Computers and Education: Artificial Intelligence 10 (2026) 100578

Contents lists available at ScienceDirect

Computers and Education: Artificial Intelligence

journal homepage: www.sciencedirect.com/journal/computers-and-education-artificial-intelligence

Enhancing large language model performance for automatic zero-shot multiple-choice question answering via single-token logit prompting

Quoc Phu Dang [a,b,1], Phat T. Tran-Truong [a,b,1], Duc-Ly Vu [c], Long S. T. Nguyen [a,b], Quynh T. N. Vo [a,b], Tho Quan [a,b,*]

[a] Faculty of Computer Science and Engineering, Ho Chi Minh City University of Technology (HCMUT), 268 Ly Thuong Kiet Street, Dien Hong Ward, Ho Chi Minh City, Viet Nam

[b] Vietnam National University Ho Chi Minh City, Linh Xuan Ward, Ho Chi Minh City, Viet Nam

[c] Faculty of Information Technolgy, Eastern International University, Nam Ky Khoi Nghia Street, Binh Duong Ward, Ho Chi Minh City,Viet Nam

A R T I C L E I N F O

2000 MSC:
0000
1111

PACS:
0000
1111

Keywords:
Large language model
Multiple choice question answering
Education
Retrieval-augmented generation
Prompting engineering

A B S T R A C T

While Large Language Models (LLMs) offer significant potential for educational applications, they exhibit distinct limitations when answering multiple-choice questions (MCQs). Because LLMs are optimized for autoregressive token prediction, their performance degrades substantially when answer choices are simply shuffled—a phenomenon known as the Multiple-Choice Symbol Binding (MCSB) limitation. To mitigate this, we introduce a novel prompting technique called Single-Token Logit (STL). Instead of evaluating the output logits of all answer labels, STL extracts and normalizes the logit value of a single token type (specifically "yes") to independently verify each option. We comprehensively evaluate STL against established baselines, including Labels Token Logits (LTL) and Chain-of-Thought (CoT), across the ARC, OpenBookQA, and SciQ datasets. In almost all configurations, STL matches or outperforms the standard baseline (LTL)—yielding gains of up to 11 percentage points—at a moderate computational overhead (1.58× latency and 3.72× GPU memory relative to LTL). Furthermore, sample-by-sample McNemar's testing ($\alpha = 0.05$) confirms STL is statistically superior to LTL and highly competitive against the computationally expensive CoT method. Finally, we demonstrate STL's robustness in knowledge-intensive environments by integrating it with Retrieval-Augmented Generation (RAG), where it achieves up to 81.06% accuracy on the combined ARC dataset with Mistral 7B—a 9.36 percentage point gain over the original no-context baseline (LTL) of 71.7%.

1. Introduction

Multiple-choice questions (MCQs; see Fig. 1) are a widely used assessment tool in education. A typical MCQ consists of a stem followed by four options, exactly one of which is correct. To be effective, MCQs should include plausible incorrect options (distractors) that probe understanding rather than rote recall. Their versatility allows test-makers to assess diverse subjects and learner levels. While test-makers traditionally design and score MCQs manually, recent advances have led to automatic and semi-automatic computer-assisted tools that support educators in these tasks. For example, Mitkov et al. (2006) proposed a semi-automatic system to generate MCQ tests from input documents, and Alomran and Chai (2018) devised an automatic scoring system that delivers results to examinees. However, these systems often require substantial human intervention, limiting flexibility and efficiency. They may also demand a deep understanding of the input documents for tasks such as defining corpora and ontologies (Mitkov et al., 2006). Moreover, because many pipelines rely on predefined vocabularies and formats, they risk encouraging rote learning. These limitations stem from hard-coded designs that lack the ability to generalize, infer, and generate new content.

Recent advancements in Artificial Intelligence (AI), particularly Large Language Models (LLMs) powered by foundation models such as the GPT series (Brown et al., 2020; Radford et al., 2019), LLaMA-2

* Corresponding author at: Faculty of Computer Science and Engineering, Ho Chi Minh City University of Technology (HCMUT), 268 Ly Thuong Kiet Street, Dien Hong Ward, Ho Chi Minh City, Viet Nam.

Email addresses: quoc.dang2022@hcmut.edu.vn (Q.P. Dang), phatttt@hcmut.edu.vn (P.T. Tran-Truong), ly.vu@eiu.edu.vn (D.-L. Vu), long.nguyencse2023@hcmut.edu.vn (L.S.T. Nguyen), quynh.vo01@hcmut.edu.vn (Q.T.N. Vo), qttho@hcmut.edu.vn (T. Quan).

