Retrieval Augmented Generation in the Wild: A System 2 Perspective

Sajjadur Rahman* Dan Zhang* Nikita Bhutani Estevam Hruschka Eser Kandogan
Megagon Labs
{sajjadur, dan_z, nikita, estevam, eser}@megagon.ai

Abstract

Large language models (LLMs), despite their impressive capabilities in natural language understanding tasks in open-domain, often lack effectiveness with similar tasks in enterprise applications due to potential hallucinations, weak multi-hop reasoning ability, and limitations in adapting to heterogeneous data types, among others. Such issues primarily arise due to the absence of private, on-premises enterprises from an LLM's training corpus. Knowledge-intensive tasks in enterprise often require multi-step reasoning, deep contextual understanding, and integration of information stored and accessed in heterogeneous formats (*e.g.*, tables, graphs, documents, and JSON), which LLMs aren't inherently equipped to handle without significant adaptation. To this end, retrieval augmented generation (RAG) offers promise in instrumenting such adaptations on demand. While RAG-based approaches focus on controlling the generation and mitigating hallucinations, existing solutions are not sufficient for the requirements of the enterprise settings.

In this paper, we outline our approaches toward understanding and implementing a more effective RAG workflow in the wild. To achieve the goal, we draw on the cognitive science concepts of System 1 (fast, intuitive thinking) and System 2 (slow, deliberate, analytical thinking.) In particular, we discuss how existing RAG approaches are more aligned to System 1 and propose to shift from traditional single-model architectures to compound AI systems within a System 2 framework to improve RAG, especially in complex enterprise applications. Such compound AI systems adopt a more systematic approach by assigning specialized tasks to different intelligent agents, optimizing retrieval and generation performance with a retrieval-augmented generation workflow.

1 Introduction

Large Language Models (LLMs), despite their impressive performance across various natural language understanding tasks, exhibit significant limitations when applied to enterprise applications in the wild. Primarily, these models may hallucinate—generating plausible-sounding but factually incorrect content—when their parametric knowledge does not align with specific enterprise data [1–3] . An LLM's parametric knowledge depends on its pre-training corpus and can also be influenced by the chosen training strategy and model architecture. This gap is especially problematic since enterprise applications frequently use private, on-premises data, which may differ substantially from the domains in LLMs' pre-training corpora. Such domain misalignment can lead to severe inaccuracies, where the LLMs produce unreliable or misleading information. In addition, enterprises often require consistency and reliability in their outputs. However, LLMs can be sensitive to prompt wording [4], producing inconsistent results even with minor phrasing changes, which undermines their reliability in high-stakes enterprise tasks.

*The first two authors contributed equally.

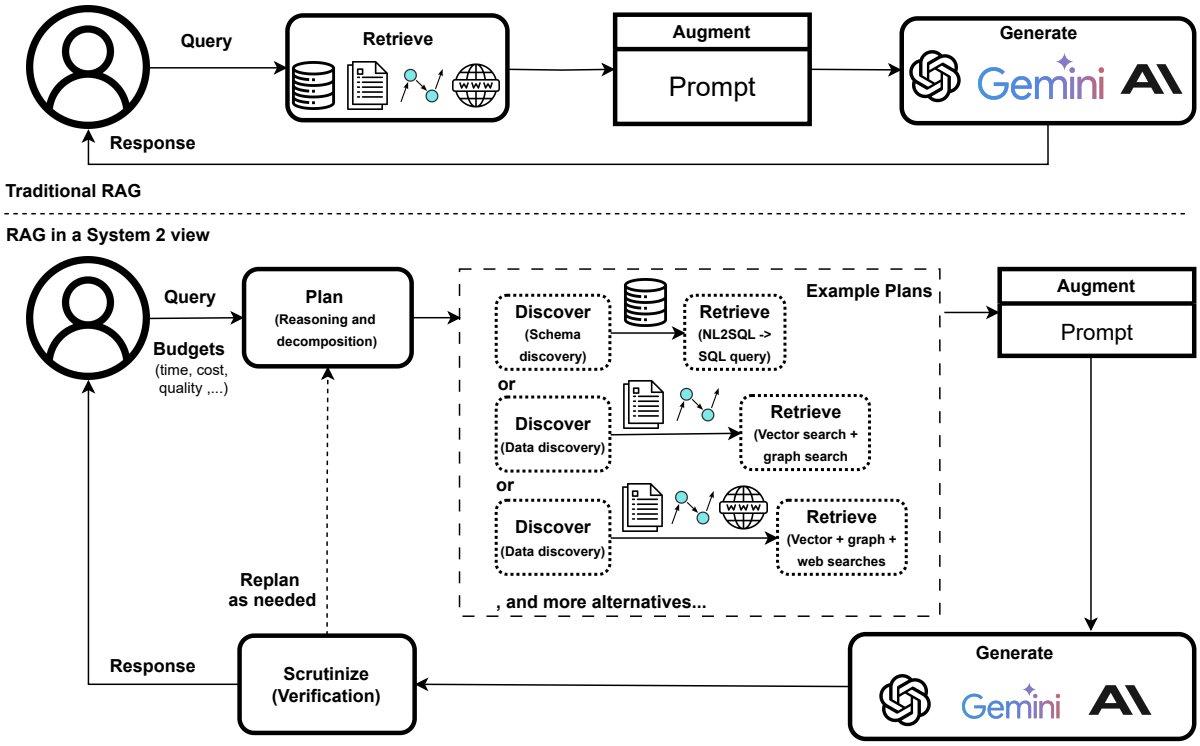


Figure 8: Compared to the traditional RAG setup (above), where a fixed Retrieval-Augmentation-Generation workflow is executed, a System 2 approach (below) involves more deliberate reasoning and action based on critical analysis. For instance, given the same user query, the system must first plan by analyzing the task and may decide to decompose it into smaller components, such as data discovery, natural language-to-query translation, and actual query execution. During the planning phase, constraints or budgets related to factors like time, cost, or quality, along with the nature of the multimodal data sources, may influence the direction of the workflow. After the generation step, a verification process is typically required to evaluate the outcome, which may lead to revisions in subsequent iterations.

Another key challenge for LLMs in enterprise settings is their limited capacity for complex reasoning. Many tasks in this domain require multi-step reasoning, deep contextual understanding, and coherent integration across different data sources, models, and pipelines. LLMs are not inherently designed to manage such complexities without extensive, task-specific adaptation and external grounding. These limitations hinder adopting LLMs in enterprise fields where accuracy, consistency, and reasoning depth are critical, such as healthcare, legal, HR, and data-driven decision-making applications.

Several approaches can help mitigate these limitations, including controlled generation, fact-checking, and post-processing. Controlled generation seeks to constrain the outputs of LLMs by guiding the model toward more reliable responses through techniques such as prompt engineering [5–7], fine-tuning [8, 9], and reinforcement learning from human feedback [10]. Fact-checking [11, 12] involves verifying generated content against highly credible sources to ensure accuracy, with any identified inaccuracies filtered out or corrected during post-processing. Although these techniques offer improvements, they may still struggle with domain-specific challenges and rapidly evolving knowledge. Among the various approaches, augmenting LLMs with external information sources has emerged as one of the most widely adopted

solutions for enhancing accuracy and robustness. This strategy enables LLMs to access up-to-date, domain-specific knowledge, making them more adaptable to new fields or emerging topics. Techniques like retrieval-augmented generation (RAG) [13] integrate external databases or knowledge graphs to ground the model's outputs in verifiable information. By incorporating external data, such as multi-modal documents or enterprise-specific datasets, LLMs can produce more accurate and contextually relevant responses, thereby reducing the likelihood of hallucinations and making them more suitable for critical applications. However, simply using RAG approaches is not necessarily yet the definitive solution. Challenges remain, such as ensuring the quality and reliability of the retrieved information, handling ambiguous or conflicting data, and seamlessly integrating retrieval with generation to maintain coherent and contextually appropriate responses. Consequently, there is still considerable room for improvement in creating more robust and reliable AI systems.

As already shown in the literature, the idea of System 1 and System 2 [14], can be helpful to contextualize the current capabilities and limitations of LLMs [15]. In cognitive sciences, System 1 refers to fast, intuitive, and automatic thinking. This type of system can be characterized by fast thinking or quick judgments and decisions that rely on heuristics and subconscious processing. System 1 is highly efficient for everyday tasks that require intuitive, fast, unconscious, and immediate responses. System 2, however, is associated with slow, deliberate, and analytical thinking. It is more helpful and used for complex problem-solving, critical analysis, and tasks that require conscious, sequential, algorithmic planning and reasoning. System 2 is more resource-intensive and slower but more reliable for tasks requiring careful consideration.

In this System 1/System 2 context, we argue that even though RAG approaches have strong potential to contribute to reducing limitations of current LLMs (by playing a role more closely related to System 2), most current RAG approaches only weakly resemble System 2 thinking. The retrieval and generation steps are often designed to be fast and instantaneous, aligning with System 1 thinking, rather than slow and logical as in System 2, which presents challenges on both the data and model sides. For example, research [16, 17] has shown that augmenting LLMs with retrieval without rigorously assessing necessity may adversely impact overall performance.

Therefore, as illustrated in Figure 8, we advocate for a shift from traditional single-model architectures to compound AI systems within a System 2 framework to enhance RAG, particularly in complex enterprise applications. Compound systems enable a collaborative approach to problem-solving by distributing specialized tasks across distinct agents, each optimized for specific functions, improving both retrieval and generation performance in challenging real-world settings.

