Iterative Utility Judgment Framework via LLMs Inspired by Relevance in Philosophy

Anonymous ACL submission

Abstract

Utility and topical relevance are critical measures in information retrieval (IR), reflecting system and user perspectives, respectively. While topic relevance has long been emphasized, utility is a higher standard of relevance and is more useful for facilitating downstream tasks, e.g., in Retrieval-Augmented Generation (RAG). When we incorporate utility judgments into RAG, we realize that the topical relevance, utility, and answering in RAG are closely related to the three types of relevance that Schutz discussed from a philosophical perspective. They are topical relevance, interpretational relevance, and motivational relevance, respectively. Inspired by the dynamic iterations of the three types of relevance, we propose an Iterative utiliTy judgmEnt fraMework (ITEM) to promote each step of the cycle of RAG. We conducted extensive experiments on multi-grade passage retrieval and factoid question-answering datasets (i.e., TREC DL, WebAP, and NQ). Experimental results demonstrate significant improvements in utility judgments, ranking of topical relevance, and answer generation upon representative baselines, including multiple single-shot utility judging approaches. Our code and benchmark can be found at `https://anonymous.4open.science/r/ITEM-B486/`.

1 Introduction

Utility and topical relevance measure Information Retrieval (IR) performance from the views of users and systems, respectively (Bruce, 1994). Utility refers to the usefulness of retrieval items to an information seeker, gauged by their "value", "appropriateness in resolution of a problem" or "reduction of uncertainty", etc. (Saracevic, 1996; Schamber and Eisenberg, 1988; Saracevic, 1975). In contrast, topical relevance measures the matching between the subjects of queries and documents, which indicates aboutness (Saracevic et al., 1988). Topical relevance does not necessarily mean utility, while utility indicates a high standard of relevance.

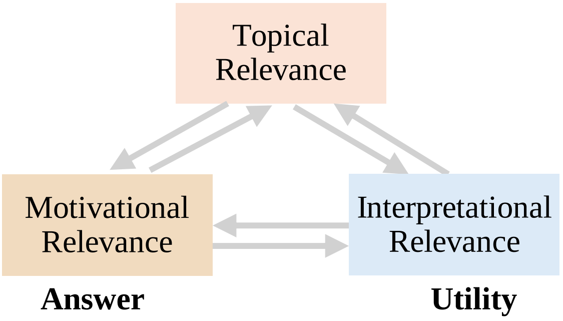

Figure 1: Schutz's "system of relevancies" and the RAG framework, as well as the correspondence between these two frameworks.

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 the system view of relevance for a long time (Bruce, 1994). In recent years, most IR models have been experimented on the MS-MARCO dataset (Nguyen et al., 2016) and optimized towards the binary labels, which is inadequate to capture utility. More recently, leveraging Large Language Models (LLMs) (OpenAI, 2022; Meta, 2024; Jiang et al., 2023a; OpenAI, 2023) for topical relevance judgments (Khramtsova et al., 2024; Faggioli et al., 2023; Thomas et al., 2024; Sun et al., 2023) has drawn more attention than judging utility (Zhang et al., 2024).

In the modern LLM era, Retrieval-Augmented Generation (RAG) has become a hot research topic that facilitates LLMs with external knowledge (Xie et al., 2023; Shi et al., 2023; Izacard et al., 2023; Su et al., 2024; Glass et al., 2022). It requests retrieval results useful for answering the question. 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. In this paper, we aim to promote the utility judgment performance of LLMs so that relevance ranking can be boosted and RAG can be enhanced by high-utility references.

We find that the interpretation of relevance in philosophy by Schutz (Schutz, 1970) is closely related to RAG. Schutz considered relevance as the property that determines the connections and relations in our lifeworld. He identified three types of basic and interdependent relevance that interact dynamically within a "system of relevancies" (Saracevic, 1996): (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 1. We consider that topical relevance, interpretational relevance, and motivational relevance are closely related to topical relevance, utility, and answering respectively in RAG. Inspired by the dynamic interaction of the relevance system, 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 1, i.e., the judgments of topical relevance, utility, and answer generation. While previous work (Zhang et al., 2024) has shown that utility judgments via LLMs can be enhanced by simultaneously answering the question, iteratively conducting utility judgments and question answering (QA) has not been explored, not to mention that topic relevance can also be involved in the iteration. We propose an Iterative utiliTy judgmEnt fraMework (ITEM) to enhance the utility judgment and QA performance of LLMs by interactions between the steps.

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), and factoid QA (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.

2 Related Work

Relevance-Focused Information Retrieval. Relevance is a crucial concept in information retrieval (IR), and the automatic generation of relevance judgments has been a long-standing goal in IR that has been studied over the past decades (Saracevic, 1975; Järvelin and Kekäläinen, 2017; Makary et al., 2017, 2016; MacAvaney and Soldaini, 2023). Recent studies demonstrate promising outcomes in utilizing LLMs to automatically generate relevance judgments (Khramtsova et al., 2024; Faggioli et al., 2023; Thomas et al., 2024) and mainly contain the following three approaches, i.e., pointwise (Zhuang et al., 2024; Nogueira and Cho, 2019), pairwise (Qin et al., 2023; Jiang et al., 2023b), and listwise (Sun et al., 2023; Pradeep et al., 2023). For example, Faggioli et al. (2023) used zero/few-shot prompting to instruct LLMs to judge the relevance of an item to a query in a pointwise approach. (Sun et al., 2023) analyzed the capabilities of LLMs in the relevance ranking.