[1] Same contribution.

https://doi.org/10.1016/j.caeai.2026.100578
Received 1 January 2025; Accepted 10 March 2026

Available online 14 March 2026

Question: Which of these is not an inherited trait in humans?
Options:
A. Height
B. Hair color
C. Skin color
D. Intelligence
Answer: D

Fig. 1. A multiple choice question (MCQ) example from ARC (AI2 reasoning challenge) dataset (Clark et al., 2018).

(Touvron et al., 2023), or Mistral (Jiang et al., 2023), offer promising potential for assisting in the design of MCQ exams. The superior linguistic understanding capabilities of LLMs can significantly aid educators in the automatic generation and answering of MCQs. Specifically, LLMs can diversify MCQs in terms of topics, input documents, and incorrect answer choices while requiring minimal human intervention. The generated MCQs not only emulate natural human language but also incorporate plausible incorrect options, challenging test-takers to demonstrate a profound understanding of the subject matter.

Furthermore, LLMs can be integrated with external tools like Retrieval-Augmented Generation (RAG) and function calls to enhance their capabilities as autonomous AI agents (Wang et al., 2024a). This integration significantly amplifies the LLM's ability to generate MCQs. Additionally, LLMs can effectively utilize short-term and long-term memory for refining MCQs and managing question banks, enabling them to generate MCQs with varying difficulty levels. When tasked with answering MCQs, LLMs excel as automatic grading tools, accurately scoring responses and providing prompt feedback to students, including explanations for both correct and incorrect options. Moreover, LLMs can analyze student responses to identify knowledge gaps and errors, empowering educators to tailor their instruction accordingly. This surpasses the capabilities of traditional computer-aided assessment systems (Alomran & Chai, 2018; Mitkov et al., 2006).

Despite their potential, LLMs exhibit limitations in understanding MCQA tasks. As demonstrated by (Dominguez-Olmedo et al., 2024; Wang et al., 2024b), LLMs often exhibit a preference for certain answer options, such as the first option "A". This selection bias significantly impacts performance, as LLMs tend to excel when correct answers are placed in preferred positions, but degrade substantially otherwise. Additionally, as explored by Wang et al. (2024c), LLMs may output labels that misrepresent their knowledge, especially when required to provide short text responses. The generated text can significantly misalign with the intended meaning of their response label.

These phenomena can be attributed to the autoregressive nature of LLMs, which generate text based on probabilities (Lyu et al., 2024). LLMs may prioritize generating the most probable token (e.g., "A") without fully considering the meaning of the answer options and their connection with associated labels (Zheng et al., 2024). When trained or fine-tuned on datasets with frequent occurrences of a specific token (e.g., "A"), LLMs may have a tendency to select that option, even if it is incorrect.

This can lead to limitations in understanding MCQA tasks, as LLMs may struggle to associate answer options with their corresponding labels. As demonstrated by (Pezeshkpour & Hruschka, 2024), rearranging the answer choices can lead to incorrect selections, even if the model initially grasped the correct response. This limitation, known as Multiple-Choice Symbol Binding (MCSB) ability (Robinson & Wingate, 2023), can hinder the effectiveness of LLMs in MCQA tasks. Fig. 2 illustrates this phenomenon, where the LLM initially chooses the correct answer but fails to recognize it when the answer choices are shuffled. This is particularly problematic as test-makers often rearrange answer choices in MCQs to create different test versions.

These observations spurred us to explore methods to enhance LLM performance in answering MCQs. Accordingly, our first research question is:

RQ1 *How can we effectively address the MCSB limitation in LLMs?*

Current dominant approaches for MCQA tasks (e.g., Hendrycks et al. (2020); Santurkar et al. (2023)) often bundle the question and answer choices in a single prompt, expecting LLMs to select the correct answer by label (A, B, C, etc.). The Fig. 3 illustrates these approaches, where a prompt for answering an MCQ (with or without instructions) is sent to the LLM for outputting the correct option. This approach exacerbates the MCSB limitation, which is intrinsic and varies significantly across LLM architectures (Robinson & Wingate, 2023; Zheng et al., 2024) . While some LLMs (such as OpenAI Codex (Chen et al., 2021)) may effectively handle such prompts, this capability is not guaranteed across all domains and tasks (Robinson & Wingate, 2023). Additionally, as explored by (Zheng et al., 2024), simple prompting strategies like Chain-of-Thought prompting techniques (Kojima et al., 2022; Wei et al., 2022b) are not capable of mitigating this phenomenon.

To address this challenge, we propose a novel prompting technique called *Single-Token Logit* (illustrated by Fig. 4). Our approach involves creating separate prompts for each answer choice and the question, followed by sending each prompt individually to the LLMs.