On the retrieval side, enterprise applications often involve complex, multi-step, and sometimes ambiguous tasks that require deeper reasoning and structured workflows. A compound system can enhance this by assigning specialized agents to handle diverse aspects of data, such as heterogeneous formats (e.g., text, tables, graphs, parametric information) and noisy or incomplete data sources. This allows for agents skilled in reconciliation and semantic querying to refine the retrieval process through iterative, logic-driven interaction, improving both precision and relevance of context.

On the generation side, challenges like hallucination, fact verification, and adherence to context remain key obstacles. In a compound system setup, individual agents can be tasked with verifying facts, maintaining context alignment, and evaluating outputs for accuracy before finalizing responses. This division of labor can be exploited towards reducing hallucinations and enhancing reliability by enabling dynamic inter-agent evaluation, where each agent iteratively cross-checks and validates the others' outputs [18, 19]. For example, in domain-specific conversational AI, particularly in regulated industries, compound AI systems offer a pathway to safer, more reliable, and robust deployments by integrating domain expertise, context sensitivity, and rigorous validation at each step.

The paper is organized as follows. First, in section 1, we provide a brief overview of traditional RAG models and highlight their limitations, especially in real-world, domain-specific applications. We then

motivate more concretely the need for a System 2 RAG approach, in section 3. In section 4, several approaches to enhance RAG adaptability to System 2 thinking are discussed. Finally, we present our vision for future research in section 5, exploring the potential of compound AI systems as System 2 solutions to RAG.

2 Background

2.1 Traditional Approaches Towards RAG

Retrieval augmented generation (RAG) aims to address hallucinations and factual inaccuracies in LLM-generated content. RAG infuses external knowledge [20, 21], such as knowledge bases and web documents, while prompting LLMs to help generate responses grounded on relevant information. The integration of RAG-based workflows in prompting LLMs has enjoyed widespread adoption, enhancing the suitability of LLMs for real-world applications. Development of a RAG system often begins with an indexing step, which involves cleaning and segmenting the documents — segmentation is required to prepare chunks of information that carry meaningful and strong signals, and in addition, to fit into an LLM's context window (when using LLMs with limited context windows). Each chunk is then encoded into a vector representation using an embedding model and stored in a vector database. This step is essential for enabling efficient similarity searches in the subsequent retrieval phase. As shown in Figure 8, the following are the standard phases of a traditional RAG workflow:

Retrieval. Given a user query, a RAG system employs an encoding model used during indexing to transform the query into a vector representation and calculates the similarity scores between two vectors: query and candidate text chunks within the indexed corpus. Based on these scores, the system retrieves the top-K chunks with the highest similarity to the query, which are then used as the expanded context for the next stage.

Augmentation. The selected chunks are incorporated into a prompt as expanded context to provide additional relevant information. The goal of such enhancement is to reduce hallucination and improve accuracy of the model's response. By providing targeted context, the RAG system ensures the model can ground its answer in the most pertinent retrieved data.

Generation. The user query, along with the augmented context of selected documents, is synthesized into a coherent prompt, which is then provided to an LLM that will perform the final generation task. Depending on the task requirements, the model may either draw upon its internal knowledge or limit its response to information in the provided documents.

2.2 Limitations of RAGs

Traditional approaches to RAG do not directly apply to real-world scenarios due to the heterogeneity of data, the complexity of workflows, and the strict constraints on expected task performances.

2.2.1 Lack of Robust Deliberation

Recent studies show how RAGs are not universally effective [22, 23]. Adding noisy or irrelevant passages can override correct LM knowledge, leading to errors (see Table 1). An effective RAG should balance accurate recall with selective retrieval. Identifying when to recall versus retrieve raises key questions: (a) What factors impact an LM's recall accuracy? (b) What influences RAG performance? (c) What error patterns are common between LM and retriever responses?

Previous research on memorization in LMs and retriever performance has some limitations: (a) it focuses only on entities, while real-world information includes both entities and relations [24, 25]. (b) It

Triple: (Chicago, country, United States of America) **Question:** What country is Chicago located in? **LM Answer:** United States [Correct] **Context:** The Chicago Municipal Tuberculosis Sanitarium was located in Chicago, Illinois, USA.... [Correct Retrieval] **RALM Answer:** USA [Correct]	**Entity Popularity:** 95.0%*ile* **Entity-Relation Popularity:** 97.4%*ile*
Triple: (George H.W. Bush, educated at, Yale University) **Question:** What educational institution did George H.W. Bush attend? **LM Answer:** Yale University [Correct] **Context:** The George H.W. Bush Presidential Library is located on a site on the west campus of Texas A&M University in College Station, Texas.... [Wrong Retrieval] **RALM Answer:** Texas A&M University [Wrong]	**Entity Popularity:** 89.5%*ile* **Entity-Relation Popularity:** 41.8%*ile*
Triple: (Ellen Litman, educated at, University of Pittsburgh) **Question:** What educational institution was Ellen Litman educated at? **LM Answer:** Stanford University [Wrong] **Context:** Ellen Litman Ellen Litman (born 1973) is an American novelist. She received the Rona Jaffe Foundation Writers' Award in 2006. Born in Moscow, Russia, she emigrated with her parents in 1992 to Pittsburgh, Pennsylvania. She was educated at the University of Pittsburgh and earned a B.S. in Information Science. ... [Correct Retrieval] **RALM Answer:** University of Pittsburgh [Correct]	**Entity Popularity:** 10.3%*ile* **Entity-Relation Popularity:** 17.9%*ile*

Table 1: QA examples from WiTQA with predictions of varying popularity of question entity and entity-relation pair. The predictions from LM (GPT-3.5) with no augmentation and RALM (GPT-3.5+BM25) are shown. In the top row, both LM and RALM provide correct answers for the popular question. In the middle row, LM generates correct answer but RALM provides incorrect answer due to retrieval errors. In the bottom row, LM provides incorrect answer for an infrequent entity-relation pair.

examines either retrievers or LM recall independently, overlooking their interplay [26–28]. To address these limitations, in previous work, [16] focused on the QA task and analyzed the performance of 10 LMs across 5 retrieval settings. They introduced WiTQA [16], a new dataset of QA pairs generated from Wikipedia triples, selected based on entity and relation popularity, each paired with supporting passages and popularity scores. The investigation of RAGs zero-shot performance on WiTQA yields the following key findings (see Figure 9):

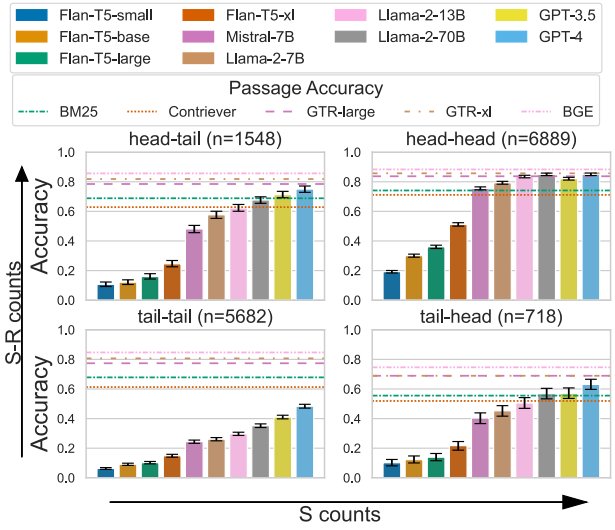


Figure 9: Analysis on Vanilla LMs with BM25, Contriver, GTR, and BGE passage accuracy over S-R counts and S counts (n = the number of questions in the group). In the top row, S-R counts are higher than the median. In the bottom row, they are less than or equal to the median. In the left column, S counts are less than or equal to the median, and in the right column, they are higher than the median.

- LMs can often recall frequently encountered entity-relation pairs from pre-training without retrieval, but this depends on model size; larger models capture more long-tail relations for popular entities, though accuracy drops for less common facts.
- For long-tail entity-relation pairs, retrieval performs better than LM recall, suggesting retrieval augmentation benefits these cases but may introduce override issues with well-known pairs.
- LMs outperform retrievers on well-known entity-relation pairs involving long-tail entities, contrasting prior studies where large LMs struggled with these pairs.

Using these insights, a selective memory integration module was designed, that applies retrieval augmentation or LM recall based on entity-relation popularity. The main idea is closely related to a System 2 approach, in which before trying to answer a question, the system first tries to figure out (reasoning task) whether it should use the retrieval mechanism or not. It was found it could improve QA performance by up to 10.1% [16].

Another limitation of most current RAG models is characterized by their reliance on localized context retrieval, making them less effective for tasks that require *holistic reasoning*—the ability to synthesize, aggregate, and analyze information across multiple documents. For instance, when asked, "Which company employed the most people?" traditional retrieval models may return individual statistics for each company without a comprehensive comparison. This core limitation reveals that while RAG systems are adept at fact retrieval, they falter with broader, cross-document reasoning. Bridging this gap requires models capable of multi-document synthesis, comparative analysis, and extensive dataset integration.

One alternative to to address these limitations of RAG models in holistic reasoning is to bypass retrieval altogether and use long-context language models (LCLMs). These models are designed to handle and process significantly larger chunks of information, enabling them to reason effectively over extensive contexts or large sets of documents without the need for iterative retrieval steps. By eliminating the dependency on retrieval mechanisms, LCLMs reduce the risk of retrieving irrelevant or incomplete information, which can compromise reasoning quality. Furthermore, their ability to maintain coherence across lengthy inputs makes them particularly well-suited for tasks requiring nuanced understanding, cross-referencing of details, and synthesis of insights from diverse sources within a single reasoning framework.