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., 2023c; 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

We formally define the task of utility judgments via LLMs and 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 $\mathcal{D} = [p_1, p_2, ..., p_n]$. The goal of utility judgments for LLMs is to identify a set of passages U that have utility in answering the question q. There are two typical input approaches for LLMs:

- Listwise, $U = f_l(q, \mathcal{D}, I)$, where I is the instruction of LLMs and $f_l(\cdot)$ is the utility judgments function based on the whole candidate list $\mathcal{D}$ and the output is a set of passages.
- Pointwise, $U = \{p_i \mid f_p(q, p_i, I) = 1, p_i \in \mathcal{D}\}$ where I is the instruction of LLMs and $f_p(\cdot)$ is the utility judgments function with a single passage as the input.

3.2 Sing-Shot Utility Judgments

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

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 2: I_a instruction contains the *implicit answer* and *explicit answer*.

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.

Figure 3: I_u instruction contains listwise and pointwise approaches.

3.3 Iterative utiliTy judgmEnt fraMework (ITEM)

Schutz (Schutz, 1970) emphasized the existence of various types of relevance and underscored the interactivity and interdependence between these various types from a much broader arena than information science. Inspired by the powerful insight, we propose an Iterative utiliTy judgmEnt fraMework (ITEM). Specifically, as shown in Figure 4, dynamic iteration of the topical relevance, pseudo answer generation, and utility judgments.

ITEM with Answering in the Loop (ITEM-A). In a formal sense, at iteration t ($t \geq 1$), the pseudo answer a_t is generated based on the judging result U_{t-1} in the previous iteration and then judged based on the a_t and candidate passages list $\mathcal{D}$ to get a set of passages that have utility U_t:

$$a_t = f_a(q, U_{t-1}, I_a), \tag{1}$$

$$U_t = f_u(q, \mathcal{D}, a_t, I_u), \tag{2}$$

where $f_a(\cdot)$ is an answer generation function, $f_u(\cdot)$ is a utility judgment function, which can be $f_l(\cdot)$ or $f_p(\cdot)$, I_a is the answer prompts for LLMs (as detailed in Figure 2), and I_u is the utility judgments prompts for LLMs (as detailed in Figure 3). a_i can be in two forms: (i) *explicit answer* to the question q; (ii) *implicit answer* that specifies necessary information to answer the question q. We consider $U_0 = \{p | p \in R_0\}$ as the initial candidate set and $R_0 = \mathcal{D}$ is the initial results ranked by a retriever like BM25 (Robertson et al., 2009).

ITEM with both Answering and Ranking of Topical Relevance in the Loop (ITEM-AR). In

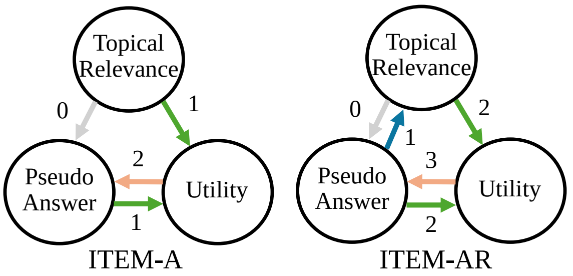

Figure 4: The flowchart shows the first iteration of the ITEM. For ITEM-A, the process involves step 1 (utility judgments) followed by step 2 (pseudo answer generation). For ITEM-AR, the process includes step 1 (relevance ranking), step 2 (utility judgments), and step 3 (pseudo answer generation). Future iterations alternate between these steps.

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 5: I_r instruction

Schutz's theory, topical relevance, motivational relevance, and interpretive relevance are all dynamically affected. In the ITEM-A framework, the topical relevance is not updated during the iteration process. Consequently, we have incorporated a relevance ranking task into the ITEM framework, which ensures 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} in the previous iteration. Subsequently, given a_t, the passage list R_{t-1} is ranked based on the relevance to the question to obtain R_t. Finally, the judging result U_t is derived using the input R_t and the answer a_t:

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

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

$$U_t = f_u(q, R_t, a_t, I_u), \tag{5}$$

where $f_r(\cdot)$ and I_r are the relevance ranking function and the relevance ranking prompt for LLMs (as detailed in Figure 5), respectively.

At iteration t, we have two ways to produce the set U_t: (i) asking LLMs to identify the set of passages that have utility, which we call ITEM-A$_s$ or ITEM-AR$_s$; (ii) asking LLMs to provide a ranked passage list with the listwise input approach and we deem the set of first k elements in the list as U_t, which we call ITEM-A$_r$ or ITEM-AR$_r$. We set k as 5 and more details of k selection are shown in Appendix B.4. In our experiments, we find that ITEM-AR$_r$ does not improve ranking performance as well as ITEM-A$_r$ (see Section 6.1 for experimental analysis), so we do not use ITEM-AR$_r$ in our experiments to obtain U_t.

