$_s$ (1)	<u>71.29</u>	<u>37.48</u>	<u>77.22</u>	<u>43.80</u>	<u>81.38</u>	48.42
ITEM-A$_r$ (3)	**74.27**	43.80	**77.34**	**45.88**	**83.12**	**51.61**
ITEM-AR$_s$ (3)	73.24	**45.45**	74.80	44.87	82.89	48.80

Table 3: The NDCG@5 performance (%) of the ranking using different LLMs on the different datasets. **Bold** and <u>Underline</u> are defined in Table 1.

Method	Llama 3		ChatGPT	
	@5	@10	@5	@10
D	29.46	45.26	29.46	45.26
RankGPT	71.50	74.05	77.27	78.64
ITEM-Ar(m=1)	74.36	76.91	<u>85.99</u>	<u>87.26</u>
ITEM-ARs(m=1)	<u>75.46</u>	<u>77.75</u>	84.54	85.14
ITEM-Ar(m=3)	75.95	78.18	**87.48**	**88.47**
ITEM-ARs(m=3)	**76.38**	**78.56**	85.95	86.39

Table 4: The NDCG performance (%) of the ranking using different LLMs on the GTI-NQ dataset. **Bold** and <u>Underline</u> are defined in Table 1.

DL, which contains factoid questions, we find that ITEM-AR$_s$ is worse than ITEM-A$_s$ most times on TREC DL. This is reasonable since factoid questions are relatively easier to answer and may not need more components involved in the iteration.

Listwise vs. Pointwise. The general performance of utility judgments for LLMs is better with the listwise approach than with the pointwise approach. The primary rationale lies in the listwise approach exposes the LLM to broader contextual information, thereby facilitating more effective interaction during the LLMs in judging the passages' utility.

5.2 Ranking Performance

We also assess whether the ranking performance has been improved within ITEM on retrieval datasets (Table 3) and utility judgments benchmark (Table 4). In terms of ranking performance, we consider two rankings: relevance ranking (ITEM-AR$_s$) and utility ranking (ITEM-A$_r$). We can observe that: (i) Our ITEM with single iteration significantly improves the ranking of topical relevance performance compared to the RankGPT. For instance, relevance ranking outperforms RankGPT in NDCG@5 by 2.1% on the TREC dataset. The performance improvement may stem from the interaction between tasks. (ii) After iterations, relevance and utility ranking performance have been improved on all datasets and all LLMs. The ranking benefits from our dynamic iterative framework, confirming Schutz's theory of dynamic iterative interaction. (iii) From Table 3&4, we can find that

References	Mistral		Llama 3		ChatGPT	
	EM	F1	EM	F1	EM	F1
Golden	46.09	62.59	64.45	76.64	66.40	76.86
D	31.58	47.69	**50.96**	62.01	46.54	57.00
Vanilla	31.16	47.43	49.09	60.56	48.52	58.64
UJ-ExpA	32.76	48.46	49.63	61.10	47.72	58.01
UJ-ImpA	30.67	46.83	48.88	60.26	49.01	59.30
5-sampling	33.24	48.84	48.72	60.71	48.90	58.97
ITEM-A$_s$ (1)	32.98	49.00	50.16	<u>61.88</u>	49.38	<u>59.78</u>
ITEM-AR$_s$ (1)	<u>33.30</u>	<u>49.26</u>	<u>50.27</u>	61.69	<u>49.52</u>	59.64
ITEM-A$_s$ (3)	<u>**33.73**</u>	<u>**49.63**</u>	<u>50.27</u>	<u>**62.09**</u>	<u>**49.69**</u>	<u>**60.18**</u>
ITEM-AR$_s$ (3)	33.40	49.27	49.36	60.97	49.06	59.67

Table 5: The answer generation performance (%) of all LLMs on the NQ dataset using reference passages collected from different methods. **Bold** means the best performance except for the answer generation with golden evidence. <u>Underline</u> is defined in Table 1.

ITEM-AR$_s$ is generally better than ITEM-A$_r$ when $m = 1$. However, when $m = 3$, it may have the opposite performance. The possible reason is that when m is small, the answer is very good, and utility is more dependent on the answer than relevance. However, as iterations occur, the quality of the answer is better, and utility performance is gradually improved, while relevance does not have as obvious improvement effect as utility in iterations. This indicates that reranking based on utility performs better than including more components in the loop, but still using relevance as the ranking criterion. These findings further confirm the importance of the concept of utility in RAG.

5.3 Results of Answer Generation

In the answer generation task, the results of utility judgments are fed to LLMs for answer generation. We use the factoid QA dataset (i.e., NQ) for answer generation evaluation, as shown in Table 5. From Table 1&2, we find that the listwise approach generally outperforms the pointwise approach for utility judgments. Consequently, our answer generation experiments utilize only the listwise utility judgments. The following observations can be made from Table 5: (i) ITEM outperforms baselines on all metrics on all LLMs (except for the EM score of Llama 3), indicating that ITEM can help the LLMs to find better evidence for generating answers. (ii) Similar to Table 1&2, when the $m = 1$, ITEM-AR$_s$ performs better than ITEM-A$_s$, which shows the importance of relevance reranking in ITEM. However, as the number of iterations increases, ITEM-A$_s$ performs better than ITEM-AR$_s$. We are keen to discern the optimal choice for different tasks: 1) More components and more iterations are not always needed, especially for simpler tasks;

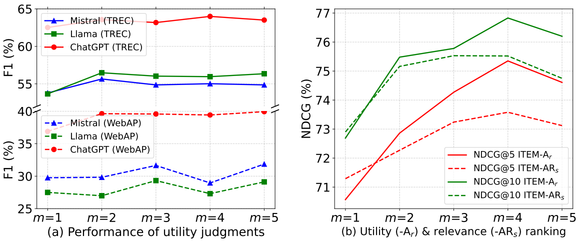

Figure 7: (a): utility judgments performance (%) in terms of m values in ITEM-A$_s$. (b): relevance ranking (ITEM-AR$_s$) and utility ranking (ITEM-A$_r$) performance (%) of Mistral on the TREC DL dataset.