To investigate into this, [29] conducted a comparative study of LCLMs and RAG models using HoloBench, a benchmark specifically designed to evaluate holistic reasoning capabilities. It compared two large LCLMs, Llama-3.1-405b and GPT-4o, alongside a smaller LCLM, Llama-3.1-8b. For document retrieval, the work employed `BAAI/bge-large-en-v1.5`, an effective embedding-based model that retrieves the 2k tokens most similar to the query.

The findings in [29] reveal that as context length exceeds 4k tokens, larger vanilla LCLMs consistently outperform RAG-based models, indicating their superior ability to manage longer contexts where RAGs struggle to retrieve relevant information (see Figure 50). Interestingly, with smaller models like Llama-3.1-8b, RAG performs better when context length surpasses 16k tokens. This aligns with previous findings[16] that retrieval models can enhance the performance of weaker models by compensating for their reasoning limitations, even in the presence of retrieval errors. A promising future direction would be to adopt a System 2-based approach and integrate a dynamic mechanism for determining the optimal amount of information to retrieve based on the query and context length, particularly when working with weaker models for holistic reasoning.

2.2.2 Impact of Prompt Sensitivity

A notable limitation of LLMs is their sensitivity to the arrangement of components within prompts, which directly influences their performance in understanding and reasoning on specific tasks. Prior

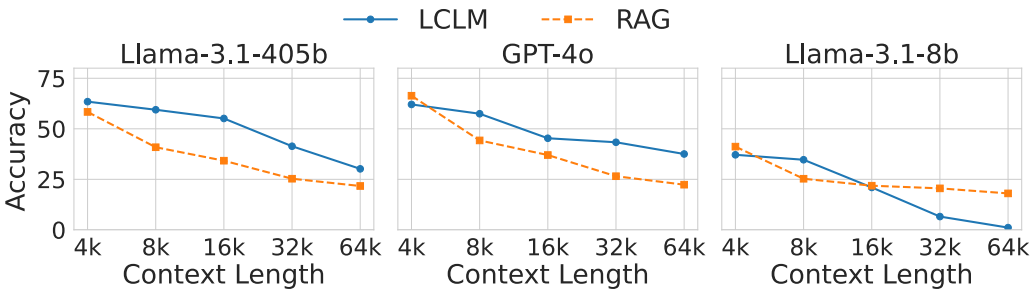


Figure 10: Performance comparison of LCLM and RAG. For long contexts, large models outperform RAG but a retriever helps a small model due to its limited ability to handle long contexts.

research has shown that LLMs are affected by the ordering of few-shot demonstrations [30]. These findings raise an important question: are LLMs similarly affected by the order of elements in prompts across diverse tasks? For instance, in multiple-choice question (MCQ) answering tasks, does the order of answer options impact LLM performance? Figure 11 (extracted from [31]) shows the sensitivity of GPT-4 to options order using a sample from the common sense QA benchmark. Within this context, [31] aims to address the following research questions: (1) To what extent do LLMs exhibit sensitivity to the order of options in multiple-choice questions? (2) What factors contribute to LLMs' sensitivity to the order of options?

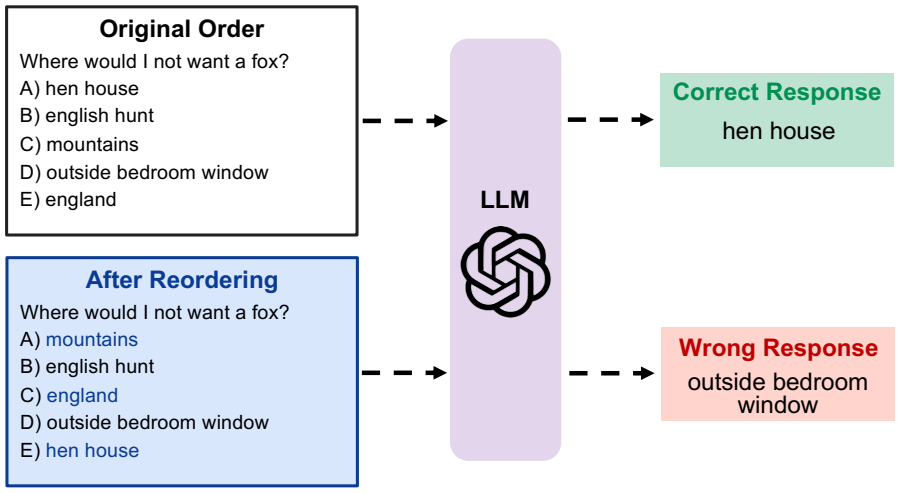


Figure 11: GPT-4 sensitivity to reordering options: Upon changing the order of choices, GPT-4 changes its prediction from "hen house" to "outside of bedroom window" (the example is from the CSQA dataset).

To address the first question, [31] conducted experiments using GPT-4, InstructGPT (text-davinci-003), and Llama-2-13b (chat version) across five multiple-choice question benchmarks. A surprisingly high sensitivity gap of up to 85% in the zero-shot setting (see Table 2) was found. Furthermore, in the few-shot setting, introducing demonstrations to the prompt led only to marginal gains in robustness, if any improvement was observed.

Regarding the second question, it is hypothesized that this sensitivity arises from positional bias, where LLMs display a preference for certain answer placements when uncertain. To investigate, [31] analyzed instances where the models' predictions shifted upon reordering answer options. Additionally, was also found that increasing the number of options, while keeping the top possible answers, only

Tasks	GPT-4			InstructGPT			Llama-2-13b		
	Vanila	Min	Max	Vanila	Min	Max	Vanila	Min	Max
CSQA	84.3	-12.6	+10.3	72.3	-24.0	+19.1	62.2	-28.9	+25.5
Logical Deduction	92.3	-8.1	+5.0	64.0	-39.4	+34.7	53.0	-30.7	+34.7
Abstract Algebra	57.0	-30.0	+23.0	33.0	-31.0	+39.0	32.0	-32.0	+53.0
High School Chemistry	71.9	-23.6	+18.2	44.8	-28.5	+38.0	40.6	-32.7	+45.6
Professional Law	66.1	-12.7	+12.1	48.6	-24.9	+25.7	43.8	-32.8	+32.9

Table 2: **Zero-shot order sensitivity**; all three LLMs display a notable level of sensitivity to the order of options across various benchmarks.

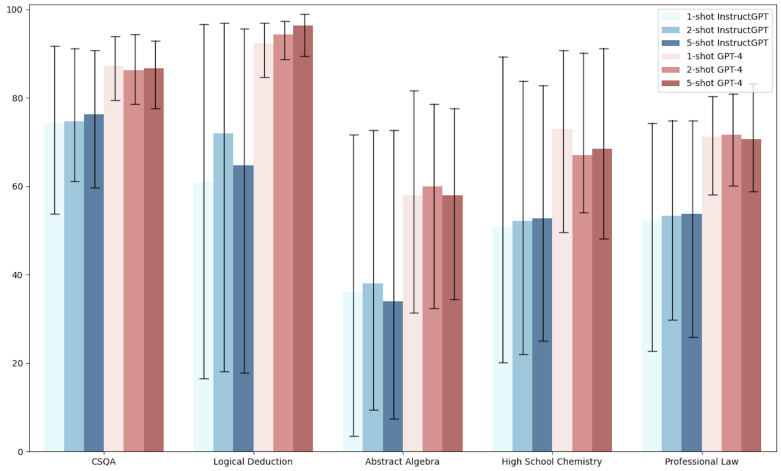


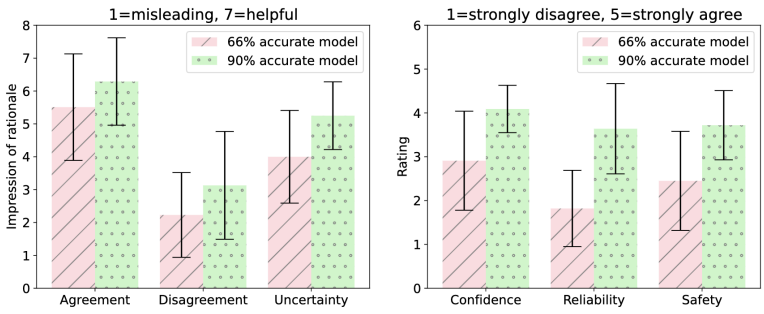
Figure 12: **Order sensitivity in the few-shot setting:** The error bars represent the range of minimum and maximum accuracy achievable in each task through oracle reordering. Our observations are as follows: (1) The sensitivity gap consistently remains substantial in the few-shot setting. (2) As performances improve, the sensitivity gap shrinks. (3) Adding more demonstrations does not necessarily results in a reduction of the gap.

gradually affected performance, suggesting that positional bias rather than option count plays a larger role in LLM sensitivity (see Figure 12).

Another interesting finding from [31] is that, instead of using the original order of the multiple-choice options, one can adopt a "system 2" approach and reason before output the first answer generated by the LLM. In this sense, before deciding on the final answer for the question, the same question can be posed to the LLM multiple times (five times in the referenced study), each time with a randomly shuffled set of choices. Afterward, the answers are aggregated through a reasoning mechanism (in the paper, a very simple majority voting reasoning was employed). This approach has been shown to improve the LLM's overall performance.

2.2.3 Transparency and Accountability in Downstream Applications

To study the implications of trust and accountability of RAG pipelines in downstream applications, [32] considers the task of generating natural language explanations of knowledge-intensive task (KIT) decisions such as multiple-choice question answering. Given the setting for generating corroborating and refutation complete rationales for KIT model decisions, the suitability of retrieval-augmented rationale



(a) Impact on user perception (b) XAI Trust Scale feedback

Figure 13: (a) Irrespective of agreement or disagreement with the KIT model prediction, participants indicated a more negative impression about the rationalization of the lower confidence model prediction. (b) Participant feedback on the trust scale indicates lower confidence for lower accuracy model rationalization.

generation using LLMs is explored. The prompt to LLMs is enriched with relevant knowledge from external sources to condition the rationale generation on facts. Three human subject studies were conducted to evaluate the effectiveness of such rationales in communicating KIT model decisions.