Full details of all prompts can be found in Appendix A. We stop the iteration when at most m iterations are reached or the set of passages considered to have utility does not change, i.e., $t = m$ or $U_t = U_{t-1}$. If the value of m is relatively small, it may negatively impact the performance of the iteration. Conversely, if the value of m is too large, it may increase the computational cost of the iteration. Considering the balance between performance and efficiency, we selected m=3. Further discussion about m is provided in Section 6.1.

4 Experimental Setup

4.1 Datasets

Our experiments are conducted on three benchmark datasets, including, two retrieval datasets, i.e., TREC DL (Craswell et al., 2020), WebAP (Yang et al., 2016), and an open-domain question answer (ODQA) dataset, i.e., NQ (Kwiatkowski et al., 2019).

TREC. 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 by the highest point and combine them to form a whole dataset, i.e., the TREC dataset in our experiments.

WebAP. WebAP (Yang et al., 2016) is a non-factoid answer passage collection built on Gov2. More details about non-factoid answer passage retrieval are provided in Appendix F. It retrieves the top 50 documents with the Sequential Dependency Model (SDM) (Metzler and Croft, 2005) for the description queries that need to be answered with passages. Relevant documents are annotated, resulting in 3843 passages from 1200 documents, categorized as "perfect", "excel", "good", and "fair". In our experiments, we split the remaining documents into passages of 2 or 3 sentences to be irrelevant and considered "perfect" 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. Detailed statistics of all the experimental datasets are shown in Table 16 in the Appendix E.

4.2 Evaluation metrics

For the utility judgments task, we evaluate the results of judgments using **P**recision, **R**ecall, and **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 metric (EM) and F1, as used by Rajpurkar et al. (2016).

4.3 LLMs

we conduct our experiments using several representative LLMs, i.e., (i) the well-known 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 after instruction tuning), and (iii) Llama 3 (Meta, 2024) (the Meta-Llama-3-8B-Instruct version promoted with instruction tuning based on the vanilla Llama 3). To ensure the reproducibility of the experiments, the temperature for all experiments is set to 0.

4.4 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 and WebAP datasets), respectively. Based on the retrieval results to build the $\mathcal{D}$ we have two settings: (i) For TREC 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).

4.5 Baselines

We choose the following baselines for comparison:

- **Sing-shot utility judgments**: (i) **Vanilla**: Ask LLMs to provide utility judgments based on the instruction directly. (ii) **UJ-ExpA**: Utility judgments facilitated with explicit answers, which is shown to be effective in Zhang et al. (2024). (iii) **UJ-ImpA**: We utilize the necessary information that is necessary for answering the question to help with the utility judgments task. More details can be found in Section 3.2.
- **k-sampling** is proposed by (Zhang et al., 2024): In addition to the original input order, k-sampling method randomizes the order of input passage list k times and aggregates the $k+1$ utility judgement results through voting. The output of the k-sampling method each time contains explicit answers and utility judgments. Our iterative framework outputs both answers and utility judgments at each iteration. To ensure fairness in calculating the parameters of LLM, we set $k = 5$. The detailed comparison of the parameter calculations of the two is in Appendix G. The k-sampling method requires $k+1$ outputs of LLMs, which can be more expensive for practical applications involving the LLMs. Moreover, it is difficult to determine the stability of the utility judgments due to the sampling technique.
- **RankGPT**: Sun et al. (2023) first proposed using the LLMs in the ranking of topical relevance by directly ranking input passages based on their relevance to the query. The prompt is shown in Table 13 in Appendix A.2

5 Experimental Results of LLMs

Our framework consists of three tasks: utility judgments, ranking of topical relevance, and answer generation. This section will present the performance of each task within our framework. By default, the pseudo answer is the *explicit answer* in all experiments, if not specified otherwise.

5.1 Utility Judgments on Retrieval Datasets

Tabel 1 shows the F1 performance on the TREC and WebAP datasets using three LLMs. Since ITEM-A$_r$ has poor F1 performance in utility judgments (refer to Table 12 in Appendix B.1 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 three LLMs using our ITEM with a single iteration generally outperform the single-shot utility judgments on both datasets and may even surpass the k-sampling method. For example, ChatGPT on the TREC 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. On the WebAP

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

Table 1: The F1 performance (%) of utility judgments with different LLMs ("Mis", "Lla", and "Cha" are Mistral, Llama 3, and ChatGPT, respectively) on the different datasets (the numbers in parentheses represent m-values). "-" indicates that the k-sampling method and our ITEM-AR$_s$ require listwise input, and no experiments are performed under the pointwise approach. Numbers in bold indicate the best performance.

Method	Mistral		Llama 3		ChatGPT	
	T	W	T	W	T	W
BM25	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-AR$_s$ (1)	71.29	37.48	77.22	43.80	81.38	48.42
ITEM-AR$_s$ (3)	73.24	**45.45**	74.80	44.87	82.89	48.80
ITEM-A$_r$ (1)	70.57	37.11	73.95	40.89	80.79	50.30
ITEM-A$_r$ (3)	**74.27**	43.80	**77.34**	**45.88**	**83.12**	**51.61**

Table 2: The NDCG@5 performance (%) of the topical relevance ranking using different LLMs on the different datasets. Numbers in bold indicate the best performance. "T" and "W" refer to the TREC dataset and the WebAP dataset, respectively.

dataset, Mistral and Llama 3 use single-iteration ITEM-AR$_s$ to outperform all baselines and ITEM-A$_s$. This might be due to that better topical relevance ranking leads to better utility judgments, which is also better than k-sampling in terms of efficacy and efficiency.