Another challenge that LLMs are facing in answering MCQs is that LLMs are trained on massive datasets, but these datasets may not always be entirely up-to-date or accurately representative of real-world information (He et al., 2022). For example, an LLM trained on a dataset from 2023 backwards, cannot know events afterward, and may then output incorrect answers with present-day facts. Furthermore, inaccuracies and potential biases within the training data can further degrade the accuracy (Chen et al., 2024a).

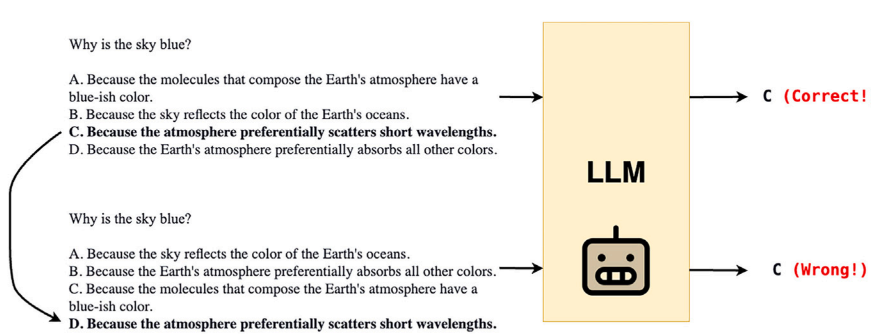

Fig. 2. Multiple-choice symbol binding (MCSB) limitation in LLMs.

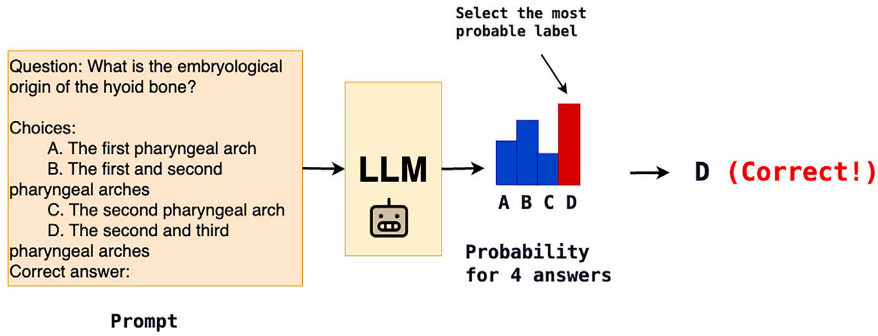

Fig. 3. Overview of approaches for answering MCQs by a single prompt.

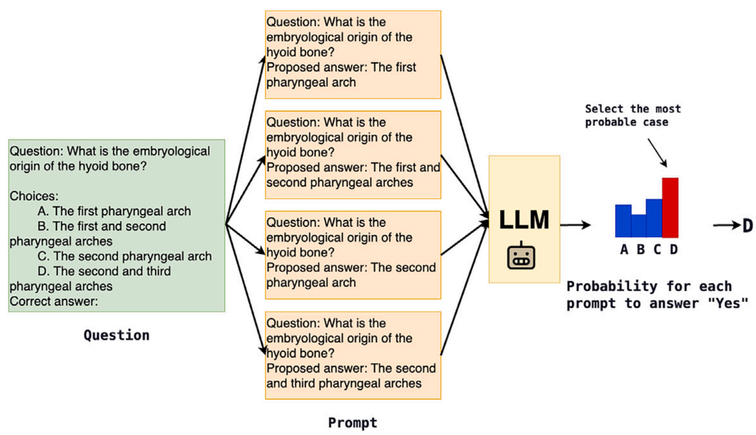

Fig. 4. Overview of our single-token logit approach.

To address these limitations, Retrieval-Augmented Generation (RAG) (Lewis et al., 2020) has been widely adopted to supplement LLMs with relevant external information. Because real-world educational deployments increasingly combine LLMs with retrieval-based context enrichment, evaluating STL under these conditions constitutes a necessary stress test of its robustness (Li et al., 2025). Therefore, our second research question is:

> **RQ2** *How robust is the proposed STL method across diverse operational environments, specifically when models are supplemented with varying levels of external knowledge?*

To answer this question, we integrate RAG systems with LLMs under multiple context conditions—including no external context (baseline), standard retrieval, and an enhanced query approach where the LLM itself acts as a knowledge generator to formulate richer retrieval queries. These three conditions can simulate operational environments of increasing information richness, allowing us to stress-test whether STL's advantages hold across realistic, knowledge-intensive scenarios rather than being limited to the no-context setting.