More specifically, two studies were conducted, via crowdsourcing, to evaluate the preferability and acceptability of such rationales to crowd-workers. In another study involving experts — motivated by existing literature on trust in explainable AI [33, 34] — the implications of faithfully rationalizing KIT model decisions irrespective of their correctness was explored. The crowd-sourced studies demonstrate that, more often than not, crowd workers prefer LLM-generated rationales to crowdsourced rationales in existing datasets, citing their factuality, sufficiency, and convincing refutation. Follow-up fine-grained analysis reveals that LLM-generated rationales still have significant room for improvement along dimensions such as insightfulness (*i.e.*, providing new information), redundancy (*i.e.*, avoiding repetitive text), and generalizability (*i.e.*, domain invariance.) The expert-sourced study confirms that faithful rationalization of incorrect model predictions degrades humans' trust in the generated rationales. The work further explores the utility of instrumenting mechanisms to intervene in the incorrect predictions via a review-then-rationalize pipeline instead of faithfully rationalizing and find that even simple strategies may help intervene up to 71% of the incorrect predictions.

Figure 13a [32] summarizes the participants' impression of a rationale immediately after viewing the model prediction. When the participants disagreed with the model prediction, they exhibited a stronger negative impression about the rationales for the 66% accuracy condition compared to the 90% accuracy condition. Even when participants agreed with the model prediction, their impression of the rationales remained more negative. The intuition is that the higher disagreement with the model coupled with observing the faithful rationalization of the incorrect prediction negatively impacted participants' perception of the reliability of the rationales. These observations are confirmed by analyzing the results of the follow-up survey (see Figure 13b.) Unsurprisingly, participants for the 66% accuracy condition rated their confidence in the generated rationales and the reliability of the rationalizer significantly lower compared to the 90% accuracy condition.

3 Towards System 2 RAG in the wild

Towards productizing generative AI, there has been a shift from monolithic models to compound AI systems [35] that incorporate various components other than LLMs for data retrieval, coordination,

and utilizing proprietary models and services. Examples of such systems include Hiring Assistant by LinkedIn [36], AI-BI Genie by Databricks [37], Agentforce by Salesforce [38], and Magentic-One by Microsoft [39], among others. These systems are designed to ensure performance for complex tasks, adaptability to heterogeneous data and use cases, and a higher degree of trust in the production setting.

Motivating example. Consider LinkedIn and Indeed, two global job-matching and hiring platforms in the HR domain. These companies are employing RAG-based workflows for a multitude of tasks in HR, such as matching, recruitment, and career guidance, among others [40, 41]. Given the task of matching job seekers with job postings, a popular use-case of generative AI is communicating meaningful explanations to job seekers about their relevance to the job. To support such a use-case, enterprises use RAG to infuse domain-specific relevant information, which guides the generation of explanations using LLMs [40]. Identifying the relevant information to be provided to the LLM in such a domain-specific context can be challenging due to the heterogeneity and scale of the data. Moreover, within such a production-oriented setting, enterprises have to remain cognizant of real-world constraints such as cost, latency, accuracy, and trustworthiness.

3.1 Data Heterogeneity in the Wild

Within an enterprise setting, different teams in an organization manage project-specific sources of data, collected and often transformed through various workflows over time. Such an observation is derived from the authors' experience working with enterprise data in the human resources (HR) domain at Megagon Labs. For example, going back to the example of explaining job seeker and job posting match, a team tasked with such work will be interested in unstructured resumes and job descriptions, structured data extracted from the text and their representations, and HR domain-specific knowledge, *i.e.*, , relationships among different concepts such as jobs and benefits. Another team working on assisting job-seekers with search (*e.g.*, role and company) will be more interested in rather job market trends, company-specific information and search logs, among others.

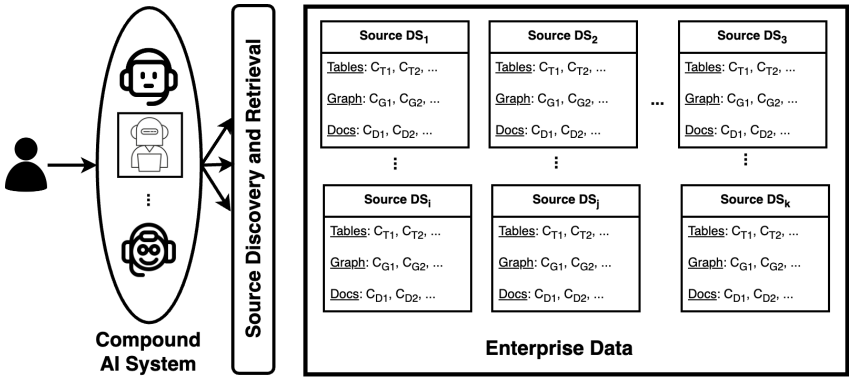


Figure 14: Conceptual data model for RAG in the wild.

We introduce the data model for such a setting in Figure 14: compound AI systems operating over such enterprise data [3]. The data model is multi-granular and multi-modal. Each team-specific data source DS_i represents a collection of sources ($S_m \in DS_i$) corresponding to different modalities (m) — such as multiple tables and documents — created by specific teams. While enterprises may contain data corresponding to other modalities such as audio, video, and image, we limit our scope to documents ($m = D$), tables ($m = T$), and graphs ($m = G$) in this case. Each data source is organized in a hierarchical manner — at the coarse-grained level data is organized in data sources DS_i. Within each data source, data is organized in various collections C_m depending on the modality. Within each

collection, data can be stored and managed by different systems (C_{mi}) depending on the downstream application such as data warehouse, data lake, and lakehouse systems [42, 43]. Therefore, the assumption of traditional RAG where these data sources or collections or databases are known beforehand, breaks down in such a setting. Rather a multi-step approach to data discovery (closer to System 2) is required to identify the relevant coarse- and fine-grained data given a user query.

3.2 Rethinking RAG

To this end, a major consideration in moving a RAG pipeline from System 1 to System 2 thinking involves shifting toward deliberate analytics and explicit reasoning. This transition means that decisions—such as when and where to retrieve information, how much data to retrieve, and how to integrate retrieved information into the generated response—should be made through rigorous reasoning. This reasoning should carefully assess the unique characteristics of the task, data, tools, and available models to ensure outcomes that are both optimal and context-aware. Note that recently released LLM such as o1[1] instruments slow thinking by executing some type of reasoning mechanism before generating the final answer. Even though there has been discussions within certain research circles, which mentions the possibility of having chain of thought and self-reflection empowering o1 models before they generate an answer, it is important to state that no official documentation confirms the use of such techniques in the o1 models) —- One consequence of this new reasoning capabilities of these models is the additional time spent thinking, that makes it more effective for complex reasoning tasks, particularly in science and mathematics. However, the additional step leads to higher latency during inference, which may vary depending on the task. In addition to that, o1 is similar to the earlier monolithic LLMs where the parametric knowledge remains abstract, thereby lacking transparency and controllability as no affordances are provided to the users to interact with the decision-making process.

Given a knowledge-intensive task that requires complex reasoning and planning, a systematic approach is necessitated to ensure transparency of the workflow, integrate user guidance at various steps, and enable optimization under real-world constraints such as cost, latency, and accuracy. As illustrated in Figure 8, at the core of the System 2 RAG pipeline lies a planner, which functions as a reasoning module. This planner is grounded in specific tasks and budgets (*e.g.*, time, cost, quality) and operates by foraging and analyzing the properties of data and agents within the registries. Moreover, retrieval is expanded to not only extracting information from documents but also other formats of raw data and data management systems as outlined in Figure 14. Adaptation of retrieval to large-scale heterogenous data necessitates reconciliation and reranking of retrieved information. With the presence of heterogeneous information, the augmented generation requires further scrutinization through fact-checking and verification,

4 Design components

To achieve effective System 2 RAG pipelines, it is essential to address challenges in both data and model aspects. System 2 decision-making relies on an organized, contextually rich data pool for informed outcomes, making efficient identification and organization of relevant information critical. Data enhancements, such as data discovery agents, play a key role by locating, structuring, and tagging pertinent data to create a streamlined and accessible knowledge base. Similarly, selecting and configuring models for each task is crucial. Choosing the right model, tuning it to align with user expectations, and ensuring seamless integration with the data pipeline all contribute to achieving optimal performance. In this section, we provide a high-level overview of our work in tackling these data and model challenges in developing a System 2 RAG pipeline.

[1] https://openai.com/o1/

4.1 Data-level enhancements

4.1.1 Data Discovery

As an initial effort to understand the potential impact in the data discovery task, we adapted existing datasets and benchmarks in open-domain — from question answering and complex reasoning tasks to natural language querying over structured data — to evaluate coarse- and fine-grained data discovery and task execution performance. Our experiments reveal the impact of data retriever design on downstream task performance — 46% drop in task accuracy on average — across various modalities, data sources, and task difficulty. The results indicate the need to develop optimization strategies to identify appropriate LLM agents and retrievers for efficient execution of CASs over enterprise data. This need is well-aligned with the System 2 type of thinking, in which before performing the retrieval, there is a need for reasoning on what data is available and how it should be retrieved. After such a reasoning step, the retrieval (and the RAG results in general) have the potential to improve.