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. 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. However, in some specific cases, multiple iterations may not outperform single iterations, likely due to the unpredictable nature of zero-shot settings. 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.

ExpA vs. ImpA. For the single-shot baselines, explicit answers generally outperform implicit answers in utility judgments when using the listwise approach, while the opposite is true for the pointwise approach. This may be because pointwise inputs provide less information, making explicit answers less effective, whereas implicit answers better address the question's information needs. Unlike the pseudo answer generated by the input passage of the single-shot pointwise approach, our ITEM's pointwise approach generates the pseudo answer based on the results of utility judgments from the previous round. Therefore, after multiple iterations, ITEM shows greater improvement with explicit answers compared to implicit answers in both input approaches in most cases.

Comparisons between LLMs. ChatGPT outperforms other LLMs on both datasets using both input approaches. For instance, on the TREC dataset, ChatGPT achieves an F1 improvement of 15.2% and 12.8% using the listwise approach after multiple iterations of ITEM-A$_s$ $w.$ ExpA method, compared to Mistral and Llama 3, respectively.

Listwise vs. Pointwise. The average performance of utility judgments for LLMs is better with the listwise approach than with the pointwise approach. For example, Mistral shows a 6.0% F1 improvement with the listwise approach over the pointwise approach after multiple iterations of the ITEM-A$_s$ $w.$ ExpA method on the TREC dataset. More detailed performance metrics, i.e., Precision, Recall, F1, are provided in Appendix B.1.

References	Mistral		Llama 3		ChatGPT	
	EM	F1	EM	F1	EM	F1
Golden	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$_s$ (1)	32.98	49.00	50.16	61.88	49.38	59.78
ITEM-A$_s$ (3)	**33.73**	**49.63**	50.27	**62.09**	**49.69**	**60.18**
ITEM-AR$_s$ (1)	33.30	49.26	50.27	61.69	49.52	59.64
ITEM-AR$_s$ (3)	33.40	49.27	49.36	60.97	49.06	59.67

Table 3: The answer generation performance (%) of all LLMs on the NQ dataset using reference passages collected from different methods (the numbers in parentheses represent m-values). Numbers in bold indicate the best performance except the answer performance using golden evidence.

5.2 Ranking Performance on Retrieval Datasets

We also assess whether the ranking of topical relevance has been improved within ITEM. In terms of topical relevance, the utility ranking results produced by ITEM-A$_r$ and the relevance ranking results yielded from ITEM-AR$_s$. From Table 2, 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 and 27.7% on the WebAP dataset using the Mistral. The performance improvement may stem from the interaction between tasks. (ii) After iterations, relevance ranking, and utility ranking performances have been improved on both datasets and all LLMs. The topical relevance ranking benefits from our dynamic iterative framework, confirming Schutz's theory of dynamic iterative interaction. (iii) Experiments show that utility ranking generally outperforms relevance ranking, which confirms the effectiveness of utility in the ranking of topical relevance.

5.3 Results of Answer Generation

From Table 1, we find that the listwise approach generally outperforms the pointwise approach for utility judgments. Consequently, our NQ dataset experiments utilize only the listwise approach. The following observations can be made from Table 3: (i) The k-sampling method has the best performing, especial on F1 score, for all the baselines. (ii) 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. (iii) The results from Llama 3 have better answer performance than ChatGPT, and we suspect that Llama 3 has been over-tuned on the NQ dataset.

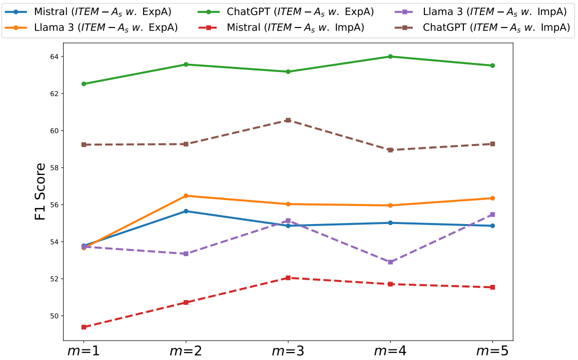

Figure 6: Utility judgments performance (%) of different LLMs in terms of m values in ITEM-A$_s$ on TREC dataset.

m	ITEM-A$_s$			ITEM-AR$_s$		
	Utility	Answer		Utility	Answer	
	F1	EM	F1	F1	EM	F1
m=1	28.83	32.98	49.00	24.96	33.30	49.26
m=2	**29.02**	**34.31**	**50.08**	25.87	33.57	49.16
m=3	28.90	33.73	49.63	26.17	33.40	49.27
m=4	29.01	34.21	50.07	**26.54**	33.46	49.24
m=5	28.91	33.78	49.63	26.48	**33.89**	**49.58**

Table 4: The utility judgments performance (%) and answer generation performance (%) of Mistral in ITEM-A$_s$ and ITEM-AR$_s$ on the NQ dataset in terms of m values. Numbers in bold indicate the best performance.