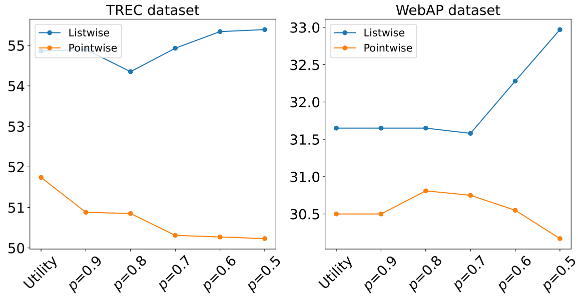

Figure 8: The utility judgments F1 performance (%) of Mistral in different iteration stop conditions (m=3) under ITEM-A$_s$.

2) Fewer iterations with numerous components, or increased iterations with few components.

6 Further Analyses

Iteration Rounds. Figure 7 shows the performance of (a): utility judgments under ITEM-A$_s$ and (b): ranking with varying maximum iteration rounds m. We observe the following: 1) Varying the value of m affects the performance of utility judgments, ranking. 2) Based on empirical observations balancing the cost and performance, the m was operationally configured with distinct values for different question types on utility judgments (m=3 in our paper on all experiments for fair comparison): m=2 for factoid questions, whereas m=3 was implemented for non-factoid questions in practical applications. 3) Utility ranking generally outperforms relevance ranking, which confirms the effectiveness of utility in the ranking task.

Iteration Stop Conditions. In addition to utility judgments, we also consider the pseudo answer generation performance in ITEM as a stopping condition. Specifically, we calculate the ROUGE-L score (Lin, 2004) of the answer in two iterations and stop the iteration if the ROUGE-L of a_t and a_{t-1} is greater than p. The utility judgments performance of different iteration stop conditions are shown in Figure 8. The results show that using different stopping conditions affects the performance of utility judgments. However, using the answer as a stopping condition, different LLMs on different datasets may need to look for different p, which is not very flexible.

7 Inference Efficiency

Table 14 in Appendix G shows the inference efficiency analysis of our ITEM. Our iteration framework surpasses k-sampling in computational efficiency during inference. The proposed approach demonstrates potential for large-scale retrieval data annotation, where ITEM-A$_s$ emerges as an optimal solution by simultaneously enhancing both performance and operational efficiency, thereby facilitating utility annotation for retrieval systems.

8 Case Study

Figure 9 in Appendix B presents two cases from the TREC DL dataset using Mistral under ITEM-A$_s$. For the first question in Figure 9, the first pseudo-answer, though relatively correct, includes irrelevant information, leading to a misjudgment of "Passage-2" as "utility". Based on the results of the first round of utility judgments, the second round of the pseudo-answer is more accurate and free of irrelevant content. Consequently, all three passages in the second round of utility judgments have utility in answering the question. For the second question in Table 9, the first pseudo-answer is correct, but two passages without utility are judged as "utility". The second pseudo-answer, with slight rewording, results in all selected passages being correct.

9 Conclusion

In this paper, we propose an Iterative utiliTy judgmEnt fraMework (ITEM) to enhance the utility judgment and QA performance of LLMs by interactions between the steps, inspired by Schutz's philosophical discussion of relevance. This is a unified framework of iterative RAG with an emphasis on utility. Our framework achieves state-of-the-art performance in zero-shot scenarios, outperforming previous methods in utility judgments, ranking of topical relevance, and answer generation tasks, indicating that the cognitive process of LLMs on a specific topic can also be improved by a similar process. Our experiments also highlight the significance of dynamic interaction in achieving high performance and stability. Future directions include developing better fine-tuning strategies for utility judgments and creating end-to-end solutions for RAG.

Limitations

There are two primary limitations that should be acknowledged: (i) Our methods are applied in zero-shot scenarios without any training. The zero-shot approach itself does not enhance the LLMs's inherent capability in utility judgments but rather employs strategies to improve performance on utility judgment tasks. Future research should explore designing more effective training methods, e.g., utilizing our iterative framework with self-evolution techniques (Singh et al., 2023), to genuinely enhance the LLMs's ability in utility judgments through training. (ii) The number of candidate passages in the search scenario is much larger than 20. The number of search results we assumed is too small. We need to continue to study utility judgments in large-scale scenarios in the future.

10 Ethics Statement

Our research does not rely on personally identifiable information. All datasets and models used in our paper are publicly available and have been widely adopted by researchers. We firmly believe in the principles of open research and the scientific value of reproducibility. To this end, we have made all data, and code associated with our paper publicly available on GitHub. This transparency not only facilitates the verification of our findings by the community but also encourages the application of our methods in other contexts.

References

Keping Bi, Qingyao Ai, and W Bruce Croft. 2019. Iterative relevance feedback for answer passage retrieval with passage-level semantic match. In Advances in Information Retrieval: 41st European Conference on IR Research, ECIR 2019, Cologne, Germany, April 14–18, 2019, Proceedings, Part I 41, pages 558–572. Springer.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, and 1 others. 2022. Improving language models by retrieving from trillions of tokens. In International conference on machine learning, pages 2206–2240. PMLR.

Harry W Bruce. 1994. A cognitive view of the situational dynamism of user-centered relevance estimation. Journal of the American Society for Information Science, 45(3):142–148.

Daniel Cohen, Liu Yang, and W Bruce Croft. 2018. Wikipassageqa: A benchmark collection for research on non-factoid answer passage retrieval. In The 41st international ACM SIGIR conference on research & development in information retrieval, pages 1165–1168.

William S Cooper. 1971. A definition of relevance for information retrieval. Information storage and retrieval, 7(1):19–37.

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

Michael Glass, Gaetano Rossiello, Md Faisal Mahbub Chowdhury, Ankita Naik, Pengshan Cai, and Alfio Gliozzo. 2022. Re2G: Retrieve, rerank, generate. In Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 2701–2715, Seattle, United States. Association for Computational Linguistics.

William Goffman and Vaun A Newill. 1964. Methodology for test and evaluation of information retrieval systems. Center for Documentation and Communication Research, School of Library ....

Helia Hashemi, Mohammad Aliannejadi, Hamed Zamani, and W Bruce Croft. 2020. Antique: A non-factoid question answering benchmark. In Advances in Information Retrieval: 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14–17, 2020, Proceedings, Part II 42, pages 166–173. Springer.