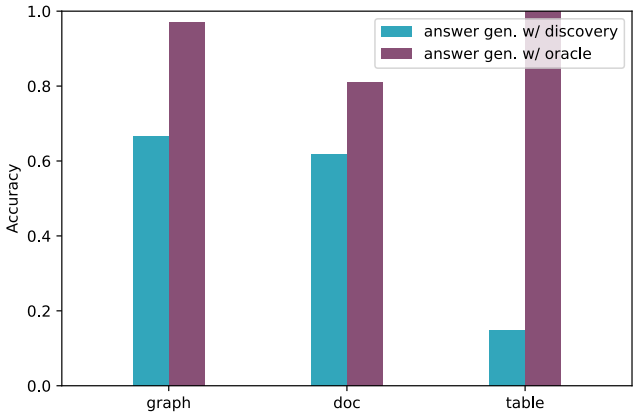


Figure 15: Task execution performance degrades due to poor data discovery despite the purported proficiency of LLMs.

When we explored the impact of data discovery performance on the accuracy of downstream task execution. Figure 15 captures the task execution accuracy for two scenarios: oracle and discovery. Oracle is the case where the ground truth discoverable element is provided to the LLM (*e.g.*, GPT-3.5 turbo) to provide the final answer to a question. Discovery captures the scenario where elements retrieved by the best-performing document, sub-graph, and table discovery models are provided to the LLM. We observe a significant drop in performance from oracle to the discovery scenario, showcasing a 46% decrease in accuracy.

4.1.2 Less is More for Evaluation

Evaluating text generation is crucial for creating high-quality systems. However, aligning automatic evaluation metrics with human judgment remains challenging [44, 45]. While LLMs demonst rate promising correlations with human evaluations, they encounter issues like high costs and the Lost-in-the-Middle problem [46], where key information in the middle of lengthy documents is frequently overlooked in summary evaluations.

To tackle these challenges, [47] introduced a straightforward yet effective approach known as *Extract-then-Evaluate*. At run-time, this method begins by extracting significant sentences from a lengthy source document and concatenating them until the extracted text reaches a predefined length. Subsequently, it assesses the quality of the summary based on this extracted content using LLMs. Figure 16 shows

Figure 16: Overview of the long document summary evaluation task by LLMs. Evaluating long document summaries by LLMs is expensive and shows limited alignment with human evaluations. This study demonstrates that extracting important sentences for evaluation in advance not only reduces evaluation costs but also exhibits better alignment with human evaluations.

the overview of our approach. Notice that Extract-then-Evaluate brings a System 2 type of thinking. The evaluation requires the analysis of the original document and the identification of the key elements. Only after reasoning and identifying such key elements is the evaluation of the quality of the summary performed.

The experiments explore various sentence extraction techniques, encompassing both matching-based and model-based methods, such as LEAD, ROUGE, BERTScore, and NLI. Their performance is evaluated across multiple datasets, including arXiv, GovReport, PubMed, and SQuALITY [48, 49]. The main results are shown in Table 3. In the experiments, LLMs demonstrated a notable enhancement in correlation with human judgment when compared to non-LLM baselines. However, this improvement came with increased evaluation costs due to the full document prompt length. Extracting key information before evaluation not only reduced costs but also improved performance, attributed to the Lost-in-the-middle problem, where LLMs struggle with critical information in lengthy documents. This trend showed that LLMs perform better with shorter, more informative documents. Lastly, even within a limited budget, the approach delivered comparable performance to top configurations, achieving similar results to the best extraction method while cutting evaluation costs in half.

Based on these observations, it is possible to conclude that effective data pre-processing can reduce costs while allowing the model to concentrate on key information, ultimately enhancing performance.

4.2 Model-level enhancements

Despite ongoing advancements in LLM and RAG models and their continual scaling, there appears to be no clear limit to their growth potential. As a result, integrated systems and workflows—often

	Consistency						Relevance						Faithfulness					
	arXiv			GovReport			arXiv			GovReport			PubMed			SQuALITY		
Methods	r	ρ	£	r	ρ	£	r	ρ	£	r	ρ	£	r	ρ	£	r	ρ	£
Reference-based metrics																		
ROUGE-1	-0.08	-0.13	-	-0.12	-0.11	-	0.29	0.25	-	0.53	0.52	-	0.32	0.30	-	-0.33	-0.13	-
BERTScore	-0.09	-0.10	-	0.00	-0.04	-	0.22	0.18	-	0.38	0.38	-	0.49	0.49	-	-0.12	0.02	-
BARTScore	0.32	0.36	-	0.51	0.48	-	0.00	0.03	-	0.18	0.24	-	0.49	0.47	-	-0.06	-0.17	-
Reference-free metrics																		
FactCC	0.22	0.19	-	0.28	0.27	-	0.13	0.13	-	0.05	0.04	-	-0.09	-0.14	-	0.13	0.14	-
SummaC	0.32	0.32	-	0.39	0.38	-	0.09	0.08	-	0.05	0.04	-	0.51	0.55	-	0.18	0.24	-
Reference-free metrics with LLM (ours)																		
Full document	0.61	0.46	$0.15	0.33	0.34	$0.10	0.58	0.52	$0.15	0.12	0.11	$0.10	0.64	0.70	$0.11	0.51	0.38	$0.14
Best extraction	0.71	0.50	$0.05	0.62	0.60	$0.09	0.63	0.58	$0.07	0.36	0.40	$0.07	0.76	0.80	$0.07	0.85	0.81	$0.04
Pareto efficient	0.71	0.50	$0.05	0.60	0.61	$0.05	0.55	0.48	$0.04	0.37	0.37	$0.05	0.75	0.75	$0.05	0.85	0.81	$0.04

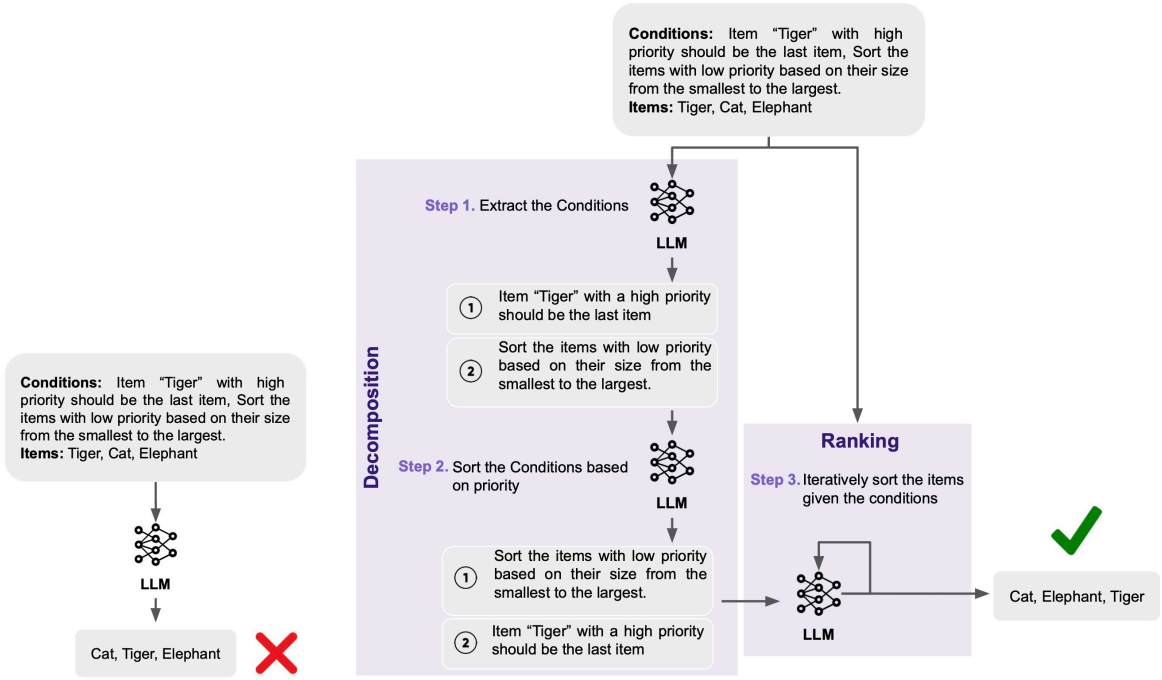
Table 3: Results for Pearson correlation (r), Spearman correlation (ρ), and the average evaluation cost per instance (£) indicate that extracting important sentences before evaluation (Best extraction) can yield a higher correlation. Even under a limited budget (Pareto efficient), these results show comparable or even higher correlations compared to the full document setting, with lower costs.

called compound systems or agentic workflows—are emerging. These systems combine LLMs with multiple components, including repeated model calls, retrievers, and external tools, through commercial frameworks like LangChain, LlamaIndex, Auto-GPT, and AgentGPT. Such frameworks empower developers to create agents with unique decision-making capabilities, specialized expertise, and integration with proprietary systems or datasets, as well as build diverse applications ranging from customer service chatbots to advanced decision-support systems. We now highlight several research efforts that showcase compound systems designed to address complex NLP tasks and emphasize that the presence of a System 2 type enhances the performance in such complex tasks.

4.2.1 Multi-conditional ranking

Ranking items based on multiple conditions has wide-ranging applications across various fields. In recommendation systems, for example, once top candidates are shortlisted, re-ranking them according to specific conditions—like genre or category—can greatly enhance the user experience. Similarly, in competitive job markets, this approach is essential for matching resumes to job postings, allowing for prioritization by skills, experience, and other relevant factors. While there has been considerable advancement in ranking extensive document collections given a query [50–52], the nuanced task of ranking a smaller set of items based on multiple conditions has not been addressed in prior research.

