Are Large Language Models Good at Utility Judgments?

Hengran Zhang
CAS Key Lab of Network Data Science and Technology, ICT, CAS
University of Chinese Academy of Sciences
Beijing, China
zhanghengran22z@ict.ac.cn

Ruqing Zhang
CAS Key Lab of Network Data Science and Technology, ICT, CAS
University of Chinese Academy of Sciences
Beijing, China
zhangruqing@ict.ac.cn

Jiafeng Guo*
CAS Key Lab of Network Data Science and Technology, ICT, CAS
University of Chinese Academy of Sciences
Beijing, China
guojiafeng@ict.ac.cn

Maarten de Rijke
University of Amsterdam
Amsterdam, The Netherlands
m.derijke@uva.nl

Yixing Fan
CAS Key Lab of Network Data Science and Technology, ICT, CAS
University of Chinese Academy of Sciences
Beijing, China
fanyixing@ict.ac.cn

Xueqi Cheng
CAS Key Lab of Network Data Science and Technology, ICT, CAS
University of Chinese Academy of Sciences
Beijing, China
cxq@ict.ac.cn

ABSTRACT

Retrieval-augmented generation (RAG) is considered to be a promising approach to alleviate the hallucination issue of large language models (LLMs), and it has received widespread attention from researchers recently. Due to the limitation in the semantic understanding of retrieval models, the success of RAG heavily lies on the ability of LLMs to identify passages with utility. Recent efforts have explored the ability of LLMs to assess the relevance of passages in retrieval, but there has been limited work on evaluating the utility of passages in supporting question answering.

In this work, we conduct a comprehensive study about the capabilities of large language models (LLMs) in utility evaluation for open-domain question answering (QA). Specifically, we introduce a benchmarking procedure and collection of candidate passages with different characteristics, facilitating a series of experiments with five representative LLMs. Our experiments reveal that: (i) well-instructed LLMs can distinguish between relevance and utility, and that LLMs are highly receptive to newly generated counterfactual passages. Moreover, (ii) we scrutinize key factors that affect utility judgments in the instruction design. And finally, (iii) to verify the efficacy of utility judgments in practical retrieval augmentation applications, we delve into LLMs' QA capabilities using the evidence judged with utility and direct dense retrieval results. (iv) We propose a k-sampling, listwise approach to reduce the dependency of LLMs on the sequence of input passages, thereby facilitating subsequent answer generation. We believe that the way we formalize and study the problem along with our findings contributes to a critical assessment of retrieval-augmented LLMs. Our code and benchmark can be found at https://github.com/ict-bigdatalab/utility_judgments.

*Jiafeng Guo is the corresponding author.

SIGIR '24, July 14–18, 2024, Washington, DC, USA

ACM ISBN 979-8-4007-0431-4/24/07
https://doi.org/10.1145/3626772.3657784

CCS CONCEPTS

• **Information systems** → **Question answering**; **Relevance assessment**; **Language models**.

KEYWORDS

Open-domain QA, Large language models, Utility judgments

ACM Reference Format:
Hengran Zhang, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Yixing Fan, and Xueqi Cheng. 2024. Are Large Language Models Good at Utility Judgments?. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24), July 14–18, 2024, Washington, DC, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3626772.3657784

1 INTRODUCTION

Retrieval-augmented generation (RAG) is considered a crucial means to effectively mitigate the hallucination issues in large language models (LLMs) [34, 36, 40, 46, 50], garnering widespread attention from researchers recently [12, 17, 39, 49]. Due to the semantic limitations in the retrieval model's understanding, the practical efficacy of RAG heavily relies on the LLMs's ability to accurately identify passages with utility among the retrieved candidate passages. Recent studies have explored the capabilities of LLMs in relevance judgments during retrieval [4, 32, 32, 41, 47, 55]. However, there has been limited focus on evaluating the utility judgment.

Relevance judgments via LLMs. Faggioli et al. [4] investigated the use of LLMs to automatically generate relevance judgments in information retrieval (IR). Many works have examined the zero-shot language understanding and reasoning capabilities of LLMs for relevance ranking, including pointwise [54, 54], pairwise [14, 33], and listwise [32, 41, 55] approaches. These publications have shown that LLMs excel in judging the relevance of passages to the query and achieve state-of-the-art ranking performance on standard benchmarks [33, 41, 55].

Utility judgments via LLMs. In this paper, we explore whether LLMs are good at judging the utility of passages. To appreciate the importance of (passage) utility, recall that LLMs find extensive use

[Question]: Why do **leaves** change **color**? (All evidence is relevant to the question.)

[Evidence-1]: During the arrival of autumn, **leaves** turn **red, orange, and yellow**.

[Evidence-2]: In the fall, sunlight decreases, and trees stop producing chlorophyll, **leading to the breakdown of chlorophyll and the exposure of other pigments**, resulting in colors like red, orange, and yellow in leaves. **(with utility)**

[Evidence-3]: Leaf color change is one of the beautiful sights in nature, attracting many tourists to visit in the autumn.

Figure 1: An example between utility and relevance.

in open-domain question answering (QA) [12, 36, 39, 49]. A common approach in open-domain QA involves retrieval-augmented LLMs [17, 47, 53], which (i) acquires a set of supporting evidence upon which to condition, and (ii) incorporates the selected evidence into the downstream LLM generation process. In open-domain QA, the goal of obtaining supporting evidence differs significantly from the goal of obtaining relevance judgments using LLMs: the supporting evidence should align with the judgments made by the LLM regarding which evidence qualifies as having *utility* for answering the question [52]. Utility and relevance are distinct concepts [20]: (i) *Relevance* signifies a connection between information and a context or question [37], and (ii) *utility* refers to the practical benefits of downstream tasks derived from consuming the information [38]. E.g., all evidence in Fig. 1 has different relevance to the question. However, only "[Evidence-2]" has utility in answering the question and other evidence, although connected to "leaf color change", lacks useful information on the underlying reasons for this phenomenon.

Research goal. The above observations naturally raise a question: *Are LLMs not only good at generating relevance judgments but also utility judgments?* To address this question, we undertake an empirical study in the setting of open-domain QA, investigating the capability of different LLMs in judging passage utility. Specifically, our study is broken down into three concrete research questions: (**RQ1**) *Can LLMs distinguish between utility and relevance?* (**RQ2**) *What factors affect the ability of LLMs to judge the utility of evidence?* (**RQ3**) *How do utility judgments impact the QA abilities of retrieval-augmented LLMs?*

Benchmarking procedure. In this work, we introduce the utility judgments task: *Given a question and a set of candidate passages, the utility judgments task is to identify supporting evidence with utility in answering the question.* We use several representative LLMs as zero-shot utility judges. As illustrated in Figure 2, we carefully design pointwise, pairwise, and listwise prompting approaches for LLM-based utility judges, as well as QA prompting approaches guiding LLM in answering questions using selected evidence.

To facilitate the study and evaluation of the utility judgments task, we formulate two hypotheses guiding the construction of novel benchmark datasets in Section 2.2: (i) *ground-truth inclusion*: the set of candidate passages must encompass ground-truth evidence. The ground-truth evidence offers the highest utility for question among the candidate passages; and (ii) *ground-truth uncertainty*: there exists uncertainty regarding the presence of ground-truth evidence in the set of candidate passages.