Helia Hashemi, Hamed Zamani, and W Bruce Croft. 2019. Performance prediction for non-factoid question answering. In Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval, pages 55–58.

Donald J Hillman. 1964. The notion of relevance (i). American Documentation, 15(1):26–34.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2023. Atlas: Few-shot learning with retrieval augmented language models. Journal of Machine Learning Research, 24(251):1–43.

Kalervo Järvelin and Jaana Kekäläinen. 2017. Ir evaluation methods for retrieving highly relevant documents. In ACM SIGIR Forum, volume 51, pages 243–250. ACM New York, NY, USA.

Kaixin Ji, Danula Hettiachchi, Flora D. Salim, Falk Scholer, and Damiano Spina. 2024. Characterizing information seeking processes with multiple physiological signals. SIGIR.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, and 1 others. 2023a. Mistral 7b. arXiv preprint arXiv:2310.06825.

Zhengbao Jiang, Frank Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023b. Active retrieval augmented generation. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pages 7969–7992, Singapore. Association for Computational Linguistics.

Mostafa Keikha, Jae Hyun Park, and W Bruce Croft. 2014a. Evaluating answer passages using summarization measures. In Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval, pages 963–966.

Mostafa Keikha, Jae Hyun Park, W Bruce Croft, and Mark Sanderson. 2014b. Retrieving passages and finding answers. In Proceedings of the 19th Australasian Document Computing Symposium, pages 81–84.

DA Kemp. 1974. Relevance, pertinence and information system development. Information Storage and Retrieval, 10(2):37–47.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Generalization through memorization: Nearest neighbor language models. ICLR.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, and 1 others. 2019. Natural questions: a benchmark for question answering research. Transactions of the Association for Computational Linguistics, 7:453–466.

F Wilfrid Lancaster. 1968. Information retrieval systems: Characteristics, testing, and evaluation. (No Title).

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. Advances in Neural Information Processing Systems, 33:9459–9474.

Xiaonan Li, Changtai Zhu, Linyang Li, Zhangyue Yin, Tianxiang Sun, and Xipeng Qiu. 2023. Llatrieval: Llm-verified retrieval for verifiable generation. arXiv preprint arXiv:2311.07838.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In Text summarization branches out, pages 74–81.

Meta. 2024. Welcome llama 3 - meta's new open llm.

Stefano Mizzaro. 1997. Relevance: The whole history. Journal of the American society for information science, 48(9):810–832.

Stefano Mizzaro. 1998. How many relevances in information retrieval? Interacting with computers, 10(3):303–320.

OpenAI. 2022. Introducing chatgpt.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. Transactions of the Association for Computational Linguistics, 11:1316–1331.

Ruiyang Ren, Yingqi Qu, Jing Liu, Wayne Xin Zhao, Qiaoqiao She, Hua Wu, Haifeng Wang, and Ji-Rong Wen. 2021. Rocketqav2: A joint training method for dense passage retrieval and passage re-ranking. EMNLP.

Stephen Robertson, Hugo Zaragoza, and 1 others. 2009. The probabilistic relevance framework: Bm25 and beyond. Foundations and Trends® in Information Retrieval, 3(4):333–389.

Tefko Saracevic. 1975. Relevance: A review of and a framework for the thinking on the notion in information science. Journal of the American Society for information science, 26(6):321–343.

Tefko Saracevic. 1996. Relevance reconsidered. In Proceedings of the second conference on conceptions of library and information science (CoLIS 2), pages 201–218.

Tefko Saracevic, Paul Kantor, Alice Y Chamis, and Donna Trivison. 1988. A study of information seeking and retrieving. i. background and methodology. Journal of the American Society for Information science, 39(3):161–176.

Linda Schamber and Michael Eisenberg. 1988. Relevance: The search for a definition.

Linda Schamber, Michael B Eisenberg, and Michael S Nilan. 1990. A re-examination of relevance: toward a dynamic, situational definition. Information processing & management, 26(6):755–776.

Alfred Schutz. 1970. Reflections on the Problem of Relevance. Greenwood Press, Westport, Conn.

Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. 2023. Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy. In Findings of the Association for Computational Linguistics: EMNLP 2023, pages 9248–9274, Singapore. Association for Computational Linguistics.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2023. Replug: Retrieval-augmented black-box language models. arXiv preprint arXiv:2301.12652.

Avi Singh, John D Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Peter J Liu, James Harrison, Jaehoon Lee, Kelvin Xu, Aaron Parisi, and 1 others. 2023. Beyond human data: Scaling self-training for problem-solving with language models. arXiv preprint arXiv:2312.06585.

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

Weiwei Sun, Lingyong Yan, Xinyu Ma, Pengjie Ren, Dawei Yin, and Zhaochun Ren. 2023. Is chatgpt good at search? investigating large language models as re-ranking agent. EMNLP.

Don R Swanson. 1986. Subjective versus objective relevance in bibliographic retrieval systems. The library quarterly, 56(4):389–398.

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), ACL 2023, Toronto, Canada, July 9-14, 2023, pages 10014–10037. Association for Computational Linguistics.

Brian C Vickery. 1959. The structure of information retrieval systems. In Proceedings of the International Conference on Scientific Information, volume 2, pages 1275–1290.

Ellen M Voorhees and 1 others. 2003. Overview of the trec 2003 robust retrieval track. In Trec, pages 69–77.

Jian Xie, Kai Zhang, Jiangjie Chen, Renze Lou, and Yu Su. 2023. Adaptive chameleon or stubborn sloth: Revealing the behavior of large language models in knowledge conflicts. In The Twelfth International Conference on Learning Representations.

Liu Yang, Qingyao Ai, Damiano Spina, Ruey-Cheng Chen, Liang Pang, W Bruce Croft, Jiafeng Guo, and Falk Scholer. 2016. Beyond factoid qa: effective methods for non-factoid answer sentence retrieval. In Advances in Information Retrieval: 38th European Conference on IR Research, ECIR 2016, Padua, Italy, March 20–23, 2016. Proceedings 38, pages 115–128. Springer.

Hengran Zhang, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Yixing Fan, and Xueqi Cheng. 2023. From relevance to utility: Evidence retrieval with feedback for fact verification. In Findings of the Association for Computational Linguistics: EMNLP 2023, pages 6373–6384, Singapore. Association for Computational Linguistics.