To address this gap, [53] defines and investigates the task of multi-conditional ranking (MCR) through the introduction of MCRank, a comprehensive benchmark that encompasses various item types and ranking conditions for evaluating MCR performance. MCRank includes a diverse array of conditions, including positional, locational, temporal, trait-based, and reasoning conditions. Specifically, MCRank was developed by creating a dataset with 18 scenarios varying in item categories, number of conditions (1, 2, or 3), and item set sizes (3, 5, or 7). Each scenario included 200 samples, generated by compiling data and labels for different condition types, featuring randomly ordered item sets with correct rankings. Positional conditions were sourced from Big-Bench's auto-categorization task and Amazon reviews. For scenarios requiring multiple conditions, additional criteria like character counts or positional conditions were added to simulate realistic complexity. This process ensured a robust dataset for evaluating holistic reasoning in scenarios that simulate situations close to real-cases in which users want to rank items based on different conditions defined by their own needs.



(a) Base Approach (b) EXSIR (Ours)

Figure 17: Overview of multi-conditional ranking. Instead of directly prompting LLMs to rank items based on the given conditions, we first extract and sort the conditions based on their priority. Then, we iteratively apply these sorted conditions to the item list.

Furthermore, we develop EXSIR, a novel decomposed reasoning method that iteratively refines rankings. The process begins with extracting individual conditions from a given string and organizing them into a coherent list. A sorting mechanism then arranges these conditions based on their assigned priorities. Finally, the sorted conditions are applied iteratively to the item list, refining the rankings in each cycle based on the current condition. Figure 17 illustrates the workflow of EXSIR along with an example of MCRank.

Initial investigations into existing LLM performance on MCRank show a clear decline in accuracy as both the number of items and conditions increase. Specifically, we observe a sharp drop in ranking accuracy for LLMs like OpenAI o1-mini, GPT-4, ChatGPT (both turbo versions), Llama 3.1-70B, and Mistral (7B) when tasked with three conditions and seven items, with accuracy nearing 0%. EXSIR improves ranking accuracy on MCRank by up to 14.4%, outperforming strong baselines such as Chain-of-Thought (CoT) (see Figure 18). These results demonstrate how the initial steps towards the System 2 type of thinking (present in the EXSIR approach) allowed to improve the performance of even very strong baselines.

4.2.2 Trust but Verify

Motivated by the observations from the study reported in Section 2.2.3, we create a two-stage review-then-rationalize (see Figure 19) pipeline to evaluate the impact of intervening incorrect model predictions before rationalization. The pipeline instruments a reviewer module that employs another LLM (GPT-3.5 `text-davinci-003` (`temperature = 0`)) to evaluate the correctness of the knowledge-intensive task (KIT) model and refrain from rationalizing potentially incorrect decisions.

Depending on the task and data domain, the suitability of the reviewer model may vary. Given the

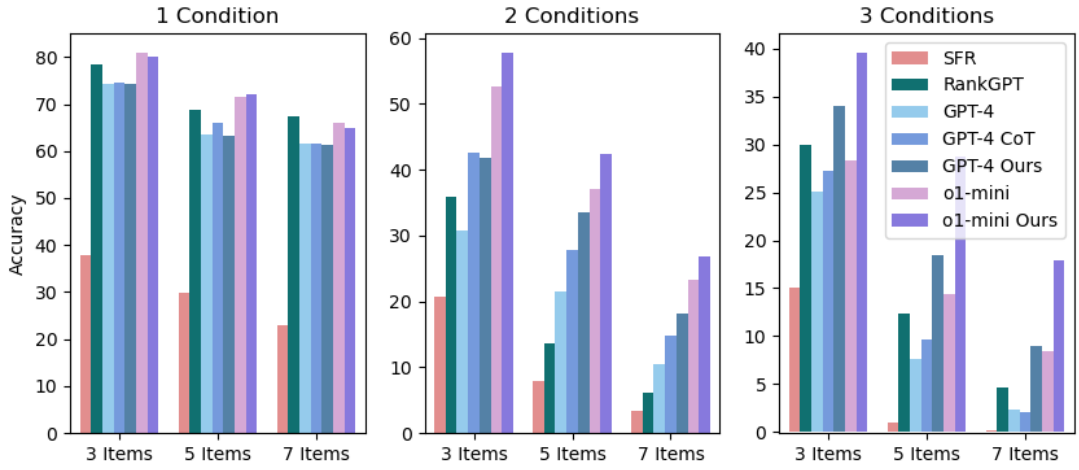


Figure 18: Evaluating the impact of EXSIR against zero-shot CoT prompting for token-level items. We additionally report SFR and RankGPT performances as representatives of existing rankers.

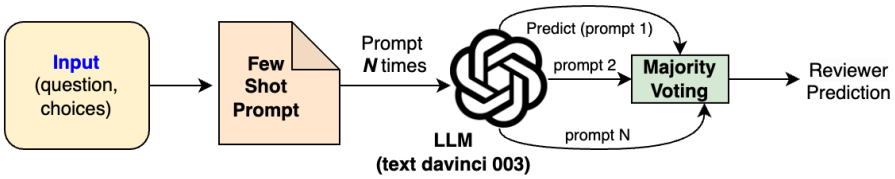


Figure 19: Self-consistency-based Reviewer—intervene for any disagreement with the KIT model prediction.

complexity of knowledge-intensive tasks, we employ a self-consistency-based decoding strategy [54] where the reviewer is asked the same question N (=5) times, and the final response is selected via majority voting. The reviewer then compares the model's prediction with its prediction, and The rationalizer is utilized only when the KIT model and the reviewer agree.

Dataset	Prediction Errors (Test Set)	Errors Intervened	
		Greedy Decoding	Self-consistency
CSQA	321	166 (51.71%)	187 (**58.26**%)
OBQA	155	102 (65.81%)	110 (**70.97**%)

Table 4: The review-then-rationalize pipeline helps intervene in incorrect predictions of a knowledge-intensive task (KIT) model. The self-consistency-based reviewer outperforms the greedy decoding-based reviewer.

As shown in Table 4, for knowledge-intensive tasks such as Commonsense QA and Openbook QA, the proposed pipeline helps intervene up to 58% and 71% of the incorrect predictions. Unsurprisingly, the self-consistency-based reviewer outperforms the greedy decoding-based reviewer. Overall, the results draw attention to the importance of responsibly communicating LLM-generated rationales to humans and, consequently, instrumenting guardrails as an effective intervention strategy.

4.3 Orchestration Under Real-World Constraints

In real-world applications, RAG systems must operate under various constraints such as processing time, resource limitations, and compliance requirements. Efficient orchestration of RAG pipelines, involving the coordination of multiple processes (retrieval, generation, and post-processing), can be challenging but offers unique advantages. We propose a blueprint architecture [35] where the key orchestration concept is "streams" to coordinate the flow of data and instructions among components of varying compute requirements.

Key components in the blueprint architecture include (Figure 52): (1) agents, agent and data registries as key touch points and interfaces to seamlessly integrate with existing deployed models, APIs, databases, and services, (2) streams to orchestrate data and instructions across components, and (3) task and data planners to optimize for cost and quality constraints in task execution and data retrieval. It is designed for seamless integration into existing infrastructure, enabling extensibility, customizability, and reusability through well-defined touchpoints and interfaces. It supports externalized orchestration and flexible task coordination via declarative plans, ensuring observability, controllability, and optimized performance while meeting quality-of-service constraints.

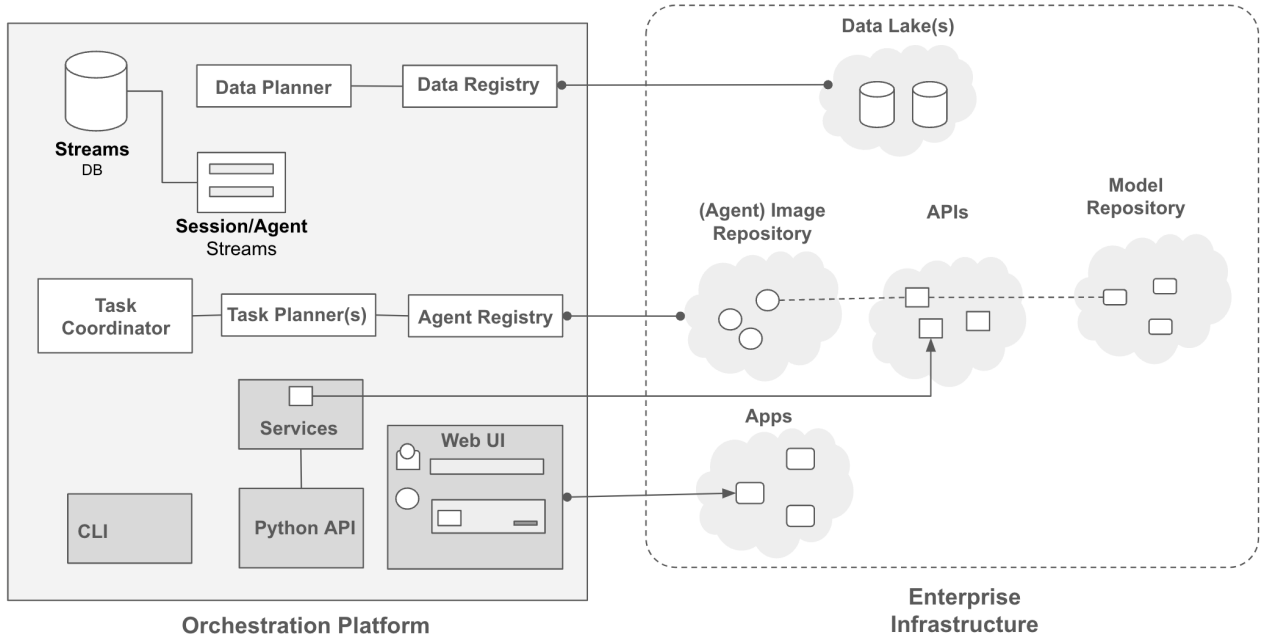


Figure 20: Blueprint Architecture: Data and Agent Registries are touch points that define existing data, models, APIs, and services in the enterprise for utilization by agents.