6 Further Analyses

6.1 Effect of Iteration Numbers

Figure 6 illustrates the performance of the LLMs in making utility judgments under ITEM-A$_s$ with varying maximum iteration rounds m on the TREC dataset. Figure 7 illustrates the ranking of topical relevance performance in terms of m values. Since the NQ dataset lacks annotations for passage utility, we automatically annotated the retrieved passages based on the utility standard, as detailed in Appendix D. Table 4 displays the results for utility judgments and answer generation using Mistral under ITEM-A$_s$ on the NQ dataset.

As can be seen from Figure 6, Figure 7 and Table 4, varying the value of m affects the performance of utility judgments, ranking of topical relevance and answer generation. However, the performance when $m > 1$ is consistently better than when $m = 1$. Moreover, utility ranking gen-

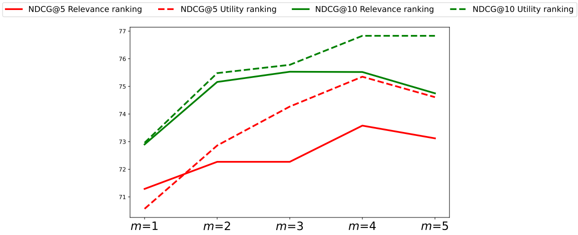

Figure 7: Topical relevance performance (%) of Mistral in ITEM-AR$_s$ and ITEM-A$_r$ on the TREC dataset in terms of m values.

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 5: Ranking of topical relevance and utility judgments performance (%) of ITEM-AR$_r$ using Mistral on the TREC dataset. "a/b" means "relevance ranking performance / utility ranking performance".

erally outperforms relevance ranking, which confirms the effectiveness of utility in the ranking of topical relevance. Further details on the experiments with different m values are provided in Appendix B.1.

In addition, we evaluate two ranking performances of ITEM-AR$_r$, with the experimental results shown in Table 5. Under the ITEM-AR$_r$ framework, relevance ranking is followed by utility ranking, resulting in further improved utility ranking performance compared to relevance ranking. However, as seen in Table 2 and Table 5, performing ranking twice in the same iteration may not yield better ranking results than performing utility ranking once in the iteration.

6.2 Iteration Stop Conditions

The stopping condition takes into account the answer performance in our ITEM in addition to the utility judgments. we compute the ROUGE-L (Lin, 2004) score 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. More details of experiments on iteration stop conditions are shown in Appendix B.2. The results show that using different stopping conditions does affect the performance of utility judgments. However, using the answer as a stopping condition, different models on different datasets may need to look for different p, which is not very flexible.

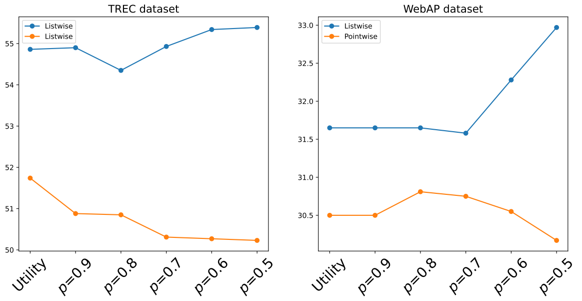

Figure 8: The utility judgments performance (%) of Mistral in different iteration stop conditions (m=3) on ITEM-A$_s$.

7 Case Study

Table 16 and Table 17 in Appendix C present two good cases (demonstrating improved utility judgments after multiple iterations) from the TREC dataset using Mistral on ITEM-A$_s$. In Table 16, the first pseudo answer, though relatively correct, includes irrelevant information, leading to a misjudgment of "Passage-2" as "utility". "Passage-2" does not address the question of what tissue bronchioles belong to but instead discusses the type of muscles that control the size of the bronchioles in the lungs. 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. In Table 17, the first pseudo answer is correct, but two irrelevant passages are judged as "utility". The second pseudo answer, with slight rewording, results in all passages being correctly identified as "utility".

8 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. 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. 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 retrieval and utility judgments.

Limitations

In this paper, we proposed an iterative utility judgment framework inspired by relevance in philosophy. There are three primary limitations that should be acknowledged: (i) Our methods are applied in zero-shot scenarios without any training. The zero-shot method has inherent disadvantages, e.g., being sensitive to prompts and having unstable performance. 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. (iii) Although our iterative framework is very effective, the number of iterations will increase the cost of calling large models. How to reduce the iteration cost is also something that needs further improvement in the future.

Ethics Statement

In conducting this research, we have prioritized ethical considerations at every stage to ensure the responsible development and application of AI technologies. Our research does not rely on personally identifiable information or require manually annotated datasets. We firmly believe in the principles of open research and the scientific value of reproducibility. To this end, we have made all models, 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, et al. 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.

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*.

Guglielmo Faggioli, Laura Dietz, Charles LA Clarke, Gianluca Demartini, Matthias Hagen, Claudia Hauff, Noriko Kando, Evangelos Kanoulas, Martin Potthast, Benno Stein, et al. 2023. Perspectives on large language models for relevance judgment. In *Proceedings of the 2023 ACM SIGIR International Conference on Theory of Information Retrieval*, pages 39–50.

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.

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.

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, et al. 2023a. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. 2023b. Llm-blender: Ensembling large language models with pairwise ranking and generative fusion. *ACL*.

Zhengbao Jiang, Frank Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023c. 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.