Hengran Zhang, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Yixing Fan, and Xueqi Cheng. 2024. Are large language models good at utility judgments? Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24).

Xinran Zhao, Tong Chen, Sihao Chen, Hongming Zhang, and Tongshuang Wu. 2024. Beyond relevance: Evaluate and improve retrievers on perspective awareness. arXiv preprint arXiv:2405.02714.

Yujia Zhou, Zheng Liu, Jiajie Jin, Jian-Yun Nie, and Zhicheng Dou. 2024. Metacognitive retrieval-augmented large language models. In Proceedings of the ACM on Web Conference 2024, WWW '24, page 1453–1463, New York, NY, USA. Association for Computing Machinery.

A Experiment Details

A.1 Effect of Iteration Numbers

The precision, recall, and F1 performance of different LLMs on different datasets with different iteration numbers is shown in Table 6, Table 7, Table 8, and Table 9.

A.2 Quality of Utility Judgments

The relevance labels of TREC DL are of a four-point scale, and we consider the highest level as having utility. To see the utility judgment performance when we consider lower grades to have utility, we measure the precision of utility judgments of Mistral on TREC DL when passages of different grades are treated as positive in Table 10. We can see that almost 70% of the results of positive utility judgments are highly relevant to the question.

A.3 k values in ITEM-A$_r$

Different ranking performance of k values in ITEM-A$_r$ is shown in Table 11. Considering the performance of utility ranking and utility judgments, we set k=5.

A.4 ITEM-AR$_r$

We evaluate two ranking performances of ITEM-AR$_r$ during the same loop, with the experimental results shown in Table 12. We find that under the ITEM-AR$_r$ framework, relevance ranking and utility ranking are both improved, and utility ranking performance is generally better than relevance ranking. However, as seen in Table 3 and Table 12, performing ranking twice in the same iteration may not yield better ranking results than performing utility ranking once in the iteration.

Method	TREC						WebAP					
	listwise			pointwise			listwise			pointwise		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Vanilla	36.82	60.13	45.67	29.92	**91.61**	45.11	13.07	50.83	20.79	13.30	86.29	23.05
UJ-ExpA	48.51	61.15	54.10	28.12	96.27	43.53	18.83	54.16	27.94	14.65	**91.82**	25.27
UJ-ImpA	40.16	60.53	48.29	33.95	83.76	48.31	16.46	52.45	25.06	17.56	73.55	28.35
5-sampling	46.64	59.56	52.31	-	-	-	20.61	56.22	30.16	-	-	-
ITEM-A$_s$ w. ExpA (m=1)	48.07	61.04	53.78	34.21	89.11	49.44	20.57	53.81	29.76	17.86	78.41	29.10
ITEM-A$_s$ w. ExpA (m=2)	50.58	61.86	55.65	35.87	88.73	51.09	21.11	50.85	29.83	18.27	82.00	29.88
ITEM-A$_s$ w. ExpA (m=3)	50.61	59.88	54.86	36.23	90.46	51.74	23.57	48.14	31.65	18.73	81.96	30.50
ITEM-A$_s$ w. ExpA (m=4)	50.01	61.15	55.02	**36.41**	90.36	**51.90**	21.44	44.62	28.96	**19.19**	80.59	**31.00**
ITEM-A$_s$ w. ExpA (m=5)	50.61	59.88	54.86	36.14	90.46	51.65	24.07	47.09	31.86	19.17	78.94	30.85
ITEM-A$_s$ w. ImpA (m=1)	39.97	64.62	49.39	30.98	89.38	46.01	16.88	57.13	26.06	17.10	81.65	28.28
ITEM-A$_s$ w. ImpA (m=2)	43.14	61.52	50.72	30.90	87.00	45.60	19.41	54.82	28.67	18.88	78.06	30.40
ITEM-A$_s$ w. ImpA (m=3)	44.43	62.82	52.05	31.68	87.99	46.59	19.21	54.20	28.36	18.69	77.77	30.13
ITEM-A$_s$ w. ImpA (m=4)	44.72	61.29	51.71	31.66	87.40	46.49	17.44	47.11	25.46	18.95	78.06	30.50
ITEM-A$_s$ w. ImpA (m=5)	44.63	60.98	51.54	31.80	89.32	46.91	18.98	48.88	27.35	19.05	76.69	30.52
ITEM-AR$_s$ (m=1)	43.65	65.34	52.34	-	-	-	25.04	**60.99**	35.50	-	-	-
ITEM-AR$_s$ (m=2)	45.10	65.46	53.40	-	-	-	24.42	51.97	33.23	-	-	-
ITEM-AR$_s$ (m=3)	49.07	**65.96**	56.27	-	-	-	**27.70**	55.95	**37.06**	-	-	-
ITEM-AR$_s$ (m=4)	50.96	62.32	56.07	-	-	-	23.77	53.40	32.90	-	-	-
ITEM-AR$_s$ (m=5)	**53.01**	63.60	**57.82**	-	-	-	25.85	47.56	33.50	-	-	-

Table 6: The utility judgments performance (%) of Mistral on retrieval datasets (Numbers in parentheses represent m-values). Numbers in bold indicate the best performance.