Our empirical work leads to the following interesting results:

- For **RQ1**: The answer is YES. LLMs can distinguish between utility and relevance given candidate passages. Specifically, utility judgments may offer more valuable guidance than relevance judgments to LLMs in identifying ground-truth evidence necessary for answering questions. Moreover, LLMs may exhibit a preference for selecting ground-truth evidence with utility when confronted with entity substitution-based counterfactual passages, compared to generated counterfactual passages.
- For **RQ2**: Different LLMs exhibit varying capabilities in utility judgment, with ChatGPT standing out as the most powerful. There is a consistent improvement in utility judgments performance the expansion of model scale. Listwises approaches may demonstrate superior performance compared to pointwise and pairwise approaches. In listwise approaches, LLMs are sensitive to the position of the ground-truth evidence in the input list. Moreover, the inclusion of chain-of-thought, reasoning process and answer generation also impact performance.
- For **RQ3**: Employing LLMs as zero-shot utility judges or relevance judges proves more advantageous for answer generation than directly using dense retrieval. The QA performance of LLMs is optimal when using evidence with utility judged by LLMs. To reduce the dependency of LLMs on the position of ground-truth evidence, we propose a k-sampling listwise approach that combines multiple utility judgments results to derive the final outcome, thereby facilitating subsequent answer generation. However, a significant gap still exists when compared to using only ground-truth evidence.

In Section 2, we provide a detailed description of the analysis setting. In Section 3–5, we address the three proposed research questions based on respective experimental results. Section 6 discusses related work and conclusions are drawn in Section 7.

2 PROBLEM STATEMENT

2.1 Task description

We introduce the utility judgments task, designed to assess the capabilities of LLMs to select supporting evidence with utility, which is useful for downstream answer generation. Formally, given a question q and a set of N retrieved passages $\mathcal{D} = \{d_1, d_2, \ldots, d_N\}$, the *utility judgments task* is to identify a subset $\mathcal{D}_u$ of $\mathcal{D}$ with passages with utility by prompting LLMs. We explore two evaluation scenarios based on the assumptions about $\mathcal{D}$:

- **Ground-truth inclusion (GTI)**: $\mathcal{D}$ should include ground-truth supporting evidence and other non-ground-truth passages. This assumption facilitates a direct assessment of the accuracy of selected evidence by LLMs with utility.
- **Ground-truth uncertainty (GTU)**: Taking into account the practical application of retrieval-augmented LLMs, $\mathcal{D}$ is directly obtained from the passage retriever without certainty regarding the presence of ground-truth evidence. Consequently, we evaluate the performance of LLMs in answering questions based on the selected evidence.

2.2 Benchmark construction

We introduce the source datasets and retrievers, and outline the construction processes for GTI and GTU.

2.2.1 Source datasets. We use three factoid QA (**FQA**) and non-factoid QA (**NFQA**) datasets as our source datasets.

In FQA datasets, answers are typically brief and concise facts, such as named entities [7]: (i) **Natural Questions (NQ)** [16] consists of real questions issued to the Google search engine. Each

question comes with an accompanied Wikipedia page with an annotated long answer (a paragraph) and a short answer (one or more entities). The long answers are denoted as ground-truth evidence, while the short answers are denoted as correct answers. (ii) **HotpotQA** [48] consists of QA pairs requiring multi-hop reasoning gathered via Amazon Mechanical Turk, each accompanied by a set of supporting evidence as ground-truth evidence.

In NFQA datasets, the answer to a non-factoid question is a chunk of one or more adjacent words [1]: **MSMARCO-QA** [26] is generated by sampling queries from Bing's search logs, consisting of annotated evidence that contains useful information for answering the questions and natural language answers. We use the evidence that is labeled as *is_selected*: 1 as our ground-truth. Following [36, 45], our experiments are conducted on the test set of NQ, and the development set of MSMARCO-QA and HotpotQA.

2.2.2 Retriever. We use two representative dense retrievers to gather supporting passages for the subsequent construction process of benchmark datasets. Specifically, we employ RocketQAv2 [35] and ADORE [51] for the FQA datasets (NQ and HotpotQA) and the NFQA dataset (MSMARCO-QA), respectively. Like [36, 39, 49], we assume that the size of $\mathcal{D}$, i.e., N, is 10, which falls within the input scope of the LLMs. In the following, we detail how to build $\mathcal{D}$, for GTI and GTU, respectively.

2.2.3 GTU benchmark. For three source datasets, we directly use the top 10 passages from the retrieved results for each question as the supporting evidence $\mathcal{D}$.

2.2.4 GTI benchmark. For three source datasets, the ground-truth evidence is introduced in Section 2.2.1. For non-ground-truth passages, we consider different characteristics as follows.

Counterfactual passages (CP). We construct synthetic passages that incorporate counter-answers, conflicting with correct answers. In order to make passages contain factual errors, for NQ and HotpotQA, we directly substitute the correct entities in the ground-truth evidence; for MSMARCO-QA, we instruct a LLM to directly generate a coherent passage that factually contradicts the correct entities in the ground-truth evidence.

- **Entity substitution method**. Both NQ and HotpotQA are FQA datasets derived from Wikipedia. Following [23], we employ an entity substitution method in the ground-truth evidence. Using the named entity recognizer SpaCy [9], we categorize all ground-truth answers into five types: person, date, numeric, organization, and location, creating an entity corpus for each dataset. For every correct answer a, we replace all instances of a in the ground-truth evidence with a different entity a' randomly selected from the entity corpus. We employ both Corpus Substitution and Type Swap Substitution as described in [23], where a and a' share the same entity type or have different entity types, respectively. This process is repeated five times for each substitution type, resulting in 10 candidate passages that are highly relevant but contain incorrect answers.
- **Generation-based method**. For MSMARCO-QA, the answers are sentence-level, manually crafted by human annotators with provided passages [26]. These answer sentences may be non-existent in the provided context. Therefore, using a hard entity substitution approach may not be suitable. Following [46], we propose employing a generative approach over the correct answers for MSMARCO-QA: (i) For each correct answer, we pick entities not mentioned in the question and randomly choose one for both *Corpus Substitution* and *Type Swap Substitution*, repeating each operation five times. This results in ten incorrect answers. (ii) Treating each incorrect answer as a claim, we employ DeBERTa-V2 [8] to identify contradictory claims. Only the claims contradicting the answer are retained. (iii) We input the retained claims into LLMs and direct LLMs to create realistic fake evidence supporting each claim. Here, we use ChatGPT with a temperature set to 0.7. The prompt is: "*Given a claim, please write a short piece of evidence to support it. The maximum length of the generated evidence is 100 words. You can fabricate content, but it should be as realistic as possible. Claim:* {*claim*} *Evidence:*" (iv) To verify that the evidence indeed supports the claim, we use an NLI model for support-checking. Specifically, the DeBERTa-V2 model determines whether the fake evidence supports the claim, and only the evidence supporting the claim is retained as the passages for subsequent experiments.

Highly relevant noisy passages (HRNP). We select original passages from the retrieved results that are highly relevant to the question but do not contain any information of the answer. For NQ and HotpotQA, we select 10 passages from top to bottom in the top 100 retrieval results of each question, that do not contain answer entities. For MSMARCO-QA, a human annotation label *is_selected* is included to indicate whether the passage, ranked among the top retrieved results by Bing, is used to answer the question. We choose passages labeled as 0 (indicating not selected as the supporting evidence) from the retrieval results. We regard the 10 passages organized in descending order of relevance as our HRNP.

Weakly relevant noisy passages (WRNP). We select passages from the retrieval results that are weakly relevant to the question and do not contain any information of the answer. For all datasets, we select 10 passages from bottom to top in the top 100 retrieval results of each question, that do not contain answer entities (excluding HRNP and ground-truth evidence). We regard the 10 passages organized in ascending order of relevance as our WRNP.

Constructing the candidate passages. For the final set $\mathcal{D}$ with a size of 10, besides the ground-truth evidence, the passages selection follows this procedure: If CP, HRNP, and WRNP can be evenly distributed, allocate them accordingly. If an even distribution is not feasible, distribute as evenly as possible initially, and then randomly assign the remaining passages. For example, with 2 ground-truth evidence, allocate 2 to CP, HRNP, and WRNP each. The remaining 2 passages are chosen randomly: for CP, we randomly sample passages from the candidates; for HRNP and WRNP, we select passages from top to bottom. The statistics are presented in Table 1.

Table 1: Dataset statistics.