In summary, by mitigating the MCSB limitation and demonstrating robustness under context-enhanced conditions, our proposed approach empowers test-makers to utilize LLMs more effectively and with greater confidence for various MCQA-related tasks. Our article makes the following contributions:

- We propose a prompting technique called *Single-Token Logit* to address the MCSB limitation of LLMs when tackling MCQs.
- We validate the robustness of our STL method across diverse operational environments by integrating it with Retrieval-Augmented Generation (RAG) under multiple context conditions—from no external context to enhanced query retrieval—demonstrating that STL maintains its performance advantage in realistic, knowledge-intensive scenarios.
- We perform a rigorous experimental evaluation of our approach on the benchmark (Hendrycks et al., 2020). The results obtained demonstrate the significant contribution of our method in improving LLM performance for MCQA tasks .[2]

The remainder of the paper is organized as follows: Section 2 provides relevant background information. Section 3 summarizes related work, highlighting our contributions. Section 4 outlines our proposed solution. Section 5 describes the experimental setup. Finally, Section 7 concludes the paper. Appendix A provides sample questions aimed at improving predictions.

2. Background

2.1. Text embedding models

Text embedding is the process of converting textual data into dense, multi-dimensional vector representations while preserving the semantic and contextual meaning of the text. This allows for measuring the semantic and grammatical similarity between different text segments. The primary application of measuring these similarities lies in tasks like

[2] The source code for reproduction is available at https://github.com/LeoTTTPhat/STL-Prompting-For-ZeroShot-MCQA.

text retrieval, where text segments are ranked based on their semantic or contextual similarity. Additionally, embedding vector representations are also applied in other NLP tasks such as sentiment analysis, text classification, and machine translation. The field of text embedding encompasses various models, including Word2Vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014), FastText (Bojanowski et al., 2017), ELMo (Peters et al., 2018), and BERT (Devlin et al., 2019), each with its strengths in representing the semantic meaning of text.

The remarkable advancement of LLMs has ignited significant interest in text embedding models. Among the various embedding models, BERT-based models have gained widespread adoption and are at the forefront of active research and development. Numerous advancements and diverse training procedures have emerged to generate the most effective text embedding vectors. Prominent examples include Sentence-BERT (Reimers et al., 2019), DPR (Karpukhin et al., 2020), BAAI general embedding (BGE) (Xiao et al., 2024), and a growing range of other models. In this work, the proposed solution leverages the BGE technique (Xiao et al., 2024) for embedding text as detailed in Section 5.

2.2. Language models and prompting techniques

Traditionally, natural language processing (NLP) relied on neural network architectures like Recurrent Neural Networks (Goodfellow et al., 2016), Long Short-Term Memory (Hochreiter & Schmidhuber, 1997) or Gated Recurrent Unit (Chung et al., 2014), often combined with sequence-to-sequence models (Sutskever et al., 2014). These architectures required training or fine-tuning for specific tasks.

The advent of foundation models (Bommasani et al., 2021) has revolutionized NLP. These models empower LLMs with state-of-the-art capabilities across a wide range of tasks, from linguistic skills and commonsense reasoning to mathematics and coding. This allows individuals to leverage LLMs for diverse tasks without specialized training in each specific domain.

However, for LLMs to achieve domain-specific expertise, prompting techniques (Liu et al., 2023) are essential. These techniques guide LLMs in in-context learning (Wei et al., 2022a) by providing specific instructions or contextual information. Examples include models like BloombergGPT for finance (Wu et al., 2023), CodeLlama for coding (Roziere et al., 2023), PharmGPT for bio-pharmaceuticals (Chen et al., 2024b), and AstroLLaMA for astronomy (Perkowski et al., 2024).

> In this work, we demonstrate the effectiveness of prompting techniques for enhancing the performance of two foundation models: LLaMA2 (Touvron et al., 2023) and Mistral (Jiang et al., 2023) in MCQA tasks.

2.3. Retrieval-augmented generation (RAG)

While LLMs represent a significant advancement in NLP and AI in general, they are not without limitations. One of the primary challenges concerns their accuracy and factuality. Regarding accuracy problems, LLMs are trained on massive datasets, but these datasets may not always be entirely up-to-date or perfectly representative of real-world information. Additionally, the veracity and potential biases within the training data can further degrade accuracy. Regarding the factuality problem, LLMs can struggle with factuality, particularly when presented with queries outside the scope of their training data. This can lead to the generation of incorrect or misleading responses, a phenomenon referred to as *hallucination* (Huang et al., 2023; Levine et al., 2022). The Retrieval-Augmented Generation (RAG) approach has been proposed (Lewis et al., 2020) to address this limitation by integrating the capabilities of LLMs with external knowledge retrieval and integration. While traditional language models excel at generating coherent and contextually relevant responses, they cannot access and leverage up-to-date or explicitly cited information (He et al., 2022). RAG overcomes this limitation by enabling LLMs to retrieve relevant contextual documents from external sources during the response generation process (as illustrated in Fig. 5). RAG consists of two main components: a retrieval component and a response generation component. The basic RAG process is described below:

- Retrieval: This component allows the model to search for and access relevant information from a vast external knowledge source. It enhances the model's ability to retrieve accurate and contextually relevant information, beyond what it has been trained on.
- Augment: The user's query and the retrieved information are augmented and combined into a comprehensive prompt.
- Generation: Building upon the foundation of LLMs, the model takes in the prompt, which includes the query and the retrieved information, and generates the response. This ensures that the generated text is not only contextually rich but also informationally accurate.