Method	TREC						WebAP					
	listwise			pointwise			listwise			pointwise		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Vanilla	34.67	**85.80**	49.39	31.42	**98.47**	47.64	12.69	77.15	21.79	14.65	**87.36**	25.09
UJ-ExpA	39.21	80.98	52.83	38.27	90.15	53.73	16.32	77.92	26.99	18.04	77.15	29.25
UJ-ImpA	33.92	83.36	48.22	38.68	71.47	50.20	15.57	**82.79**	26.22	17.22	47.61	25.29
5-sampling	39.04	80.98	52.68	-	-	-	17.52	83.49	28.97	-	-	-
ITEM-A$_s$ w. ExpA (m=1)	39.68	82.88	53.66	37.58	84.84	52.09	17.54	63.67	27.50	19.65	74.31	31.08
ITEM-A$_s$ w. ExpA (m=2)	**42.35**	84.77	**56.48**	38.25	84.58	52.68	17.39	60.25	26.99	20.23	73.01	31.68
ITEM-A$_s$ w. ExpA (m=3)	42.00	84.15	56.03	37.84	85.50	52.46	19.12	62.87	29.32	**20.91**	74.63	32.67
ITEM-A$_s$ w. ExpA (m=4)	41.85	84.41	55.96	38.12	85.16	52.67	17.53	61.85	27.31	20.44	73.83	32.02
ITEM-A$_s$ w. ExpA (m=5)	42.36	84.15	56.35	37.35	84.69	51.84	18.94	62.87	29.12	20.88	75.45	**32.71**
ITEM-A$_s$ w. ImpA (m=1)	39.63	83.42	53.73	39.70	82.87	53.68	15.48	73.66	25.59	20.04	64.06	30.53
ITEM-A$_s$ w. ImpA (m=2)	38.75	85.63	53.35	38.15	82.36	52.14	15.50	76.47	25.77	18.54	62.69	28.62
ITEM-A$_s$ w. ImpA (m=3)	40.84	84.86	55.14	40.58	79.64	53.76	15.99	70.99	26.10	19.54	61.32	29.64
ITEM-A$_s$ w. ImpA (m=4)	38.88	82.74	52.90	39.34	81.74	53.12	15.03	74.41	25.01	19.72	59.95	29.68
ITEM-A$_s$ w. ImpA (m=5)	41.26	84.61	55.47	**40.92**	82.14	**54.63**	15.49	68.93	25.29	19.84	57.21	29.46
ITEM-AR$_s$ (m=1)	34.53	84.17	48.97	-	-	-	**20.05**	72.88	**31.44**	-	-	-
ITEM-AR$_s$ (m=2)	36.27	83.19	50.51	-	-	-	15.92	79.01	26.50	-	-	-
ITEM-AR$_s$ (m=3)	38.04	82.68	52.10	-	-	-	17.93	76.87	29.08	-	-	-
ITEM-AR$_s$ (m=4)	37.28	83.70	51.58	-	-	-	16.60	78.81	27.42	-	-	-
ITEM-AR$_s$ (m=5)	40.25	81.37	53.86	-	-	-	17.04	74.83	27.75	-	-	-

Table 7: The utility judgments performance (%) of Llama 3 on retrieval datasets (Numbers in parentheses represent m-values). Numbers in bold indicate the best performance.

Method	TREC						WebAP					
	listwise			pointwise			listwise			pointwise		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Vanilla	42.13	**79.98**	55.19	33.86	94.40	49.84	17.13	**83.45**	28.43	15.80	**89.42**	26.85
UJ-ExpA	45.74	77.36	57.49	32.06	**96.19**	48.09	19.51	69.86	30.50	16.23	88.74	27.44
UJ-ImpA	44.19	77.11	56.18	33.45	90.36	48.83	18.37	80.14	29.89	15.58	84.51	26.32
5-sampling	50.78	74.77	60.49	-	-	-	20.70	65.83	31.49	-	-	-
ITEM-A$_s$ w. ExpA (m=1)	55.55	71.48	62.52	37.83	91.94	53.61	26.74	59.45	36.89	19.73	84.95	32.02
ITEM-A$_s$ w. ExpA (m=2)	57.95	70.40	63.57	40.74	93.04	56.67	29.43	60.58	39.62	19.62	78.62	31.40
ITEM-A$_s$ w. ExpA (m=3)	58.36	68.88	63.18	40.00	91.88	55.74	29.30	60.91	39.57	19.80	76.20	31.43
ITEM-A$_s$ w. ExpA (m=4)	**58.48**	70.67	**64.00**	40.25	93.38	56.25	29.11	61.03	39.42	20.48	79.63	32.58
ITEM-A$_s$ w. ExpA (m=5)	58.34	69.69	63.51	39.29	92.16	55.09	29.76	60.68	**39.93**	20.58	80.42	**32.77**
ITEM-A$_s$ w. ImpA (m=1)	54.36	65.08	59.24	40.89	82.20	54.61	24.79	64.37	35.80	18.78	67.00	29.34
ITEM-A$_s$ w. ImpA (m=2)	55.88	63.11	59.27	43.32	83.13	**56.96**	27.68	62.03	38.28	20.70	70.54	32.00
ITEM-A$_s$ w. ImpA (m=3)	57.33	64.17	60.56	41.66	80.48	54.90	**30.01**	63.60	40.78	21.51	66.77	32.54
ITEM-A$_s$ w. ImpA (m=4)	55.98	62.24	58.95	**42.34**	80.65	55.53	28.43	60.11	38.60	20.60	65.63	31.36
ITEM-A$_s$ w. ImpA (m=5)	56.63	62.19	59.28	41.49	83.57	55.45	29.05	60.66	39.29	**21.51**	68.03	32.68
ITEM-AR$_s$ (m=1)	51.94	76.90	62.00	-	-	-	25.32	65.84	36.58	-	-	-
ITEM-AR$_s$ (m=2)	53.77	76.19	63.05	-	-	-	25.55	59.26	35.70	-	-	-
ITEM-AR$_s$ (m=3)	52.41	74.04	61.37	-	-	-	27.61	63.96	38.58	-	-	-
ITEM-AR$_s$ (m=4)	52.75	73.78	61.52	-	-	-	28.84	61.85	39.34	-	-	-
ITEM-AR$_s$ (m=5)	52.77	76.28	62.39	-	-	-	28.76	62.54	39.40	-	-	-

Table 8: The utility judgments performance of ChatGPT on retrieval datasets (Numbers in parentheses represent m-values). Numbers in bold indicate the best performance.