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

Ekaterina Khramtsova, Shengyao Zhuang, Mahsa Baktashmotlagh, and Guido Zuccon. 2024. Leveraging llms for unsupervised dense retriever ranking. *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24)*.

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

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 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.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Sean MacAvaney and Luca Soldaini. 2023. One-shot labeling for automatic relevance estimation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2230–2235.

Mireille Makary, Michael Oakes, Ruslan Mitkov, and Fadi Yammout. 2017. Using supervised machine learning to automatically build relevance judgments for a test collection. In *2017 28th International Workshop on Database and Expert Systems Applications (DEXA)*, pages 108–112. IEEE.

Mireille Makary, Michael Oakes, and Fadi Yamout. 2016. Towards automatic generation of relevance judgments for a test collection. In *2016 Eleventh International Conference on Digital Information Management (ICDIM)*, pages 121–126. IEEE.

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

Donald Metzler and W Bruce Croft. 2005. A markov random field model for term dependencies. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 472–479.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. In *Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, December 9, 2016*, volume 1773 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Rodrigo Frassetto Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with BERT. *CoRR*, abs/1901.04085.

OpenAI. 2022. Introducing chatgpt.

OpenAI. 2023. Gpt-4 technical report.

Ronak Pradeep, Sahel Sharifymoghaddam, and Jimmy Lin. 2023. Rankvicuna: Zero-shot listwise document reranking with open-source large language models. *arXiv preprint arXiv:2309.15088*.

Zhen Qin, Rolf Jagerman, Kai Hui, Honglei Zhuang, Junru Wu, Jiaming Shen, Tianqi Liu, Jialu Liu, Donald Metzler, Xuanhui Wang, et al. 2023. Large language models are effective text rankers with pairwise ranking prompting. *2024 Annual Conference of the North American Chapter of the Association for Computational Linguistics*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 2383–2392. The Association for Computational Linguistics.

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, et al. 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, et al. 2023. Beyond human data: Scaling self-training for problem-solving with language models. *arXiv preprint arXiv:2312.06585*.

Ivan Stelmakh, Yi Luan, Bhuwan Dhingra, and Ming-Wei Chang. 2022. ASQA: factoid questions meet long-form answers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 8273–8288. Association for Computational Linguistics.

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*.

Paul Thomas, Seth Spielman, Nick Craswell, and Bhaskar Mitra. 2024. Large language models can accurately predict searcher preferences. *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24)*.

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.

Ellen M Voorhees et al. 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.

Honglei Zhuang, Zhen Qin, Kai Hui, Junru Wu, Le Yan, Xuanhui Wang, and Michael Berdersky. 2024. Beyond yes and no: Improving zero-shot llm rankers via scoring fine-grained relevance labels. *2024 Annual Conference of the North American Chapter of the Association for Computational Linguistics*.

A Instruction Details

A.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. Considering the quality of the pseudo answer, we design a note, i.e., “The reference answer may not be the correct answer, but it provides a pattern of the correct answer.”, adding before judging. We do experiments on listwise and pointwise respectively, and find that listwise is not sensitive to this note, note has a great influence on pointwise, as shown in Table 6 Finally, we choose prompt as shown in Figure 9 and Figure 10.

	listwise		pointwise	
m	$w.$ Note	w/o Note	$w.$ Note	w/o Note
m=1	53.56	53.91	49.44	48.34
m=2	55.30	55.65	51.09	49.44
m=3	54.57	54.86	51.74	50.02
m=4	53.87	55.02	51.90	50.30
m=5	53.99	54.86	51.65	50.60

Table 6: Different F1 performance (%) of ITEM using Mistral on TREC dataset in the listwise approach and the pointwise approach under different m values between “w/o Note”, i.e., without the note and ”$w.$ Note”, i.e., with the note.

Moreover, for the implicit answer, we consider “what information is necessary to answer the question” and “which information is necessary to answer the question” in the prompt. Experiments are shown in Table 7. Finally, we choose prompt as shown in Figure 15. For the experiments of ChatGPT API, we have the cost of approximately $248.

	TREC		WebAP	
m	$w.$ which	$w.$ what	$w.$ which	$w.$ what
m=1	49.39	48.34	26.06	25.81
m=2	50.72	50.13	28.67	25.38
m=3	52.05	50.94	28.36	25.04
m=4	51.71	51.54	25.46	26.87
m=5	51.54	50.66	27.35	26.69

Table 7: Different F1 performance (%) of ITEM using Mistral on TREC dataset and WebAP dataset in the listwise approach under different m values between “$w.$ which” and ”$w.$ what”.

A.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 13. For the relevance ranking in our ITEM, the instructions are shown in Figure 11 and Figure 12.

A.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. For “words/sentences”, the experiments found that the

user:
You are the utility judger, an intelligent assistant that can select the passages that have utility in answering the question.

assistant:
Yes, i am the utility judger.

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:
[1] {{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 9: Instruction in the listwise approach.

length of the explicit answers generated by Mistral using “sentences” is too long for factual questions, whereas Llama 3 and chatGPT use “sentences“ to generate answers of moderate length. So we only use “words“ on the TREC dataset and the NQ dataset using Mistral. The two instructions are shown in Figure 14 and Figure 15.

B Experiment Details

We utilized AI models such as ChatGPT to perform grammar corrections on portions of our paper.