	NQ	HotpotQA	MSMARCO-QA
#queries	1863	4407	3121
#passages	21M	21M	8.8M
#ground-truth evidence	1.0	2.4	1.1
#counterfactual passages	3.0	2.4	2.7
#highly relevant	3.0	2.6	3.1
#weakly relevant	3.0	2.6	3.1

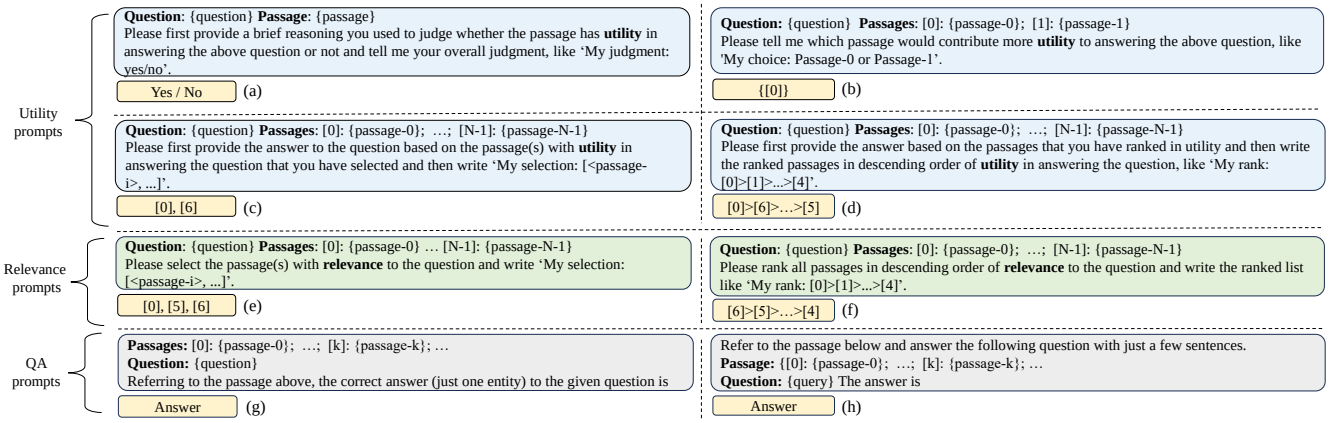

Figure 2: Prompts in blue blocks are utility prompts, where (a) is pointwise, (b) is pairwise, (c) is listwise-set, and (d) is listwise-rank. Prompts in green blocks are relevance prompts, where (e) is listwise-set and (f) is listwise-rank. Prompts in gray blocks are QA prompts, where (g) is designed for FQA datasets and (h) is designed for NFQA dataset.

2.3 Instructing LLMs with prompts

We introduce the LLMs used for evaluation and the prompts used for guiding LLMs. We design three types of prompts, i.e., utility prompts and relevance prompts for evidence selection, and QA prompts for answer generation; see Fig. 2.

LLMs for evaluation. We have selected several representative closed-source and open-source LLMs for our analysis: (i) Closed–source LLMs For the closed-source LLMs, we conduct our experiments using OpenAI's API [29], specifically gpt-3.5-turbo-1106 (abbreviated as ChatGPT). (ii) Open-source LLMs As for the open–source LLMs, our experiments involve four models: Llama2-7B-chat (abbreviated as Llama2-7B) [42], Llama2-13B-chat (abbreviated as Llama2-13B) [42], Vicuna-7B [2], and Vicuna-13B [2].

Utility prompts design. The goal of utility prompts is to guide LLMs to select evidence with utility in answering the question. According to different input forms, we consider three ways of presenting candidate passages as input to LLMs: (i) **Pointwise**: Each candidate passage $d \in \mathcal{D}$ is individually concatenated with the question q, and N such inputs are separately presented to LLMs. If the LLMs output *yes* (Fig. 2(a)), this indicates the passage's utility in addressing the question. (ii) **Pairwise**: Two candidate passages, $d_i \in \mathcal{D}$ and $d_j \in \mathcal{D}$, are concatenated with q. If LLMs output d_i (Fig. 2(b)), this suggests that LLMs find that d_i contributes more utility than d_j in answering the question. This process iterates $N(N-1)/2$ times, yielding the overall ranking of supporting evidence. (iii) **Listwise**: All candidate passages $\{d_1, d_2, \ldots, d_N\}$ in $\mathcal{D}$ are concatenated with q. Two output formats are designed: (i) the set of evidence with utility (**Listwise-set**, Fig. 2(c)), and (ii) the evidence list ranked by utility (**Listwise-rank**, Fig. 2(d)).

Relevance prompts design. The goal of relevance prompts is to guide LLMs to select evidence that are relevant to the question. Following [41], we only consider listwise input. Similar to utility prompts, we also account for two distinct outputs: listwise-set and listwise-rank, as elaborated in Fig. 2(e) and 2(f).

QA prompts design. For GTU, since the presence of ground-truth evidence is uncertain, directly assessing the chosen evidence by LLMs becomes challenging. Consequently, we design QA prompts to evaluate their QA abilities, which guide LLMs to obediently answer the questions based on evidence with utility or relevance. As depicted in Fig. 2(g) and 2(h), QA prompts are crafted for the FQA and NFQA datasets, respectively.

2.4 Evaluation metrics

GTI evaluation. We assess the performance of LLMs in selecting evidence for GTI, categorizing the output format into two types: (i) **The evidence set**. For pointwise and listwise-set approaches, the final output of LLMs is a set of evidence based on utility or relevance. We employ **Precision** (**P**), **Recall** (**R**) and **F1**. (ii) **The evidence list**. For pairwise and listwise-rank approaches, the output of LLMs is a ranked list based on utility or relevance. We use widely-used evaluation metrics in IR, i.e., mean reciprocal rank (**MRR**) [3] and normalized discounted cumulative gain (**NDCG**) [13].

GTU evaluation. We consider the final answering performance based on selected evidence by LLMs for GTU. Following [11, 36], we use the exact match (**EM**) score and **F1** score to evaluate the answer performance of LLMs on the FQA datasets. Following [26], we use **ROUGE-L** [19] and **BLEU** [30] to evaluate the answer performance of LLMs on the NFQA dataset.

3 RELEVANCE VS. UTILITY

We use ChatGPT as an example to investigate whether LLMs can distinguish between utility and relevance (*RQ1*).

Experimental setup. We evaluate ChatGPT [29] using the NQ, HotpotQA, and MSMARCO-QA datasets incorporated into the GTI benchmark. We employ the listwise forms for both relevance and utility judgments, including listwise-set in Fig. 2(c) and 2(e), and listwise-rank in Fig. 2(d) and 2(f). For each question, with 10 candidate passages, we shuffle them as the input and the input passages for relevance judgments and utility judgments remain the same.

LLMs can distinguish between utility and relevance. Table 2 presents a comparative analysis of ChatGPT's performance in relevance and utility judgments across three datasets. Utility-based prompts prove more effective in assisting LLMs in identifying ground-truth evidence compared to relevance-based prompts, particularly in NQ dataset. E.g., in NQ, the F1 and NDCG@1 scores for