5 Challenges and Opportunities

The work discussed covers limited aspects of building an effective System 2 RAG solution. Numerous intriguing research questions remain unanswered in the fields of NLP, AI, databases, and HCI, presenting ample opportunities for interdisciplinary collaboration. Here, we will discuss some of these questions.

5.1 Planning and reasoning

Despite emerging attempts to explore LLMs' reasoning capabilities and use them as planners [55, 56] or 'routers' of existing tools and APIs [57–59], LLMs alone still cannot solve the planning problem [60, 61]. Key questions remain, including: How to exploit LLMs for planning, yet add verification and constraints? How to perform planning over multi-modal (relational, graph, documents, parametric) data sources? How to interact with the user in regards to planning, present and refine plans collaboratively? How to learn feedback and attribute back to agents and operators?

In addition, Optimization is critical for planning for production both as a driver of QoS and business-wise, as cost and performance affect the bottom line. Optimization is a well-studied subject, but new questions emerge: How to perform cost estimation for (new) agents, given the dependence on data (size and beyond)? How to handle uncertainty in sources such as LLMs? How to estimate the overall plan cost? How to incorporate an accrued budget into planners?

Additional future research opportunities in reasoning and decision-making systems exist in addressing the overhead introduced by deliberate, system 2 thinking processes, such as planning, which can be slow and cause notable differences in enterprise setups. A promising direction involves alleviating this burden by moving parts of the system 2 thinking process offline, continuously distilling and materializing knowledge into diverse representations. Metaphorically, this is akin to learning to drive: while initial skill acquisition requires deliberate system 2 reasoning, skilled drivers rely on system 1 instincts, reacting fluidly without explicitly thinking about each action, as their expertise becomes ingrained like muscle memory. Similarly, RAG systems can benefit significantly from insights derived from both online and offline learning processes. These insights can extend beyond traditional model weights to include artifacts like graphs, tables, and natural language documents. This calls for research in areas such as knowledge distillation, insight extraction, and planning, with a focus on understanding when and where to trust instinctual, system 1 insights versus when to engage in more rigorous, system 2 reasoning.

5.2 Multi-modal data

Enterprise environments often contain highly varied data sources, including databases, document collections, graphs, and structured tables with heterogeneous schemas. RAG systems designed for these environments must be capable of handling data from diverse formats while preserving contextual coherence across data types. They face challenges ranging from architectural and representational choices to managing ambiguity and uncertainty across modalities. How can RAG models balance capturing detailed, structured knowledge from tables or graphs with synthesizing general information from unstructured text? How should confidence levels and uncertainty be managed when retrieving from different data types? And how should the relevance of retrieved information be measured when dealing with multiple data types, given that existing metrics are often optimized for text-based retrieval?

5.3 From Data to Insights

A core opportunity in RAG systems lies in their ability to transform raw data into actionable insights. This insight-driven retrieval allows systems to dynamically generate responses tailored to user-specific needs or industry contexts. For example, RAG systems in human resources might leverage real-time job market statistics to enhance job recommendations, improving matches based on current industry trends. By deriving insights, systems can synthesize contextual knowledge from data, supporting more accurate and adaptive output generation.

However, research challenges remain. For instance, how can RAG systems accurately capture and prioritize real-time, evolving information from different data sources to ensure that insights remain relevant and current? In dynamic fields such as job searching, the timeliness and accuracy of insights

can be critical. Moreover, what methods can be developed to quantify and communicate the reliability or confidence level of synthesized insights to end users? Trustworthiness becomes especially important when RAG systems support high-stakes decision-making.

6 Conclusion

In this work, we explored the limitations of current Retrieval-Augmented Generation (RAG) models and proposed that a System 2 perspective should be adopted to address the challenges faced by LLMs in complex, domain-specific enterprise applications. Despite the advancements in integrating external information for grounding LLM outputs, we highlighted the shortcomings of existing RAG approaches, which often lack rigorous reasoning and deliberative analytics characteristic of System 2 thinking. Our analysis is based on the literature review and results obtained in previous work on different aspects of LLMs limitations and current RAG approaches, and it reinforces the necessity of transitioning from monolithic LLM architectures to compound AI systems, which employ specialized agents to enhance retrieval, ensure factual correctness, and mitigate issues like hallucination.

Based on the results already obtained by the previously described approaches that incorporate initial steps towards the System 2 type of thinking, we outlined a vision for the future, emphasizing the design of compound systems that better align with System 2 principles, featuring coordinated, logic-driven workflows capable of holistic reasoning and cross-document synthesis. While our work provides a foundational perspective for these advancements, there are still open questions about optimizing retrieval strategies, seamlessly integrating multiple data types, and fine-tuning decision-making modules. Addressing these challenges will be crucial for deploying robust, trustworthy AI systems that meet the high standards of reliability and precision required in enterprise contexts.

Acknowledgment

We thank the Editors of IEEE Data Engineering Bulletin for their valuable feedback: their suggestions immensely improved the quality of the article. Exploring these research problems required a significantly larger cast of characters than the author list in this paper does justice. We thank the co-authors of all the research papers cited in this article.

References

[1] J. Maynez, S. Narayan, B. Bohnet, and R. McDonald, "On faithfulness and factuality in abstractive summarization," in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, 2020, pp. 1906–1919.

[2] S. J. Semnani, V. Z. Yao, H. C. Zhang, and M. S. Lam, "Wikichat: Stopping the hallucination of large language model chatbots by few-shot grounding on wikipedia," arXiv preprint arXiv:2305.14292, 2023.

[3] Y. Feng, S. Rahman, A. Feng, V. Chen, and E. Kandogan, "Cmdbench: A benchmark for coarse-to-fine multimodal data discovery in compound ai systems," in Proceedings of the Conference on Governance, Understanding and Integration of Data for Effective and Responsible AI, 2024, pp. 16–25.

[4] M. Sclar, Y. Choi, Y. Tsvetkov, and A. Suhr, "Quantifying language models' sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting,"

in The Twelfth International Conference on Learning Representations, 2024. [Online]. Available: https://openreview.net/forum?id=RIu5lyNXjT

[5] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell et al., "Language models are few-shot learners," Advances in neural information processing systems, vol. 33, pp. 1877–1901, 2020.

[6] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou et al., "Chain-of-thought prompting elicits reasoning in large language models," Advances in neural information processing systems, vol. 35, pp. 24 824–24 837, 2022.

[7] P. Sahoo, A. K. Singh, S. Saha, V. Jain, S. Mondal, and A. Chadha, "A systematic survey of prompt engineering in large language models: Techniques and applications," arXiv preprint arXiv:2402.07927, 2024.

[8] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-efficient transfer learning for nlp," in International Conference on Machine Learning, 2019, pp. 2790–2799.

[9] Z. Qin et al., "Towards better parameter-efficient fine-tuning for large language models: A position paper," arXiv preprint arXiv:2311.13126, 2023.

[10] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray et al., "Training language models to follow instructions with human feedback," Advances in neural information processing systems, vol. 35, pp. 27 730–27 744, 2022.

[11] J. Thorne, A. Vlachos, C. Christodoulopoulos, and A. Mittal, "FEVER: a large-scale dataset for fact extraction and VERification," in Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), M. Walker, H. Ji, and A. Stent, Eds. New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 809–819. [Online]. Available: https://aclanthology.org/N18-1074

[12] R. Aly, Z. Guo, M. Schlichtkrull, J. Thorne, A. Vlachos, C. Christodoulopoulos, O. Cocarascu, and A. Mittal, "Feverous: Fact extraction and verification over unstructured and structured information," arXiv preprint arXiv:2106.05707, 2021.

[13] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel et al., "Retrieval-augmented generation for knowledge-intensive nlp tasks," Advances in Neural Information Processing Systems, vol. 33, pp. 9459–9474, 2020.

[14] D. Kahneman, Thinking, fast and slow. London: Penguin, 2012.

[15] Y. Bengio et al., "From system 1 deep learning to system 2 deep learning," in Posner lecture at NeurIPS'2019, Neural Information Processing Systems, Vancouver, BC, 2019.

[16] S. Maekawa, H. Iso, S. Gurajada, and N. Bhutani, "Retrieval helps or hurts? a deeper dive into the efficacy of retrieval augmentation to language models," in Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL), K. Duh, H. Gomez, and S. Bethard, Eds., 2024, pp. 5506–5521.

[17] H. Ding, L. Pang, Z. Wei, H. Shen, and X. Cheng, "Retrieve only when it needs: Adaptive retrieval augmentation for hallucination mitigation in large language models," arXiv preprint arXiv:2402.10612, 2024.

[18] Y. Zhang and Q. Yang, "A survey on multi-task learning," IEEE transactions on knowledge and data engineering, vol. 34, no. 12, pp. 5586–5609, 2021.

[19] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. Hruschka, and T. Mitchell, "Toward an architecture for never-ending language learning," in Proceedings of the AAAI conference on artificial intelligence, vol. 24, no. 1, 2010, pp. 1306–1313.