B.1 Effect of Iteration Numbers

The performance of different LLMs in different datasets is shown in Table 8, Table 9, Table 10, Table 11 and Table 12.

B.2 Iteration Stop Conditions

We consider two stop conditions for ITEM, i.e., utility judgments and answer. The different performances of Mistral with different stop conditions are in Table 13.

B.3 Quality of Utility Judgments

The retrieved data we use are a four-point scale, and we consider the highest level as having utility. To see the different points in our results for utility judgments, we analyzed the various ranks as a percentage of the results on TREC for mistral in Table 14. From Table 14, we can see that almost 70% of the results of utility judgments are very relevant to the question.

B.4 k values in ITEM-A$_r$

Different ranking performance of k values in ITEM-A$_r$ is shown in Table 15. Considering the performance of utility ranking and utility judgments, we set k=5.

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 8: 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 9: 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 10: 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 11: 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.

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 10: Instruction in the pointwise approach.

m	NDCG@5	NDCG@10	NDCG@20	Utility-judgments F1
1	71.29 / 70.57	72.90 / 72.69	84.56 / 84.08	52.34 / 42.02
2	72.27 / 72.86	75.16 / 75.48	85.76 / 86.09	53.40 / 42.10
3	73.24 / 74.27	75.53 / 75.78	86.59 / 86.80	56.27 / 44.18
4	73.58 / 75.35	75.52 / 76.83	86.46 / 87.23	56.07 / 44.68
5	73.12 / 74.61	74.75 / 76.20	85.95 / 86.82	57.82 / 44.25

Table 12: Ranking and utility judgments performance (%) of Mistral in ITEM-AR$_s$ and ITEM-A$_r$ on the TREC dataset on different m values. "a/b" means ITEM-AR$_s$/ITEM-A$_r$.

C Case Study

We show two cases on the TREC dataset in Table 16 and Table 17.

D Annotations in NQ Dataset

To evaluate the performance of utility judgments under different values of m, we relabeled the retrieved passages following Stelmakh et al. (2022). Specifically, we used a QA model based on RoBERTa (Liu et al., 2019) to extract answers from the retrieved passages. A passage is considered utility if the extracted answers match the ground-truth answers. The advantage of this labeling method is its relatively high accuracy, while its disadvantage is the potentially low recall rate.

E Datasets

Detailed statistics of the experimental datasets are shown in Table 16.

F 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.

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 11: 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 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 12: Instruction of the utility ranking approach in our ITEM.

Iteration stop condiction	TREC		WebAP	
	Listwise	Pointwise	Listwise	Pointwise
Utility (m=1)	53.78	49.44	29.76	29.10
Utility (m=2)	**55.65**	51.09	29.83	29.88
Utility (m=3)	54.86	51.74	31.65	30.50
Utility (m=4)	55.02	**51.90**	28.96	31.00
Utility (m=5)	54.86	51.65	31.86	30.85
Answer w. 0.9 (m=1)	53.78	49.44	29.76	29.10
Answer w. 0.9 (m=2)	55.65	50.45	29.83	29.88
Answer w. 0.9 (m=3)	54.90	50.88	31.65	30.50
Answer w. 0.9 (m=4)	55.02	51.23	28.96	31.00
Answer w. 0.9 (m=5)	54.90	50.79	31.86	30.85
Answer w. 0.8 (m=1)	53.78	49.44	29.76	29.10
Answer w. 0.8 (m=2)	55.16	50.46	29.83	30.20
Answer w. 0.8 (m=3)	54.35	50.85	31.65	30.81
Answer w. 0.8 (m=4)	54.53	51.20	28.96	31.30
Answer w. 0.8 (m=5)	54.35	50.76	31.86	**31.35**
Answer w. 0.7 (m=1)	53.78	49.44	29.76	29.10
Answer w. 0.7 (m=2)	55.37	50.00	29.62	29.95
Answer w. 0.7 (m=3)	54.93	50.31	31.58	30.75
Answer w. 0.7 (m=4)	55.11	50.51	29.20	31.11
Answer w. 0.7 (m=5)	54.93	50.28	31.77	31.31
Answer w. 0.6 (m=1)	53.78	49.44	29.76	29.10
Answer w. 0.6 (m=2)	54.98	50.02	29.76	30.02
Answer w. 0.6 (m=3)	55.34	50.27	32.28	30.55
Answer w. 0.6 (m=4)	54.78	50.53	30.29	30.92
Answer w. 0.6 (m=5)	55.34	50.23	32.14	31.05
Answer w. 0.5 (m=1)	53.78	49.44	29.76	29.10
Answer w. 0.5 (m=2)	55.30	49.98	31.42	30.15
Answer w. 0.5 (m=3)	55.39	50.23	**32.97**	30.17
Answer w. 0.5 (m=4)	55.10	50.42	31.73	29.92
Answer w. 0.5 (m=5)	55.39	50.16	32.50	29.92

Table 13: The utility judgments F1 performance (%) of different iteration stop conditions using Mistral on retrieval TREC and WebAP datasets (Numbers in parentheses represent m-values). Numbers in bold indicate the best performance.

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 14: The different precision score (%) of utility judgments using Mistral in different m values. “label” is the human-labeled label of the original dataset, i.e., [3]: Perfectly relevant; [2]: Highly relevant; [1]: Related; [0]: Irrelevant. Numbers in bold indicate the best performance.