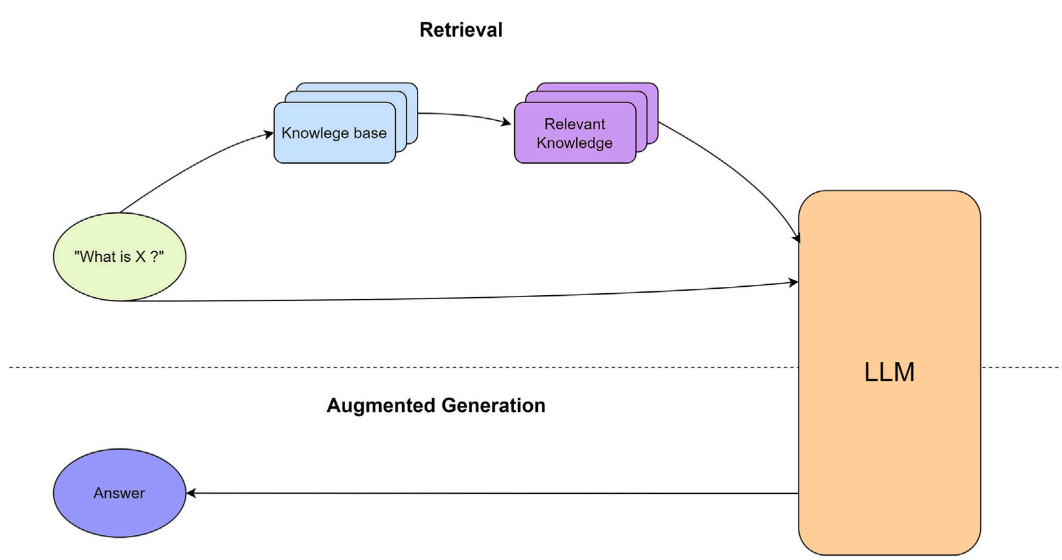

Fig. 5. Overview of RAG techniques.

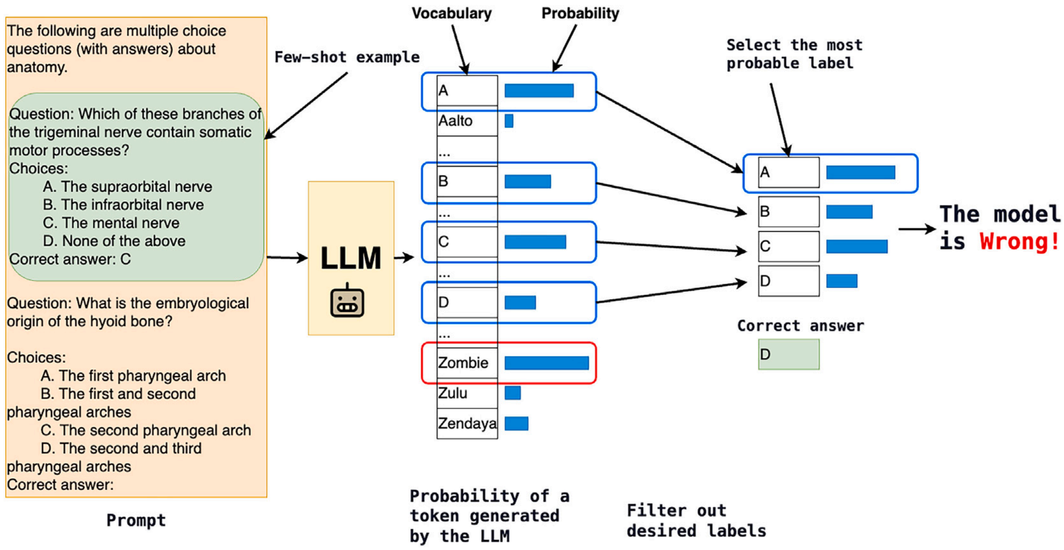

Fig. 6. Overview of LLM's MCQA approach by Hendrycks et al. (2020).