	NQ	MSMARCO-QA
Question	(ID: 3514) When did the **right to buy** scheme start?	(ID: 138793) Definition of the **paleo diet.**
Ground-truth answer	1980	A modern diet designed to emulate the diet of wild animals and plants eaten by humans during the Paleolithic era.
Utility judgement	**Passage-9 :** After Margaret Thatcher became Prime Minister in May 1979, the legislation to implement the Right to Buy was passed in the Housing Act 1980.	**Passage-3:** A paleo diet, or paleolithic diet, is a modern diet designed to emulate the diet of wild animals and plants eaten by humans during the Paleolithic era, or as far as this is possible in …
Answer generation based on selected passages with utility	1980	The paleo diet, also known as the paleolithic diet, is a modern diet that aims to replicate the food consumption of humans during the Paleolithic era, focusing on wild animals and plants. It is often referred to as a Stone Age, hunter-gatherer, or caveman diet.
Relevance judgement	**Passage-1 (Counterfactual passage)**: …the legislation to implement the **Right to Buy** was passed in the Housing Act Gibraltar. **Passage-3 (Highly relevant noisy passage)**: **Buy council housing** a Conservative Party policy nationally. The **policy** was largely in place for the 1974 Conservative manifesto, but did not prove an asset in… **Passage-9 (Ground-truth evidence)**: After Margaret Thatcher became Prime Minister in May 1979, the legislation to implement the **Right to Buy** was passed in the Housing Act **1980**.	**Passage-3 (Ground-truth evidence)**: A **paleo diet**, or paleolithic diet, is a modern diet designed to emulate the diet of wild animals and plants eaten by humans during the Paleolithic era, or … **Passage-6 (Highly relevant noisy passage)**: The definition of a **diet** is the kinds of food a person, animal or community habitually eats, By definition, the paleo diet should be known … **Passage-7 (Weakly relevant noisy passage)**: Eat like a caveman and shed pounds. That's the theory behind the **Paleo Diet**. Loren Cordain, PhD, who literally wrote the book on the … **Passage-9 (Counterfactual passage)**: …the **diet** primarily consisted of locally sourced wild game and foraged plants, emphasizing emphasizing a high protein and low carbohydrate …
Answer generation based on selected passages with relevance	1979	The definition of the paleo diet is a modern diet that aims to emulate the diet of wild animals and plants eaten by humans during the Paleolithic era, focusing on lean meats, fish, fruits, vegetables, and nuts while avoiding processed foods and grains. It is also known as the Stone Age, hunter-gatherer, or caveman diet.

Figure 3: Given 10 candidate passages, ChatGPT is employed to select evidence with utility and relevance respectively using listwise-set approaches. The selected evidence are then used by ChatGPT to answer questions. In the examples from NQ and MSMARCO-QA, "Passage-9" and "Passage-3" respectively denote the ground-truth supporting evidence. The full set of 10 candidate passages for each question can be accessed at https://github.com/ict-bigdatalab/utility_judgments.

Table 2: The performance (%) of utility judgments and relevance judgments using ChatGPT under the listwise approaches on the GTI benchmark.

Dataset	Judgment	Listwise-set			Listwise-rank		
		P	R	F1	NDCG		MRR
					@1	@5	@5
NQ	Relevance	36.65	73.75	48.97	39.08	67.62	59.42
	Utility	57.19	73.00	64.14	57.80	77.43	71.87
HotpotQA	Relevance	70.02	47.45	56.56	74.54	79.67	85.75
	Utility	76.83	46.54	57.97	78.28	80.29	87.52
MSMARCO-QA	Relevance	32.80	85.04	46.87	40.82	66.07	60.17
	Utility	36.77	63.50	46.57	40.07	64.90	60.35

utility judgments exhibit a notable increase of 30.98% and 47.90%, respectively, compared to relevance judgments.

LLMs exhibit distinct performance on multi-hop and single-passage QA datasets. (i) In the listwise-rank approach, the performance of utility judgments and relevance judgments is superior on the multi-hop dataset, i.e., HotpotQA, compared to the single-passage QA dataset, e.g., NQ. This could be attributed to the presence of multiple pieces of ground-truth evidence for each question in HotpotQA, leading to a higher probability of the ground-truth evidence appearing at the top of the ranked list. (ii) In the listwise-set approach, the performance of relevance judgments on HotpotQA surpasses that on NQ in terms of F1, likely due to the increased probability of the set containing ground-truth in HotpotQA. However, concerning utility judgments, the F1 score for HotpotQA is lower than for NQ. This discrepancy may arise from the LLMs's capability to address multi-hop questions, where they may not recall all necessary evidence required at each step, consequently impacting their judgment, particularly when selecting precise sets.

LLMs are highly receptive to generated counterfactual passages. (i) In the listwise-set and listwise-rank scenarios, ChatGPT's performance on MSMARCO-QA is significantly worse than on NQ in terms of utility judgments. This disparity may stem from the construction of counterfactual passages [46]. In NQ, counterfactual passages are built through entity substitution, potentially leading to passage incoherence. LLMs might be sensitive to incoherent passages, resulting in their rejection during utility judgments. Conversely, the construction of counterfactual passages in MSMARCO-QA involves LLMs generating coherent passages, which may confuse the utility judgments of LLMs. (ii) However, for relevance judgments the performance gap between MSMARCO-QA and NQ is small. Both counterfactual passages and entity substitution are highly relevant to the question, so the performance of selected passages remains low regardless of different construction approaches.

Case study. Fig. 3 illustrates two examples based on utility and relevance judgments. The evidence selected by LLMs based on utility is more precise than that selected based on relevance. When generating answers using utility judgments results and relevance judgments results, respectively, it becomes evident that the noise or misinformation in the relevance results significantly impacts the LLMs's answer generation. E.g., the "Passage-9" obtained from relevance judgments contains misinformation about the focus of paleo diet, which misguides the answer generator's understanding of the paleo diet. This underscores the importance of enhancing the quality of supporting evidence for answer generation.

4 UTILITY JUDGMENTS DEPEND ON INSTRUCTION DESIGN

Our analysis suggests that utility judgments may offer more effective guidance to LLMs in identifying ground-truth evidence for answering questions. In light of this, we extend our analysis to explore how different factors affect utility judgments (*RQ2*). Specifically, we examine key factors in instruction design, including the input form of passages (i.e., pointwise, pairwise, and listwise), the sequence of input between the question and passages, and additional requirements (i.e., chain-of-thought, reasoning, and providing answers). We evaluate ChatGPT, Llama2-7B, Llama2-13B, Vicuna-7B, and Vicuna-13B, on the NQ, HotpotQA, and MSMARCO-QA datasets incorporated into the GTI benchmark.

Table 3: The performance (%) of different LLMs in utility judgments under different datasets and input forms. Bold indicates the best performance among the same type of input.

Pointwise / Listwise-set

Model	NQ			HotpotQA			MSMARCO-QA		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
ChatGPT	**22.40** / **57.19**	92.97 / **73.00**	**41.00** / **64.14**	**46.04** / **76.83**	46.07 / 46.54	46.05 / **57.97**	**25.26** / **36.77**	73.82 / **63.50**	**37.64** / **46.57**
LlaMa2-7B	18.25 / 15.46	81.27 / 39.56	29.80 / 22.23	33.97 / 49.13	48.59 / 35.83	39.98 / 41.44	14.41 / 21.84	89.78 / 41.40	24.83 / 28.59
LlaMa2-13B	17.69 / 27.56	36.71 / 38.43	23.88 / 32.10	26.71 / 61.49	20.68 / 36.80	23.31 / 46.04	16.23 / 23.15	48.40 / 27.45	24.31 / 25.12
Vicuna-7B	13.39 / 18.67	**100.00** / 45.14	23.62 / 26.42	37.27 / 44.46	**89.76** / 45.77	**52.67** / 45.10	13.23 / 18.41	**96.87** / 35.27	23.29 / 24.19
Vicuna-13B	15.02 / 35.24	86.96 / 66.02	25.62 / 45.95	36.75 / 60.23	76.77 / **46.72**	49.70 / 52.62	14.39 / 24.62	85.91 / 53.84	24.66 / 33.79