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

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

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

Table 16: Statistics of experimental datasets.

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 14: 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 15: Instruction of the implicit answer generation.

G Parameters Comparison

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.

Question: what type of tissue are bronchioles.
First pseudo answer:
The bronchioles are tubes made of epithelial tissue and smooth muscle tissue. They are part of the respiratory system and carry air from the trachea into the lungs.
First utility judgment: My selection: [2, 3, 15].
Second pseudo answer:
The bronchioles are lined with epithelial tissue. Specifically, they are lined with ciliated cuboidal epithelium.
Second utility judgment: My selection: [3], [12], [13].
Third pseudo answer: Simple cuboidal epithelium.
Third utility judgment: My selection: [3], [12], [13].

Content and annotations of passages ([3]: Perfectly relevant [2]: Highly relevant [1]: Related [0]: Irrelevant**):**
(label:2)[2] What type of muscles control the size of the bronchioles in the lungs? The type of muscles that control the size of the bronchioles in the lungs are known as smooth muscles. Smooth muscles are found in both walls of the bronchioles and arterioles lo cated in both lungs. They happen to be the only kind of muscle found in the lungs. Full Answer >.
(label:3)[3] Simple cuboidal epithelium is a type of epithelial tissue found in the body. It is a single layer thick and made of cube-shaped cells. This type of tissue is found lining parts of the body such as the kidney tubules and walls of respiratory bronchioles.
(label:3)[12] Simple cuboidal epithelium is a type of tissue that is found lining parts of organs and ducts in the body. Its structure a llows for absorption and diffusion in those areas. Learn more about this tissue and quiz yourself at the end.t is a single layer thick a nd made of cubeshaped cells. This type of tissue is found lining parts of the body such as the kidney tubules and walls of respirator y bronchioles. The cells of simple cuboidal epithelium are specialized for diffusion and absorption. To unlock this lesson you must be a Study.com Member.
(label:3)[13] Cuboidal epithelium is a type of tissue in animal anatomy. It is a very important type of tissue in the human body. Cu boidal epithelia are found inside the kidneys, in parts of the eye and the thyroid, in the ovaries and testes, in parts of the brain and l ungs, and in the ducts of exocrine glands.Cuboidal epithelium is only one type of epithelial tissue in the body. Simple cuboidal epith elia are found in the bronchioles of the lungs. The simple cuboidal epithelium are cubeshaped cells that line various surfaces and c avities throughout the body, including the thyroid gland. Simple cuboidal epithelia line the nephron, or the basic structural unit of t he kidney.
(label:3)[15] A pulmonary lobule is the portion of the lung ventilated by one bronchiole. Bronchioles are approximately 1mm or le ss in diameter and their walls consist of ciliated cuboidal epithelium and a layer of smooth muscle. Bronchioles divide into even s maller bronchioles, called terminal, which are 0.5mm or less in diameter.he diameter of the bronchioles is often said to be less than 1 mm, though this value can actually range from 5 mm to 0.3 mm. As stated, these bronchioles do not have hyaline cartilage to ma intain their patency. Instead, they rely on elastic fibers attached to the surrounding lung tissue for support**.**

Figure 16: An example of our ITEM-A$_s$ using Mistral on the TREC dataset.

Question: when did family feud come out?
First pseudo answer: Family Feud has been on air since 1976.
First utility judgment: My selection:[1, 3, 13].
Second pseudo answer: The original Family Feud debuted in 1976.
Second utility judgment: My selection:[13].
Third pseudo answer: The Family Feud debuted in 1976.
Third utility judgment: My selection:[13].

Content and annotations of passages ([3]: Perfectly relevant [2]: Highly relevant [1]: Related [0]: Irrelevant):
(label:0)[1] What time is Celebrity Family Feud on tonight? Tonight's episode of Celebrity Family Feud airs on ABC at 8:00 PM EST. The episode usually runs around 60 minutes (including commercials). What day does Celebrity Family Feud come on? Make sure you have your TV ready every Sunday at 8:00 PM EST to catch new episodes. ABC airs Celebrity Family Feud episodes on S unday at 8:00 PM EST. You can also watch episodes on Amazon. You can also check out Celebrity Family Feud's official website and IMDb page. Celebrity Family Feud viewing details. Status: PENDING Current Season: 4 TV Channel: ABC Air Time: 8:00 P M EST Genre: Game Show Region: United States
(label:0)[3] When will Celebrity Family Feud Season 3 start? Looking for the premiere date for Celebrity Family Feud Season 3? We continuously monitor the official release date for Celebrity Family Feud Season 3. Stay tuned for the latest Celebrity Family Fe ud air-date information.
(label:3)[13] On your marks... let's start... the FAMILY FEUD! Family Feud is a Game Show from Mark Goodson Productions in which two families compete to guess the most popular answers to survey questions. Richard Dawson was the original host when the show debuted on ABC. The original version began in 1976, with a concurrent syndication run starting up a year later; both end ed in 1985 within a month of each other. Ray Combs was the host of the first Family Feud revival on CBS and in syndication starting in 1988.

Figure 17: An example of our ITEM-A$_s$ using Mistral on the TREC dataset.