[20] B. Peng, M. Galley, P. He, H. Cheng, Y. Xie, Y. Hu, Q. Huang, L. Liden, Z. Yu, W. Chen et al., "Check your facts and try again: Improving large language models with external knowledge and automated feedback," arXiv preprint arXiv:2302.12813, 2023.

[21] A. Lazaridou, E. Gribovskaya, W. Stokowiec, and N. Grigorev, "Internet-augmented language models through few-shot prompting for open-domain question answering," arXiv preprint arXiv:2203.05115, 2022.

[22] F. Petroni, P. S. H. Lewis, A. Piktus, T. Rocktäschel, Y. Wu, A. H. Miller, and S. Riedel, "How context affects language models' factual predictions," in Conference on Automated Knowledge Base Construction, (AKBC), 2020.

[23] D. Li, A. S. Rawat, M. Zaheer, X. Wang, M. Lukasik, A. Veit, F. Yu, and S. Kumar, "Large language models with controllable working memory," in Findings of the Association for Computational Linguistics (ACL), 2023, pp. 1774–1793.

[24] K. Sun, Y. Xu, H. Zha, Y. Liu, and X. L. Dong, "Head-to-tail: How knowledgeable are large language models (LLMs)? A.K.A. will LLMs replace knowledge graphs?" in Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL), 2024.

[25] A. Mallen, A. Asai, V. Zhong, R. Das, D. Khashabi, and H. Hajishirzi, "When not to trust language models: Investigating effectiveness of parametric and non-parametric memories," in Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL), 2023.

[26] F. Petroni, T. Rocktäschel, S. Riedel, P. Lewis, A. Bakhtin, Y. Wu, and A. Miller, "Language models as knowledge bases?" in Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), K. Inui, J. Jiang, V. Ng, and X. Wan, Eds., 2019, pp. 2463–2473.

[27] C. Sciavolino, Z. Zhong, J. Lee, and D. Chen, "Simple entity-centric questions challenge dense retrievers," in Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP), M.-F. Moens, X. Huang, L. Specia, and S. W.-t. Yih, Eds., 2021, pp. 6138–6148.

[28] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing," ACM Computing Surveys, vol. 55, no. 9, pp. 1–35, 2023.

[29] S. Maekawa, H. Iso, and N. Bhutani, "Holistic reasoning with long-context lms: A benchmark for database operations on massive textual data," arXiv preprint arXiv:2410.11996, 2024.

[30] Z. Zhao, E. Wallace, S. Feng, D. Klein, and S. Singh, "Calibrate before use: Improving few-shot performance of language models," in International Conference on Machine Learning. PMLR, 2021, pp. 12 697–12 706.

[31] P. Pezeshkpour and E. Hruschka, "Large language models sensitivity to the order of options in multiple-choice questions," in Findings of the Association for Computational Linguistics: NAACL 2024, 2024, pp. 2006–2017.

[32] A. Mishra, S. Rahman, H. Kim, K. Mitra, and E. Hruschka, "Characterizing large language models as rationalizers of knowledge-intensive tasks," Findings of the Association for Computational Linguistics ACL 2024, 2024.

[33] R. R. Hoffman, S. T. Mueller, G. Klein, and J. Litman, "Metrics for explainable ai: Challenges and prospects," arXiv preprint arXiv:1812.04608, 2018.

[34] M. C. Stites, M. Nyre-Yu, B. Moss, C. Smutz, and M. R. Smith, "Sage advice? the impacts of explanations for machine learning models on human decision-making in spam detection," in International Conference on Human-Computer Interaction. Springer, 2021, pp. 269–284.

[35] E. Kandogan, S. Rahman, N. Bhutani, D. Zhang, R. Li Chen, K. Mitra, S. Gurajada, P. Pezeshkpour, H. Iso, Y. Feng et al., "A blueprint architecture of compound ai systems for enterprise," arXiv e-prints, pp. arXiv–2406, 2024.

[36] LinkedIn. Introducing hiring assistant for recruiter & jobs. [Online]. Available: https://business.linkedin.com/talent-solutions/hiring-assistant

[37] Databricks. Musings on building a generative ai product. [Online]. Available: https://www.microsoft.com/en-us/research/articles/magentic-one-a-generalist-multi-agent-system-for-solving-complex-tasks/

[38] Salesforce. Agentforce: Build the future with ai agents. [Online]. Available: https://www.salesforce.com/agentforce/

[39] Microsoft. Magentic-one: A generalist multi-agent system for solving complex tasks. [Online]. Available: https://www.microsoft.com/en-us/research/articles/magentic-one-a-generalist-multi-agent-system-for-solving-complex-tasks/

[40] Indeed. Indeed uses openai to deliver contextual job matching to millions of job seekers. [Online]. Available: https://openai.com/index/indeed/

[41] LinkedIn. Musings on building a generative ai product. [Online]. Available: https://www.linkedin.com/blog/engineering/generative-ai/musings-on-building-a-generative-ai-product

[42] A. M. Nambiar and D. Mundra, "An overview of data warehouse and data lake in modern enterprise data management," Big Data Cogn. Comput., vol. 6, no. 4, p. 132, 2022. [Online]. Available: https://doi.org/10.3390/bdcc6040132

[43] M. Armbrust, A. Ghodsi, R. Xin, and M. Zaharia, "Lakehouse: a new generation of open platforms that unify data warehousing and advanced analytics," in Proceedings of CIDR, vol. 8, 2021, p. 28.

[44] M. Bhandari, P. N. Gour, A. Ashfaq, P. Liu, and G. Neubig, "Re-evaluating evaluation in text summarization," in Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics, Nov. 2020, pp. 9347–9359.

[45] A. R. Fabbri, W. Kryściński, B. McCann, C. Xiong, R. Socher, and D. Radev, "SummEval: Re-evaluating summarization evaluation," Transactions of the Association for Computational Linguistics (TACL), vol. 9, pp. 391–409, 2021.

[46] N. F. Liu, K. Lin, J. Hewitt, A. Paranjape, M. Bevilacqua, F. Petroni, and P. Liang, "Lost in the middle: How language models use long contexts," arXiv preprint arXiv:2307.03172, 2023. [Online]. Available: https://arxiv.org/abs/2307.03172

[47] Y. Wu, H. Iso, P. Pezeshkpour, N. Bhutani, and E. Hruschka, "Less is more for long document summary evaluation by llms," in Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers), 2024, pp. 330–343.

[48] H. Y. Koh, J. Ju, H. Zhang, M. Liu, and S. Pan, "How far are we from robust long abstractive summarization?" in Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2022, pp. 2682–2698.

[49] K. Krishna, E. Bransom, B. Kuehl, M. Iyyer, P. Dasigi, A. Cohan, and K. Lo, "LongEval: Guidelines for human evaluation of faithfulness in long-form summarization," in Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, 2023, pp. 1650–1669.

[50] O. Khattab and M. Zaharia, "Colbert: Efficient and effective passage search via contextualized late interaction over bert," in Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, 2020, pp. 39–48.

[51] S. Zhuang, H. Zhuang, B. Koopman, and G. Zuccon, "A setwise approach for effective and highly efficient zero-shot ranking with large language models," arXiv preprint arXiv:2310.09497, 2023.

[52] Z. Qin, R. Jagerman, K. Hui, H. Zhuang, J. Wu, J. Shen, T. Liu, J. Liu, D. Metzler, X. Wang et al., "Large language models are effective text rankers with pairwise ranking prompting," arXiv preprint arXiv:2306.17563, 2023.

[53] P. Pezeshkpour and E. Hruschka, "Multi-conditional ranking with large language models," arXiv preprint arXiv:2404.00211, 2024.

[54] X. Wang, J. Wei, D. Schuurmans, Q. V. Le, E. H. Chi, S. Narang, A. Chowdhery, and D. Zhou, "Self-consistency improves chain of thought reasoning in language models," in The Eleventh International Conference on Learning Representations.

[55] Z. Zhao, W. S. Lee, and D. Hsu, "Large language models as commonsense knowledge for large-scale task planning," Advances in Neural Information Processing Systems, vol. 36, 2024.

[56] W. Huang, P. Abbeel, D. Pathak, and I. Mordatch, "Language models as zero-shot planners: Extracting actionable knowledge for embodied agents," in International conference on machine learning. PMLR, 2022, pp. 9118–9147.

[57] Y. Liang, C. Wu, T. Song, W. Wu, Y. Xia, Y. Liu, Y. Ou, S. Lu, L. Ji, S. Mao et al., "Taskmatrix. ai: Completing tasks by connecting foundation models with millions of apis," Intelligent Computing, vol. 3, p. 0063, 2024.

[58] Y. Qin, S. Liang, Y. Ye, K. Zhu, L. Yan, Y. Lu, Y. Lin, X. Cong, X. Tang, B. Qian et al., "Toolllm: Facilitating large language models to master 16000+ real-world apis," arXiv preprint arXiv:2307.16789, 2023.

[59] S. Kim, S. Moon, R. Tabrizi, N. Lee, M. Mahoney, K. Keutzer, and A. Gholami, "An llm compiler for parallel function calling," arXiv, 2023.

[60] S. Kambhampati, K. Valmeekam, L. Guan, K. Stechly, M. Verma, S. Bhambri, L. Saldyt, and A. Murthy, "Llms can't plan, but can help planning in llm-modulo frameworks," arXiv preprint arXiv:2402.01817, 2024.

[61] K. Valmeekam, M. Marquez, S. Sreedharan, and S. Kambhampati, "On the planning abilities of large language models-a critical investigation," Advances in Neural Information Processing Systems, vol. 36, pp. 75 993–76 005, 2023.