In essence, RAG augments the input provided to an LLM with knowledge retrieved from external sources. This retrieval step allows the LLM to incorporate the most pertinent information into its generated responses. By leveraging external knowledge, RAG enhances the accuracy and reliability of the LLM's output, making it particularly valuable for tasks such as question answering and information retrieval.

In the original RAG study (Lewis et al., 2020), the authors applied a vector-based retrieval method generated from the Dense Passage Retrieval (DPR) model (Karpukhin et al., 2020). DPR is a separate study on a model that generates embedding vectors for text, using the BERT model architecture. The results showed that the text retrieval quality from DPR's embedding vectors outperformed the BM25 algorithm (Robertson & Zaragoza, 2009), a common traditional text retrieval algorithm.

3. Related works

BERT-Based Models for Reading Comprehension Multiple Choice (Xu et al., 2019): The authors fine-tuned the BERT model to address multiple-choice reading comprehension (MC-RC) tasks. They augmented the original BERT with a Deep Comatch Network (DCN), which outperformed the baseline on the RACE dataset (Lai et al., 2017). The DCN features a novel comatch attention layer placed after BERT's five coattention layers and encoding layer. Fine-tuning standard BERT achieved 62.2% accuracy, whereas the DCN variant reached 66.2%. An ensemble of all models further improved this to 67.9% on RACE.

Enhancing Question Answering with External Knowledge (Pan et al., 2019): Led by Xiaoman Pan, the team fine-tuned BERT for MC-RC by integrating two strategies to incorporate external knowledge. The first retrieved pertinent Wikipedia snippets via Lucene using TF-IDF. The second enriched training data with question patterns from various topics. Experiments used ARC-Easy, ARC-Challenge (Clark et al., 2018), and OpenBookQA (Mihaylov et al., 2018a), yielding accuracy gains of up to 8.1%, 13.0%, and 12.8% over baselines without these enhancements.

> Answering knowledge-based questions effectively relies on robust information retrieval and strong reasoning within the model. Beyond BERT for MC-RC, other large language models (LLMs) offer text generation and multi-tasking, including GPT (Yenduri et al., 2024), T5 (Raffel et al., 2020), and the recent LLaMA2 (Touvron et al., 2023).

Measuring Massive Multitask Language Understanding (Hendrycks et al., 2020): The authors presented a comprehensive dataset assessing LLMs via multiple-choice questions (MCQs) across diverse domains and difficulty levels (Hendrycks et al., 2020). They also released a framework to facilitate LLM responses to MCQs (Hendrycks, 2020).

As shown in Fig. 6, the method formats the question and choices as a prompt, assigning labels (e.g., "A", "B") to each option. The prediction uses the first generated token post-prompt. Rather than picking the highest-probability token from the full vocabulary, it focuses on logits for "A", "B", "C", and "D", ignoring others and normalizing their probabilities.

The probability for each choice is:

$$P(a_i) = \frac{\exp(l_{a_i})}{\sum_{j \in V} \exp(l_j)} \tag{1}$$

where:

- A is the set of answer choices (e.g., $A = \{$"A", "B", "C", "D"$\}$).
- $a_i \in A$ is an individual choice.
- l_i is the logit for token "i" from the LLM.
- V is the complete vocabulary.

LLMs might generate unintended tokens like "Zombie" instead of labels, but this filters to label logits only, ensuring a label output. To encourage label generation, few-shot examples in training can illustrate the format.

Per Eq. (1), this logit-based approach is termed Label Token Logits (LTL) for brevity.

Zero-Shot Chain-of-Thought Prompting (Kojima et al., 2022): The authors introduced zero-shot Chain-of-Thought (CoT) prompting, a method that elicits multi-step reasoning in LLMs by simply appending "Let's think step by step" to a query. This approach generates a transparent reasoning trace before the final answer, entirely without task-specific exemplars. Its superiority over standard zero-shot prompting has been demonstrated across diverse benchmarks, including arithmetic and commonsense reasoning.

In the context of MCQ tasks, zero-shot CoT prompts the model to reason through the problem and conclude with the correct label (e.g., 'A', 'B'). The answer is thus explicitly dictated by the preceding rationale within the generated text. This paradigm is fundamentally different from logit-based methods: it renders the extraction of token probabilities (logits) meaningless, as the answer is not selected from a vocabulary but is instead a conclusion reached through generated language. The final output must therefore be parsed textually.

This represents a paradigm shift in prompt engineering for MCQs, emphasizing auditable reasoning processes over opaque, probabilistic token selection. The value of zero-shot CoT lies in this dual benefit: it enhances accuracy while providing a window into the model's "thought process". This stands in stark contrast to methods like LTL, which offer no such interpretability, making CoT particularly critical for educational and high-stakes evaluative applications where justification is as important as the answer itself.