References of RAG	Mistral		Llama 3		ChatGPT	
	EM	F1	EM	F1	EM	F1
Golden Evidence	46.09	62.59	64.45	76.64	66.40	76.86
RocketQAv2	31.58	47.69	**50.96**	62.01	46.54	57.00
Vanilla	31.16	47.43	49.09	60.56	48.52	58.64
UJ-ExpA	32.76	48.46	49.63	61.10	47.72	58.01
UJ-ImpA	30.67	46.83	48.88	60.26	49.01	59.30
5-sampling	33.24	48.84	48.72	60.71	48.90	58.97
ITEM-A$_s$ w. ExpA (m=1)	32.98	49.00	50.16	61.88	49.38	59.78
ITEM-A$_s$ w. ExpA (m=2)	**34.31**	**50.08**	50.48	**62.32**	49.22	59.99
ITEM-A$_s$ w. ExpA (m=3)	33.73	49.63	50.27	62.09	**49.69**	**60.18**
ITEM-A$_s$ w. ExpA (m=4)	34.21	50.07	50.43	62.20	-	-
ITEM-A$_s$ w. ExpA (m=5)	33.78	49.63	50.27	62.07	-	-
ITEM-A$_s$ w. ImpA (m=1)	32.17	48.51	50.37	61.89	48.75	58.99
ITEM-A$_s$ w. ImpA (m=2)	32.49	48.67	49.63	61.16	49.11	59.14
ITEM-A$_s$ w. ImpA (m=3)	32.39	48.47	49.68	61.48	48.69	58.94
ITEM-A$_s$ w. ImpA (m=4)	32.71	48.84	49.41	61.03	-	-
ITEM-A$_s$ w. ImpA (m=5)	32.33	48.44	49.73	61.42	-	-
ITEM-AR$_s$ (m=1)	33.30	49.26	50.27	61.69	49.52	59.64
ITEM-AR$_s$ (m=2)	33.57	49.16	50.70	61.92	49.01	59.75
ITEM-AR$_s$ (m=3)	33.40	49.27	49.36	60.97	49.06	59.67
ITEM-AR$_s$ (m=4)	33.46	49.24	49.84	61.54	-	-
ITEM-AR$_s$ (m=5)	33.89	49.58	49.20	60.84	-	-

Table 9: The answer generation performance (%) of all LLMs in the listwise approach. Numbers in bold indicate the best performance except the answer performance using golden evidence. Due to the high cost of using ChatGPT, we only tested with m=1,2,3 on ChatGPT.

m	label≥1	label≥2	label≥3
m=1	82.08	68.34	48.07
m=2	83.86	69.53	50.58
m=3	84.23	**71.06**	**50.61**
m=4	**84.63**	70.18	50.01
m=5	84.52	70.69	50.61

Table 10: The precision scores (%) of utility judgments using Mistral in different m (iteration) values. "label" is the manual annotation in the original dataset, i.e., [3]: Perfectly relevant; [2]: Highly relevant; [1]: Related; [0]: Irrelevant.

B Case Study

We show two cases on the TREC dataset in Table 9.

C Datasets and Evaluation

Detailed statistics of the experimental datasets are shown in Table 13. We use the *trec_eval* official tool for evaluation of NDCG.

D Answer Passage Retrieval

Non-factoid questions are usually expected longer answers, such as sentence-level or passages-level (Keikha et al., 2014a; Yang et al., 2016; Keikha et al., 2014b). Yang et al. (2016) developed an annotated dataset for answer passage retrieval called WebAP, which has an average of 76.4 qrels per query. Cohen et al. (2018) and Hashemi et al. (2020) introduced the WikiPassageQA dataset and ANTIQUE dataset for answer passage retrieval, respectively. Compared to the WebAP dataset, WikiPassageQA and ANTIQUE have incomplete annotations, with an average of 1.7 qrels and 32.9 qrels per query (Hashemi et al., 2019, 2020). Bi et al. (2019) created the PsgRobust dataset for answer passage retrieval, which is built on the TREC Robust collection (Voorhees et al., 2003) following a similar approach to WebAP but without manual annotation.

E k-sampling

The output of k-sampling each time contains explicit answers and utility judgments. If the question length is l_q, the total length of the input passages is l_p, and the average length of a single passage is l_{avg}, then the k-sampling input cost is $(k+1) \times (l_q + l_p)$. If the average length of the output explicit answer is l_a, and the average length of the output utility judgments is l_u, then the k-sampling output cost is $(k+1) \times (l_a + l_u)$. Taking ITEM-As as an example, with a maximum of three iterations, the maximum input cost for utility judgments is $3 \times (l_q + l_p)$. For answer generation, the longest input is $l_q + l_p$ and the shortest is $l_q + l_{\text{avg}}$. Therefore, the maximum input cost for ITEM-As is $6 \times (l_q + l_p)$ and the minimum is $4 \times (l_q + l_p) + 2 \times (l_q + l_{\text{avg}})$. The maximum output cost is $3 \times (l_a + l_u)$. Therefore, in order to ensure fairness in the calculation of large language model parameters, we choose k=5.

F Retrievers

We use two representative retrievers to gather candidate passages in $\mathcal{D}$ for utility judgments. Following with previous works (Zhang et al., 2024; Sun et al., 2023), we employ RocketQAv2 (Ren et al., 2021) and BM25 (Robertson et al., 2009) for the NQ dataset and retrieval datasets(i.e., TREC DL and WebAP datasets), respectively. Based on the retrieval results to build the $\mathcal{D}$ we have two settings: (i) For TREC DL and WebAP datasets, we select the top-20 BM25 retrieval results. If these do not include passages with utility, we replaced the last one with a utility-annotated passage. (ii) For the NQ dataset, we use the top-10 dense retrieval results to form the candidate list $\mathcal{D}$, following the GTU setting of Zhang et al. (2024).

G Inference Efficiency

Table 14 shows more analysis of Mistral on the TREC DL dataset using the listwise input form. The temperature of the LLMs is set to 0 during the generation process, and we used a single run. Due to the iterations, the average input token length of our methods is relatively large. The cost of ITEM-A$_s$ is about 0.5 times that of the 5-sampling, and ITEM-AR$_s$ is about 1.5 times the cost of 5-sampling. Our framework provides a way of automatically obtaining high-quality labeled data for each task in RAG. These annotations can be used to train regular RAG models. The cost should be worth it (much better than k-sampling).