Pairwise / Listwise-rank

Model	NQ			HotpotQA			MSMARCO-QA		
	NDCG@1	NDCG@5	MRR@5	NDCG@1	NDCG@5	MRR@5	NDCG@1	NDCG@5	MRR@5
ChatGPT	**57.11** / **57.80**	**78.23** / **77.43**	**72.18** / **71.87**	**74.67** / **78.28**	**75.11** / **80.29**	**84.94** / **87.52**	**29.89** / **40.07**	**57.56** / **64.90**	**50.63** / **60.35**
LlaMa2-7B	13.53 / 7.66	34.99 / 23.12	27.69 / 17.86	29.18 / 24.03	42.33 / 32.85	49.21 / 38.99	11.49 / 4.69	32.85 / 11.48	26.76 / 9.45
LlaMa2-13B	16.96 / 6.93	39.03 / 12.52	31.66 / 10.95	31.02 / 20.10	42.63 / 20.22	50.34 / 24.78	14.64 / 6.89	35.20 / 14.07	29.49 / 12.21
Vicuna-7B	10.03 / 10.22	30.18 / 29.47	23.37 / 23.12	26.11 / 27.66	39.49 / 37.75	46.24 / 45.79	11.22 / 12.14	30.59 / 29.93	24.97 / 25.04
Vicuna-13B	12.37 / 34.93	33.00 / 57.79	25.93 / 51.59	29.45 / 67.62	41.30 / 66.36	48.71 / 78.85	12.27 / 22.91	31.46 / 44.45	25.91 / 38.87

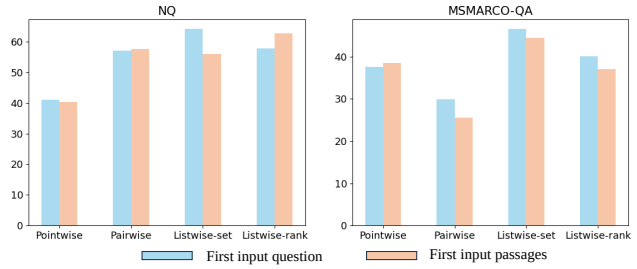

Figure 4: Performance of ChatGPT with different input forms on the NQ and MSMARCO-QA datasets in different sequences of input between the question and passages. We use "F1" score on pointwise and listwise-set forms and "NDCG@1" score on pairwise and listwise-rank forms. "NDCG@5" has same trend with 'NDCG@1".

Different input forms have varying impacts on utility judgments. Table 3 shows the performance of pointwise, pairwise, and listwise inputs. We directly use the prompt in Fig. 2 and the position of ground-truth evidence is random in the input passage list.

Pointwise. LLMs show high recall but low precision, leading to low F1 across all datasets. Analyzing LLMs outputs reveals a tendency, except for Llama2-13B, to frequently output "yes."

Pairwise. For the same LLMs, the performance of utility judgments is better on the NQ and HotpotQA datasets. There is room for improvement in the ability to perform utility judgments on the constructed MSMARCO-QA dataset. The reasons could be twofold. Firstly, the MSMARCO-QA dataset includes numerous non-factual questions and might impact utility judgments capabilities of the LLMs compared to NQ. Secondly, the input passages contain smoothly generated counterfactual passages, which could potentially confuse the utility judgments capabilities of the LLMs.

Listwise. There are two forms, i.e., listwise-set and listwise-rank in the listwise forms. (i) In the listwise-set form, The utility judgments of LLMs, particularly ChatGPT, excels on the NQ dataset compared to HotpotQA and MSMARCO-QA datasets under listwise-set form. The reason may be that the NQ dataset comprises relatively simple factual questions, where LLMs demonstrate a superior ability to generate answers, positively influencing utility judgments. (ii) In the listwise-rank form, all LLMs excel on HotpotQA compared to other datasets in ranking scores, possibly due to multiple pieces of ground-truth evidence per question, increasing the probability of ground-truth evidence ranking higher.

Overall analysis. (i) ChatGPT outperforms other LLMs, highlighting the challenges of open-source models in zero-shot utility judgments. (ii) For LLMs of the same family, utility judgments generally improve as the scale increases. For instance, Vicuna-13B achieves a 73.92% F1 improvement over Vicuna-7B when using listwise-set input on the NQ dataset. (iii) Except for Vicuna-13B and ChatGPT, LLMs exhibit superior utility judgments in pairwise form compared to listwise form. LLMs demonstrate better utility judgments in listwise form than in pointwise form.

The sequence of inputs between the question and passages has important effects on utility judgments. In this analysis, we use optimal prompts from Fig. 2 with different sequences of input between the question and passages. Fig. 4 shows how the sequence of questions and passages affects ChatGPT's utility judgments, with differing effects seen across datasets and input forms. For instance, in the listwise-rank input, ChatGPT prioritizes passages first in the NQ dataset but favors questions first in the MSMARCO-QA dataset. In the NQ dataset, ChatGPT prioritizes questions first when using the listwise-set form, but favors passages first with the listwise-rank form. To ensure consistent prompt design, our future experiments will adopt the question-first input sequence, as shown in Fig. 2.

LLMs demonstrate sensitivity to the order of ground-truth evidence in the listwise input. For the experimental setting in this analysis, (i) We directly use the optimal prompt in Fig. 2. (ii) The position of ground-truth evidence is fixed in the input passage list. Prior research has demonstrated a propensity in retrieval-augmented LLMs [36] to prioritize evidence presented in the top position [31], and has highlighted the order sensitivity in LLMs [24]. Therefore, as shown in Fig. 5, to analyze whether LLMs exhibit sensitivity to order in utility judgments, we position the ground-truth evidence at different positions under listwise-set and listwise-rank

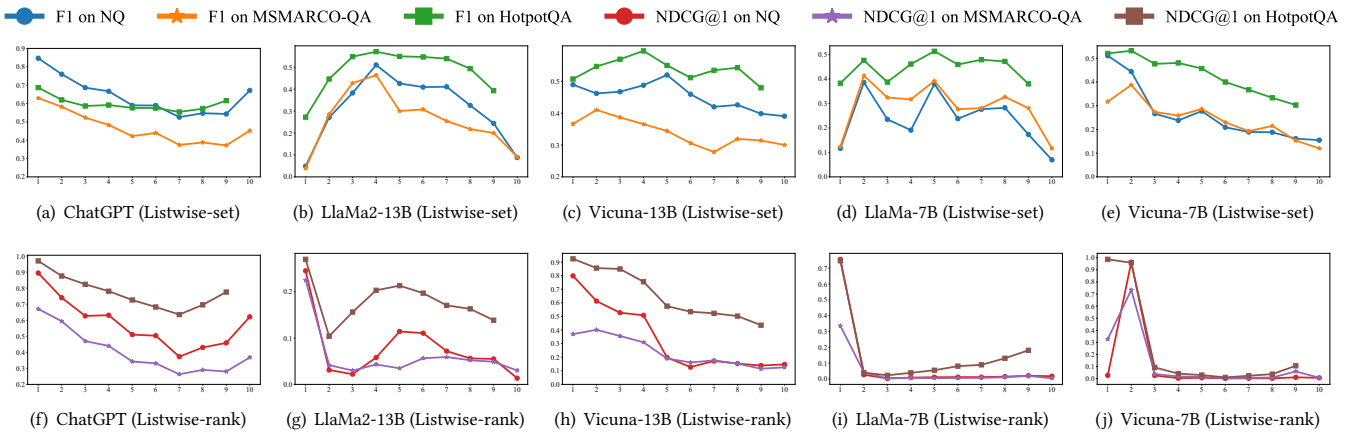

Figure 5: The performance of five different LLMs in utility judgments based on the positions of ground-truth evidence in the input list across listwise-set and listwise-rank forms on different datasets.

inputs. We observe that: (i) All LLMs exhibit a notable sensitivity to the position of ground-truth evidence, showcasing significant fluctuations across different positions. (ii) For the listwise-set form, different LLMs have different sensitivity to ground-truth evidence positions in the input list. The performance of utility judgments for ChatGPT on three datasets first decreases and then increases as the ground-truth evidence position is lower in the input list. For LlaMa2-13B, the performance of utility judgments first increases and then decreases as the ground-truth evidence position is lower in the input list on three datasets. (iii) For the listwise-rank input form, LLMs of the same family with different scales have very different performances in utility judgments capabilities under the listwise-rank form, except for ChatGPT. Vicuna-13B shows a gradual decline in utility judgment performance as ground-truth evidence is positioned further towards the end. However, Vicuna-7B performs well only at the specific ground-truth evidence position, displaying almost zero performance at other positions. This indicates that models with smaller parameter scales have significantly poor utility judgment capabilities in the listwise-rank form. (iv) The sensitivity of a given LLM to the position of ground-truth evidence in the input list can vary across different forms. For instance, Vicuna-13B excels with evidence in the middle for listwise-set input but performs better with evidence at the beginning for listwise-rank input.