Given the efficacy of these prompting strategies for MCQ performance in LLMs, we adopt LTL and CoT as benchmarks to gauge our proposed method's improvements in Section 5.

4. Proposed approach

This paper proposes two approaches to improve LLMs' ability to answer MCQs. In Section 4.1, we present a new prompt engineering technique called *Single-Token Logit Prompting* to address the issue of inconsistent answering when the option positions change in MCQs. Notably, this LLM-based approach does not depend on the output values of the label tokens. Our second contribution leverages a customized RAG system for improved context retrieval, as detailed in Section 4.3. This system utilizes knowledge derived from the question rather than from the raw question to achieve more accurate retrieval. The effectiveness of these methodologies is rigorously evaluated in the Section 5.

4.1. Single-token logit prompting technique

Our approach eliminates the need for output logit values associated with label tokens (e.g., "A", "B", "C", "D", etc.) by leveraging the normalized logit value of a single token type (specifically, the "yes" token in the current implementation). Consider a prompt as exemplified in Fig. 7. By analyzing the LLM's output at the position of the first token, we can determine the logit value for any token in the LLM's vocabulary. Capitalizing on this principle, we use the normalized logit value of the "yes" token to represent the model's agreement with a chosen answer presented in a "yes" or "no" formatted prompt.

As demonstrated in Fig. 8, for a question with k answer choices, each option is converted into a corresponding prompt. Each prompt incorporates the question content along with a single corresponding answer choice. More specifically, each prompt includes the question and one of the k answer choices, and the LLM is instructed to respond with either "yes" or "no", signifying its agreement with the presented answer choice. Using only a single logit value from one prompt is insufficient to generate meaningful predictions. By generating k analogous prompts, one for each answer choice, we can obtain k corresponding logit values, as presented in Eq. (3). By comparing these values and applying the softmax function, we can determine the probabilities associated with each of the model's answer choices, as expressed in Eq. (4).

The normalized logit for the "yes" token is defined as:

$$\tilde{l}_{\text{yes}}(q, a_i) = \log\left(\frac{\exp(l_{\text{yes},i})}{\sum_{v\in V}\exp(l_{v,i})}\right) \tag{2}$$

where V is the entire vocabulary, $l_{\text{yes},i}$ is the raw logit of the "yes" token for answer choice a_i, and $l_{v,i}$ represents the raw logit of token v for answer choice a_i.

The formula for the probability of each prompt pass can be expressed as:

$$P(a_i|q) = \frac{\exp(\tilde{l}_{\text{yes}}(q, a_i))}{\sum_{a_j\in A}\exp(\tilde{l}_{\text{yes}}(q, a_j))} \tag{3}$$

where:

- q is the question content.
- A is the set of answer choices (eg. $A = \{$"A", "B", "C", "D"$\}$).
- $a_j \in A$ represents an answer choice within A.
- $\tilde{l}_{\text{yes}}$ is the normalized logit value of the token "yes".

Additionally, we can define c as the supplementary context of the question, if available. For example, in our work, supplementary context could

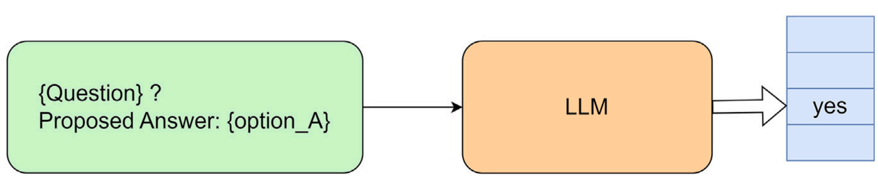

Fig. 7. Calculate the normalized logit value of a token.

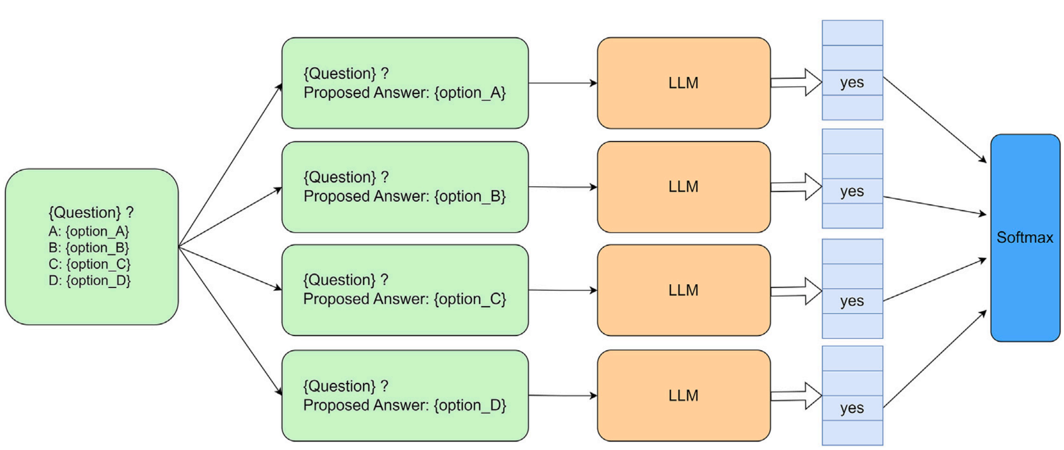

Fig. 8. Single token Logits approach.