H Instruction Details

H.1 Instruction of Listwise and Pointwise Approaches

For the prompts of the NQ dataset using ChatGPT, we follow the setting of Zhang et al. (2024), otherwise, we use the following prompts. Following Sun et al. (2023), we input N passages using the form of multiple rounds of dialogue in the listwise approach. The prompts we used in our experiments are shown in Figure 10 and Figure 11.

k, m	Ranking					Utility judgments		
	N@1	N@3	N@5	N@10	N@20	P	R	F1
k=1, m=1	72.76	71.27	70.57	72.69	84.08	53.66	24.09	33.25
k=1, m=2	76.02	71.54	71.38	73.66	84.78	58.54	28.73	38.54
k=1, m=3	77.24	72.83	71.83	73.87	85.20	**59.76**	28.84	38.90
k=1, m=4	77.24	73.04	71.91	73.90	85.25	59.76	28.84	38.90
k=1, m=5	76.02	72.11	71.42	73.45	84.98	58.54	28.71	38.53
k=5, m=1	72.76	71.27	70.57	72.69	84.08	33.17	57.31	42.02
k=5, m=2	78.46	73.74	72.86	75.48	86.09	32.93	58.37	42.10
k=5, m=3	79.27	75.00	74.27	75.78	86.80	34.15	62.57	44.18
k=5, m=4	79.67	75.92	**75.35**	**76.83**	**87.23**	35.12	61.40	**44.68**
k=5, m=5	79.67	75.32	74.61	76.20	86.82	34.63	61.25	44.25
k=10, m=1	72.76	71.27	70.57	72.69	84.08	22.56	68.03	33.88
k=10, m=2	78.05	72.64	72.90	75.48	85.74	23.66	75.47	36.02
k=10, m=3	**80.89**	**76.58**	74.54	76.30	86.94	23.78	**75.65**	36.19
k=10, m=4	78.05	74.70	72.85	75.12	85.72	24.51	74.17	36.85
k=10, m=5	79.67	75.60	74.84	76.54	86.88	23.66	74.42	35.90

Table 11: The utility ranking performance and utility judgments performance of Mistral on TREC DL dataset in ITEM-A$_r$. "N@k" means "NDCG@k". Numbers in bold indicate the best performance.

m	NDCG@5	NDCG@10	NDCG@20	Utility-F1
1	71.29 / 72.77	72.90 / 74.96	84.56 / 85.75	43.13
2	**72.54** / 70.99	74.81 / 73.76	85.77 / 85.28	40.21
3	72.07 / **74.14**	74.14 / **76.63**	85.53 / **86.57**	**45.67**
4	71.02 / 71.06	74.30 / 74.03	85.09 / 85.16	43.82
5	72.26 / 70.12	**75.83** / 72.59	**85.88** / 84.77	44.10

Table 12: Ranking of topical relevance and utility judgments performance (%) of ITEM-AR$_r$ using Mistral on the TREC DL dataset. "a/b" means "relevance ranking performance / utility ranking performance". Numbers with underline mean better performance among all variants of ITEM with the same m.

Dataset	#Psg	#PsgLen	#Q	#Rels/Q
TREC	8.8M	93	82	212.8
WebAP	379k	45	73	76.4
NQ	21M	100	1868	1.0
GTI-NQ	10	100	1863	1.0

Table 13: Statistics of experimental datasets.

H.2 Instruction of the Ranking Approach

For RankGPT, we directly use the instruction of Sun et al. (2023) for relevance ranking, as shown in Figure 14. For the relevance ranking in our ITEM, the instructions are shown in Figure 12 and Figure 13.

H.3 Instruction of Answer Generation

Li et al. (2023) utilize LLM to generate the missing information in the provided documents for the current question and then re-retrieve it as relevant feedback. Therefore, we have also designed two kinds of pseudo answers for utility judgments, i.e., (i) the explicit answer, which produces an answer based on the given information, and (ii) the implicit answer, which does not answer the question directly but gives the information necessary to answer the question. The two instructions are shown in Figure 15 and Figure 16.

Methods	#IT/Q	#OT/Q	#RT(s)/Q
Vanilla	2507	23	0.49
UJ-ExpA	2529	59	0.94
UJ-ImpA	2533	41	0.71
5-sampling	12647	296	4.72
ITEM-A$_s$(m=1)	4628	47	0.96
ITEM-A$_s$(m=3)	10603	154	2.13
ITEM-AR$_s$(m=1)	7107	211	3.15
ITEM-AR$_s$(m=3)	18224	624	8.61

Table 14: An empirical analysis of the actual cost of baselines and our ITEM. "IT", "OT", and "RT" mean input tokens, output tokens, and run time, respectively.

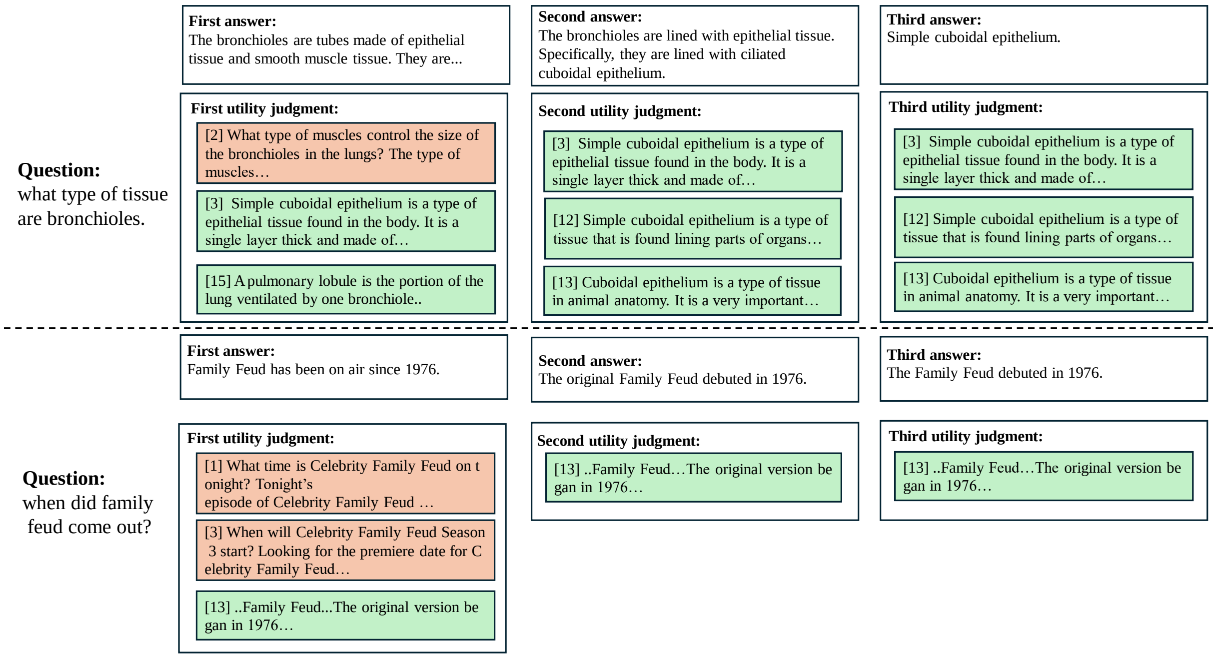

Figure 9: An example of our ITEM-A$_s$ using Mistral on the TREC dataset. Green means the passage has utility, orange means the passage does not have utility.