In practical retrieval augmentation, information about ground-truth evidence positions is often lacking, and the input may not even contain such evidence. Addressing the sensitivity of LLMs to ground-truth evidence positions is crucial and requires immediate attention. To mitigate this, we propose a simple k-sampling approach in Section 5, i.e., shuffling the input passages list multiple times and performing utility judgments task. This aims to reduce LLMs' reliance on specific ground-truth evidence positions.

Utility judgments also depend on additional requirements. For the experimental setting in this analysis, (i) We directly use the optimal prompt in Fig. 2. (ii) The position of ground-truth evidence is random in the input passage list. (iii) Due to the excessively large number of pairwise input instances, in order to reduce the frequency of API calls, we randomly selected 200 questions from the NQ dataset for pairwise testing. We consider three additional requirements in the instructions, i.e., (i) **Chain-of-Thought (COT)** [15, 44] has been proven to be useful for LLMs in handling complex problems. We incorporate guidance from Zero-shot-CoT [15], i.e., simply adding *"Let's think step by step"* before giving utility judgments. (ii) **Reasoning (RA)**: Inspired by COT, many NLP tasks have empirically demonstrated performance improvements in LLMs when reasoning is incorporated into prompts [18]. We also design reasoning requirements in prompts like *provide a brief reasoning* before giving the output. (iii) **Answer**: We further guide LLMs in confirming their utility judgments by having it *"provide the answer to the question"* to the question before giving the output.

Fig. 6 shows the performance of utility judgments using ChatGPT on the NQ dataset. We observe that: (i) For pointwise form, incorporating COT, RA, and answer requirements enhances F1 performance compared to scenarios lacking additional requirement, possibly due to the challenge posed by limited input information, especially with only one passage available for LLMs to assess utility directly. (ii) However, when applied to listwise-set inputs with reasoning and listwise-rank inputs with COT, there is a 11.80% decrease in F1 and a 3.01% reduction in NDCG@1 compared to no requirement, respectively. This is likely due to the potential influence of noise or incorrect information in passages of varying quality, impacting the reasoning process and overall judgment capability. (iii) Incorporating answer requirements significantly boosts ChatGPT's ability to judge utility in all input forms by implicitly defining passage utility through provided answers. (iv) However, for pairwise inputs, the requirements do not help ChatGPT. E.g., after using the reasoning requirement, the performance of ChatGPT is decreased by 9.37% in terms of NDCG@1. The reason might be that ChatGPT already has strong pairwise preference judgment capabilities, as evidenced in previous work [14].

5 RETRIEVAL-AUGMENTED LLMs: USING SELECTED EVIDENCE FOR QA

The open-domain QA task with LLMs concerns the process of retrieving external knowledge as evidence and subsequently using the LLMs to answer questions based on the evidence. How do the utility judgments impact the QA abilities of retrieval-augmented LLMs (*RQ3*)? Specifically, the procedure begins with the retrieval of passages by dense retrievers, without certainty regarding the

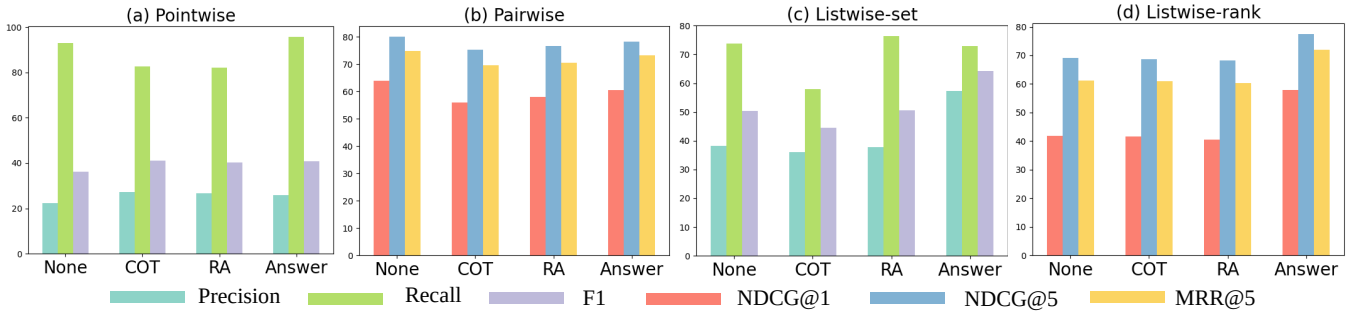

Figure 6: The performance (%) of utility judgments using ChatGPT under different forms, i.e., pointwise, pairwise, listwise-rank and listwise-set, on the NQ dataset using the prompts with different requirements.

presence of ground-truth evidence (the GTU setting). Then, the utility of the retrieved passages is evaluated by LLMs, and finally, LLMs are guided to provide answers based on the selected passages.

Experimental setup. We evaluate ChatGPT and Vicuna-13B on the NQ and MSMARCO-QA datasets incorporated into the GTU benchmark. We employ various types of knowledge as evidence for answer generation: (i) No evidence input (**None**); (ii) Directly using the top-10 retrieval results as evidence, i.e., candidate passages in the GTU benchmark (**Dense**); (iii) Ground-truth evidence (**Ground-truth**); (iv) Passages with **relevance**: using GTU's candidate passages for LLMs in relevance judgments using the optimal prompt from Fig. 2 under listwise forms; and (v) Passages with **utility**: using GTU's candidate passages for LLMs in utility judgments using the optimal prompt from Fig. 2 under all forms. To ensure fairness, the number of results obtained in the form of sets and in the form of a ranked list, after selecting candidate passages, is the same when used in answer generation.

Different sources of evidence improve the performance of answer generation to different extents. From Table 4, we can observe that (i) Using external evidence from a dense retriever markedly improves LLMs's answer generation compared to not using external evidence, emphasizing the crucial role of retrieval enhancement in open-domain QA tasks. (ii) Both utility judgments and relevance judgments enhance LLMs's answer generation performance, demonstrating that employing either relevance or utility can effectively filter input passages as evidence. (iii) Using utility judgments for evidence yields better performance in enhancing answer generation compared to using evidence based on relevance judgments in the same input form, further reflecting that LLMs can distinguish relevance and utility, and then select evidence that has more utility in answering questions. (iv) The listwise-set input outperforms other input forms in answer generation on the MSMARCO-QA dataset using both LLMs. Meanwhile, the pairwise input achieves the highest performance on the NQ dataset using ChatGPT among all utility judgments approaches, but its real-life implementation involves prohibitively high input costs. (v) Overall, the degree to which evidence from different sources improves the performance of answer generation is as follows: *Ground-truth* > *Utility judgments* > *Relevance judgments*> *Dense* > *None*.

Novel k-sampling listwise approach. According to the conclusions in Section 4, LLMs exhibit sensitivity to the position of ground-truth evidence in listwise inputs when judging utility. We propose a k-sampling listwise approach (we only use the listwise-set input form as an example). Specifically, we randomize the input passage list k times, conduct utility judgments for each iteration, and then aggregate the results through voting. The evidence chosen for answer generation is determined by the highest vote count. For each query, the number of evidence for answer generation is based on the most frequently occurring listwise-set result across the k iterations. From Table 4, we can observe that the performance using the k-sampling method, such as 10-sampling, improves answer generation on the NQ dataset by 2.84% in terms of F1 compared to not using sampling in the listwise-set input form. Moreover, the k-sampling method demonstrated superior answer generation performance, surpassing the usage of ground-truth evidence, in the two LLMs of the MSMARCO-QA dataset. The performance improvement indicates that the use of k-sampling effectively mitigates the LLMs's dependence on the position of ground-truth evidence.

6 RELATED WORK

LLMs for relevance judgments. With exhibited unprecedented proficiency in language understanding, large language models (LLMs) such as ChatGPT [29] and Llama 2 [42] have seen widespread applications across various tasks [10, 12, 27, 43]. IR is a representative work of LLMs applications, with many studies incorporating LLMs into relevance ranking [21, 22, 33, 55]. Research into LLMs in relevance ranking mainly contains the following three approaches: (i) pointwise [28, 54], (ii) pairwise [14, 33], and (iii) listwise [32, 41, 55]. Zhuang et al. [54] employed LLMs in scoring fine-grained pointwise relevance labels. Jiang et al. [14] and Qin et al. [33] employed a pairwise relevance comparison method to distinguish differences between candidate outputs. Previous works [32, 41, 55] analyzed the capabilities of LLMs in the relevance ranking task.

Faggioli et al. [4] demonstrated LLMs' proficiency in relevance assessment in IR. However, relevance in IR and utility in answering specific questions are distinct concepts. This paper investigates whether LLMs excel in judging the utility of retrieved passages. Similar to relevance ranking tasks, we devise pointwise, pairwise, and listwise approaches for utility judgments.

Retrieval-augmented LLMs for QA. The application of LLMs in QA [12, 36, 39, 49] is mainly retrieval-augmented LLMs [5, 6, 34, 36, 50]. Current researches on retrieval-augmented LLMs can be categorized into two main groups, i.e., independent architectures [25, 46, 49] and joint architectures [12, 20, 39, 52].

Table 4: Performance (%) of question answering using different evidence and different LLMs. Bold indicates the best answer generation performance among different methods for evidence other than using ground-truth evidence.

Evidence	ChatGPT							Vicuna-13B						
	NQ		MSMARCO-QA					NQ		MSMARCO-QA				
	EM	F1	ROUGE-L	BLEU-1	BLEU-2	BLEU-3	BLEU-4	EM	F1	ROUGE-L	BLEU-1	BLEU-2	BLEU-3	BLEU-4
None	42.49	54.55	29.78	22.64	13.63	9.10	6.41	12.40	25.09	27.41	18.34	10.82	7.09	4.91
Dense	46.54	57.00	35.07	25.58	17.48	13.15	10.39	21.52	36.84	29.69	17.14	11.83	8.93	7.11
Ground-truth	66.40	76.86	51.07	40.78	33.46	28.52	24.73	34.73	52.19	48.95	36.00	29.72	25.47	22.25
Relevance judgments														
Listwise-set	47.29	57.30	35.11	25.66	17.46	13.07	10.30	21.47	36.26	30.55	18.49	12.70	9.56	7.58
Listwise-rank	47.07	57.14	35.41	25.81	17.65	13.27	10.46	21.20	36.37	30.45	18.38	12.60	9.48	7.51
Utility judgments														
Pointwise	46.16	56.59	34.51	25.41	17.18	12.84	10.12	20.61	36.58	30.26	17.82	12.33	9.34	7.46
Pairwise	**49.97**	**62.06**	34.86	25.82	17.37	12.90	10.08	23.24	38.31	30.98	19.64	13.35	10.00	7.91
Listwise-set	47.72	58.01	35.68	26.52	18.15	13.68	10.85	24.10	39.07	31.00	19.04	12.92	9.68	7.69
Listwise-rank	48.63	58.76	35.62	26.55	18.12	13.66	10.84	23.40	37.86	30.71	19.68	13.29	9.94	7.86
5-sampling	48.90	58.97	35.97	26.83	18.31	13.78	10.90	24.91	40.10	31.28	19.30	13.15	9.86	7.81
10-sampling	49.49	59.66	**36.00**	**26.85**	**18.33**	**13.81**	**10.94**	**25.39**	**40.56**	**31.59**	**19.87**	**13.50**	**10.11**	**8.00**

In independent architectures, the retriever and LLMs operate independently, with the retriever's sole role being to provide relevant external knowledge to the LLMs [52]. For example, Yu et al. [49] demonstrated that using retrieval-augmented methods can improve GPT-3 performance on open-domain question answering. However, these retrieval models are usually based on the probability ranking principle (PRP) [52], ranking passages based on their likelihood of being relevant to the question [50, 52], which may not align with a retrieval-augmented framework. In the joint architecture, the LLMs actively engage in the training process of the retriever [12, 17, 39, 52]. Shi et al. [39] used the performance of the LLMs in answer generation as feedback to train the retriever to retrieve the evidence that contribute more utility to answering the question.

The independent retriever may struggle to align well with the utility requirements of LLMs on the retrieval passages. Although joint architecture partially alleviates this issue, depending on the answers outputted by LLMs as utility judgments for retrieved passages is influenced by the LLMs' internal knowledge. Since LLMs may produce different answers for the same input passages, assessing passage utility based solely on the quality of LLMs' answers in joint architecture may not always accurately reflect the passages' inherent utility for answering questions. Therefore, we directly investigate the LLMs's capability of utility judgment. We hope our work provides useful insights for understanding and improving retrieval-augmented LLMs in the future.

7 CONCLUSION

We have studied the abilities of LLMs to produce utility judgments for passages. We have found that LLMs have different understandings of utility and relevance. Moreover, we have shown that utility judgments of LLMs are influenced by the input forms and positions of ground-truth evidence in the input list, none of which may be a desired property for retrieval-augmented LLMs. The susceptibility of LLMs to these external factors could stem from a limited instruction-following capability. We anticipate that as LLMs continue to advance, the influence of these factors on their capabilities will gradually diminish. Finally, we have found that using utility judgments can further improve the performance of answer generation compared to relevance judgments.

As a preliminary exploration into utility judgments within LLMs, our analysis has solely focused on evaluating the utility of a small set of candidate passages. In the future, it is imperative to devise methodologies for assessing the utility of large-scale candidate passages within the LLMs. This is essential for enhancing utility judgments capabilities in practical applications of retrieval-augmented LLMs. Furthermore, we have only scratched the surface in exploring the zero-shot utility judgments of LLMs. It is crucial to investigate additional scenarios, e.g., the few-shot scenario, to further uncover the capabilities of LLMs in utility judgments. We hope our work provides a solid evaluation testbed and meaningful insights for understanding, improving, and deploying utility judgments by LLMs in the future. We envision a future where an increasing number of research endeavors contribute to the field of utility judgments in LLMs.

ACKNOWLEDGMENTS

This work was funded by the Strategic Priority Research Program of the CAS under Grants No. XDB0680102, the National Key Research and Development Program of China under Grants No. 2023YFA1011602 and 2021QY1701, the National Natural Science Foundation of China (NSFC) under Grants No. 62372431, the Youth Innovation Promotion Association CAS under Grants No. 2021100, the Lenovo-CAS Joint Lab Youth Scientist Project, and the project under Grants No. JCKY2022130C039. This work was also (partially) funded by the Dutch Research Council (NWO), under project numbers 024.004.022, NWA.1389.20.183, and KICH3.LTP.20.006, and the European Union's Horizon Europe program under grant agreement No. 101070212. All content represents the opinion of the authors,

which is not necessarily shared or endorsed by their respective employers and/or sponsors.

REFERENCES

[1] Ahmad Aghaebrahimian. 2018. Linguistically-based Deep Unstructured Question Answering. *CoNLL*, 433–443.

[2] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality. https://vicuna.lmsys.org (Accessed 14 April 2023).

[3] Nick Craswell. 2009. Mean Reciprocal Rank. *Encyclopedia of Database Systems* 1703 (2009).

[4] 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 *SIGIR*. 39–50.

[5] Run-Ze Fan, Yixing Fan, Jiangui Chen, Jiafeng Guo, Ruqing Zhang, and Xueqi Cheng. 2024. RIGHT: Retrieval-Augmented Generation for Mainstream Hashtag Recommendation. In *European Conference on Information Retrieval*. Springer, 39–55.