user:
You are the utility judge, an intelligent assistant that can select the passages that have utility in answering the question.
assistant:
Yes, i am the utility judge.
user:
I will provide you with {num} passages, each indicated by number identifier [].
I will also provide you with a reference answer to the question.
Select the passages that have utility in generating the reference answer to the following question from the {num} passages: {query}.
assistant:
Okay, please provide the passages and the reference answer.
user:
[1] {{passage_1}}
assistant:
Received passage [1]
user:
[2] {{passage_2}}
assistant:
Received passage [2]
(more passages) ...
user:
Question: {query}.
Reference answer: {answer}.
The requirements for judging whether a passage has utility in answering the question are: The passage has utility in answering the question, meaning that the passage not only be relevant to the question, but also be useful in generating a correct, reasonable and perfect answer to the question. Directly output the passages you selected that have utility in generating the reference answer to the question. The format of the output is: 'My selection:[[i],[j],...].'. Only response the selection results, do not say any word or explain.

Figure 10: Instruction in the listwise approach.

user:
You are the utility judger, an intelligent assistant that can judge whether a passage has utility in answering the question or not.

assistant:
Yes, i am the utility judger.

user:
I will provide you with a passage and the reference answer to the question. \n Judge whether the passage has utility in generating the reference answer to the following question or not: {query}.

assistant :
Okay, please provide the passage and the reference answer to the question.

user:
Question: {query}.
Reference answer: {answer}.
Passage: {passage}
The requirements for judging whether a passage has utility in answering the question are: The passage has utility in answering the question, meaning that the passage not only be relevant to the question, but also be useful in generating a correct, reasonable and perfect answer to the question.
The reference answer may not be the correct answer, but it provides a pattern of the correct answer. Directly output whether the passage has utility in generating the reference answer to the question or not. If the passage has utility in generating the reference answer, output 'My judgment: Yes, the passage has utility in answering the question.'; otherwise, output 'My judgment: No, the passage has no utility in answering the question.'.

Figure 11: Instruction in the pointwise approach.

user:
You are RankGPT, an intelligent assistant that can rank passages based on their relevance to the query.

assistant:
Yes, i am RankGPT.

user:
I will provide you with {num} passages, each indicated by number identifier []. I will also give you a reference answer to the query. Rank the passages based on their relevance to query: {query}.

assistant :
Okay, please provide the passages and the reference answer.

user:
[1] {{passage_1}}

assistant :
Received passage [1]

user:
[1] {{passage_2}}

assistant :
Received passage [2]

(more passages) ...
user:
Query: {query}.
Reference answer: {answer}
Rank the {num} passages above based on their relevance to the query. The passages should be listed in descending order using identifiers. The most relevant passages should be listed first. The output format should be [] > [] > [] > ..., e.g., [i] > [j] > [k] > ...
Only response the ranking results, do not say any word or explain.

Figure 12: Instruction of the relevance ranking approach in our ITEM.

user:
You are RankGPT, an intelligent assistant that can rank passages based on their utility in generating the given reference answer to the question.
assistant:
Yes, i am RankGPT.
user:
I will provide you with {num} passages, each indicated by number identifier []. I will also give you a reference answer to the question.
Rank the passages based on their utility in generating the reference answer to the question: {query}.
assistant :
Okay, please provide the passages and the reference answer.
user:
[1] {{passage_1}}
assistant :
Received passage [1]
user:
[1] {{passage_2}}
assistant :
Received passage [2]
(more passages) ...
user:
Question: {query}.
Reference answer: {answer}
Rank the {num} passages above based on their utility in generating the reference answer to the question. The passages should be listed in utility descending order using identifiers. The passages that have utility in generating the reference answer to the question should be listed first. The output format should be [] > [] > [] > ..., e.g., [i] > [j] > [k] > ... Only response the ranking results, do not say any word or explain.

Figure 13: Instruction of the utility ranking approach in our ITEM.

user:
You are RankGPT, an intelligent assistant that can rank passages based on their relevance to the query.
assistant:
Yes, i am RankGPT.
user:
I will provide you with {num} passages, each indicated by number identifier [].
Rank the passages based on their relevance to query: {query}.
assistant :
Okay, please provide the passages.
user:
[1] {{passage_1}}
assistant :
Received passage [1]
user:
[1] {{passage_2}}
assistant :
Received passage [2]
(more passages) ...
user:
Query: {query}.
Rank the {num} passages above based on their relevance to the query. The passages should be listed in descending order using identifiers. The most relevant passages should be listed first. The output format should be [] > [] > [] > ..., e.g., [i] > [j] > [k] > ... Only response the ranking results, do not say any word or explain.

Figure 14: Instruction of the ranking approach in Sun et al. (2023).

user:
You are a faithful question and answer assistant. Answer the question based on the given information with one or few words/sentences without the source.
assistant:
Yes, i am the faithful question and answer assistant.

user:
Given the information:
{passage}
Answer the following question based on the given information with one or few words/sentences without the source.
Question: {question}
Answer:

Figure 15: Instruction of the explicit answer generation.

user:
You are a faithful question and answer assistant. Given a question and references. To answer the question, output which information is necessary to answer the question based on the references.
assistant:
Yes, i am the faithful question and answer assistant.

user:
References: {pas}
Question: {question}
To answer the question, output which information is necessary to answer the question based on the references. Do not mention references when printing out necessary information. The format of the output is: 'Necessary information: [xxx]'.

Figure 16: Instruction of the implicit answer generation.