[6] Run-Ze Fan, Xuefeng Li, Haoyang Zou, Junlong Li, Shwai He, Ethan Chern, Jiewen Hu, and Pengfei Liu. 2024. Reformatted Alignment. *arXiv preprint arXiv:2402.12219* (2024). https://arxiv.org/abs/2402.12219

[7] Helia Hashemi, Mohammad Aliannejadi, Hamed Zamani, and W Bruce Croft. 2020. ANTIQUE: A Non-factoid Question Answering Benchmark. In *ECIR 2020*. Springer, 166–173.

[8] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. DeBERTa: Decoding-enhanced BERT with Disentangled Attention. In *ICLR*.

[9] Matthew Honnibal. 2017. spaCy. (2017). https://spacy.io/

[10] Yupeng Hou, Junjie Zhang, Zihan Lin, Hongyu Lu, Ruobing Xie, Julian McAuley, and Wayne Xin Zhao. 2023. Large Language Models are Zero-shot Rankers for Recommender Systems. *arXiv preprint arXiv:2305.08845* (2023).

[11] Gautier Izacard and Edouard Grave. 2021. Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering. *EACL* (2021).

[12] Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2022. Few-shot Learning with Retrieval Augmented Language Models. *arXiv preprint arXiv:2208.03299* (2022).

[13] Kalervo Järvelin and Jaana Kekäläinen. 2017. IR Evaluation Methods for Retrieving Highly Relevant Focuments. In *SIGIR*, Vol. 51. 243–250.

[14] Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. 2023. LLM-Blender: Ensembling Large Language Models with Pairwise Ranking and Generative Fusion. *ACL* (2023).

[15] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large Language Models are Zero-shot Reasoners. *NeurIPS* 35 (2022), 22199–22213.

[16] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural Questions: A Benchmark for Question Answering Research. *TACL* 7 (2019), 453–466.

[17] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *NeurIPS* 33 (2020), 9459–9474.

[18] Xian Li, Ping Yu, Chunting Zhou, Timo Schick, Luke Zettlemoyer, Omer Levy, Jason Weston, and Mike Lewis. 2023. Self-alignment with instruction backtranslation. *arXiv preprint arXiv:2308.06259* (2023).

[19] Chin-Yew Lin. 2004. Rouge: A Package for Automatic Evaluation of Summaries. In *Text summarization branches out*. 74–81.

[20] Xiao Liu, Hanyu Lai, Hao Yu, Yifan Xu, Aohan Zeng, Zhengxiao Du, Peng Zhang, Yuxiao Dong, and Jie Tang. 2023. WebGLM: Towards An Efficient Web-Enhanced Question Answering System with Human Preferences. *KDD* (2023).

[21] Yu-An Liu, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Wei Chen, Yixing Fan, and Xueqi Cheng. 2023. Black-box Adversarial Attacks against Dense Retrieval Models: A Multi-view Contrastive Learning Method. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 1647–1656.

[22] Yu-An Liu, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Wei Chen, Yixing Fan, and Xueqi Cheng. 2023. Topic-oriented Adversarial Attacks against Black-box Neural Ranking Models. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1700–1709.

[23] Shayne Longpre, Kartik Perisetla, Anthony Chen, Nikhil Ramesh, Chris DuBois, and Sameer Singh. 2021. Entity-based Knowledge Conflicts in Question Answering. *EMNLP* (2021).

[24] Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically Ordered Prompts and Where to Find Them: Overcoming Few-Shot Prompt Order Sensitivity. *ACL* (2022).

[25] Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When Not to Trust Language Models: Investigating Effectiveness of Parametric and Non-parametric Memories. In *ACL*. 9802–9822.

[26] 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. *choice* 2640 (2016), 660.

[27] Shiyu Ni, Keping Bi, Jiafeng Guo, and Xueqi Cheng. 2024. When Do LLMs Need Retrieval Augmentation? Mitigating LLMs' Overconfidence Helps Retrieval Augmentation. *arXiv preprint arXiv:2402.11457* (2024).

[28] Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. *arXiv preprint arXiv:1901.04085* (2019).

[29] OpenAI. 2022. Introducing ChatGPT. (2022). openai.com/blog/chatgpt.

[30] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *ACL*. 311–318.

[31] Pouya Pezeshkpour and Estevam Hruschka. 2023. Large language models sensitivity to the order of options in multiple-choice questions. *arXiv preprint arXiv:2308.11483* (2023).

[32] 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* (2023).

[33] 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. *arXiv preprint arXiv:2306.17563* (2023).

[34] Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-Context Retrieval-Augmented Language Models. *TACL* (2023).

[35] 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* (2021).

[36] Ruiyang Ren, Yuhao Wang, Yingqi Qu, Wayne Xin Zhao, Jing Liu, Hao Tian, Hua Wu, Ji-Rong Wen, and Haifeng Wang. 2023. Investigating the Factual Knowledge Boundary of Large Language Models with Retrieval Augmentation. *arXiv preprint arXiv:2307.11019* (2023).

[37] Tefko Saracevic. 2016. *The Notion of Relevance in Information Science: Everybody knows what relevance is. But, what is it really?* Morgan & Claypool Publishers.

[38] 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 (1988), 161–176.

[39] 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* (2023).

[40] Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval Augmentation Reduces Hallucination in Conversation. *arXiv preprint arXiv:2104.07567* (2021).

[41] 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* (2023).

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

[43] Yequan Wang, Hengran Zhang, Aixin Sun, and Xuying Meng. 2022. Cort: A new baseline for comparative opinion classification by dual prompts. In *Findings of the Association for Computational Linguistics: EMNLP 2022*. 7064–7075.

[44] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought Prompting Elicits Reasoning in Large Language Models. *NeurIPS* 35 (2022), 24824–24837.

[45] Chen Wu, Ruqing Zhang, Jiafeng Guo, Yixing Fan, and Xueqi Cheng. 2022. Are Neural Ranking Models Robust? *TOIS* 41, 2 (2022), 1–36.

[46] Jian Xie, Kai Zhang, Jiangjie Chen, Renze Lou, and Yu Su. 2023. Adaptive Chameleon or Stubborn Sloth: Unraveling the Behavior of Large Language Models in Knowledge Conflicts. *arXiv preprint arXiv:2305.13300* (2023).

[47] Zhouhang Xie, Sameer Singh, Julian McAuley, and Bodhisattwa Prasad Majumder. 2023. Factual and Informative Review Generation for Explainable Recommendation. In *AAAI*, Vol. 37. 13816–13824.

[48] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. *ACL* (2018).

[49] Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, Soumya Sanyal, Chenguang Zhu, Michael Zeng, and Meng Jiang. 2023. Generate Rather than Retrieve: Large Language Models are Strong Context Generators. *ICLR* (2023).
[50] Hamed Zamani, Fernando Diaz, Mostafa Dehghani, Donald Metzler, and Michael Bendersky. 2022. Retrieval-enhanced Machine Learning. In *SIGIR*. 2875–2886.
[51] Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2021. Optimizing Dense Retrieval Model Training with Hard Negatives. In *SIGIR*. 1503–1512.
[52] 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. *EMNLP Findings* (2023).
[53] Xuchao Zhang, Menglin Xia, Camille Couturier, Guoqing Zheng, Saravan Rajmohan, and Victor Ruhle. 2023. Hybrid Retrieval-Augmented Generation for Real-time Composition Assistance. *arXiv preprint arXiv:2308.04215* (2023).
[54] Honglei Zhuang, Zhen Qin, Kai Hui, Junru Wu, Le Yan, Xuanhui Wang, and Michael Berdersky. 2023. Beyond Yes and No: Improving Zero-shot LLM Rankers via Scoring Fine-grained Relevance Labels. *arXiv preprint arXiv:2310.14122* (2023).
[55] Shengyao Zhuang, Honglei Zhuang, Bevan Koopman, and Guido Zuccon. 2023. A Setwise Approach for Effective and Highly Efficient Zero-shot Ranking with Large Language Models. *arXiv preprint arXiv:2310.09497* (2023).