be the context retrieval enhancement. All four passes of prompts are processed before extracting the answered label using the argmax function. The final predicted result is defined as:

$$\text{Answered label} = \arg\max_{a \in A} P(a|q,c) \tag{4}$$

As expressed in Eq. (3), since our approach only considers the "yes" token normalized logit score in a calculation, we refer to it as Single Token Label (STL) for the ease of reference.

4.2. Prompt template design for LLM approaches

In this work, we use the prompting technique proposed by Hendrycks et al. (2020) (we call it LTL) as a baseline. This section details the prompt design employed for both the baseline (Hendrycks et al., 2020) and the proposed LLM approaches.

4.2.1. The baseline approach

The baseline approach leverages prompts that incorporate the question itself alongside all answer options, including their designated labels. To facilitate improved task comprehension by the LLM, an instructional component is prepended to the prompt, explicitly outlining the task objective.

LTL Prompt
Your task is to analyze the question and answer options A, B, C or D below.
Question: …
Options: …
A. …
B. …
C. …
D. …
Answer: *LLM_completes_here*

An example of LTL Prompt
Your task is to analyze the question and the answer options A, B, C or D below.
Question: The primary cause of continental drift, earthquakes, and volcanic eruptions is
A. convection currents beneath Earth's crust.
B. the rotation of Earth on its axis.
C. the gradual sinking of Earth's crust.
D. heat from the Sun warming Earth.
Answer:

4.2.2. Chain of thought (CoT) prompting technique

We employ the Chain of Thought (CoT) prompting technique, which enables the LLM to reason step by step before selecting the final answer from the provided options. The prompt includes detailed rules for structuring the response and an illustrative example to guide the LLM. To ensure the model adheres to the strict format for extracting the label (Answer: [A/B/C/D]), which can be challenging for the LLM to follow consistently, we also include a detailed example within the prompt. This differs from other methods that rely on logit values for selection.

CoT Prompt
Your task is to analyze the question and answer options A, B, C, or D below. Rules: First, write your reasoning step by step. - Do NOT state the final answer inside the reasoning. - After the reasoning is fully complete, on a new line, write ONLY the final answer in the exact format: Answer: [A/B/C/D] - Do not add any words, explanations, or text after the final answer line. — Now answer this question:
Question: …
A. …
B. …
C. …
D. …

An example of CoT Prompt
Your task is to analyze the question and answer options A, B, C, or D below. Rules: First, write your reasoning step by step. - Do NOT state the final answer inside the reasoning. - After the reasoning is fully complete, on a new line, write ONLY the final answer in the exact format: Answer: [A/B/C/D] - Do not add any words, explanations, or text after the final answer line. — Now answer this question:
Question: The primary cause of continental drift, earthquakes, and volcanic eruptions is
A. convection currents beneath Earth's crust.
B. the rotation of Earth on its axis.
C. the gradual sinking of Earth's crust.
D. heat from the Sun warming Earth.

4.2.3. STL prompting technique

In contrast, the proposed approach utilizes prompts that solely include the question content and a single answer choice. The LLM is then instructed to read and comprehend the provided information. Subsequently, the LLM is tasked with responding with "yes" if it agrees with the presented answer choice or "no" if it disagrees.

STL Prompt
Your task is to analyze the question and answer below. If the answer is correct then respond yes, if it is not correct then respond no.
Question: …
Proposed answer: …
Response: *LLM_completes_here*

An example of STL Prompt
Your task is to analyze the question and answer below. If the answer is correct then respond yes, if it is not correct then respond no.
Question: An astronomer observes that a planet rotates faster after a meteorite impact. What is the most likely effect of this increase in rotation?
Proposed answer: Planetary density will decrease.
Response:

4.3. Context retrieval

In real-world scenarios, question-answering systems often have access to external knowledge sources through retrieval mechanisms. To evaluate our proposed method under such rich contextual conditions, we integrate a Retrieval-Augmented Generation (RAG) system into our experimental framework. This allows us to assess how different prompting approaches perform when supplemented with relevant background information, rather than relying solely on the model's parametric knowledge.

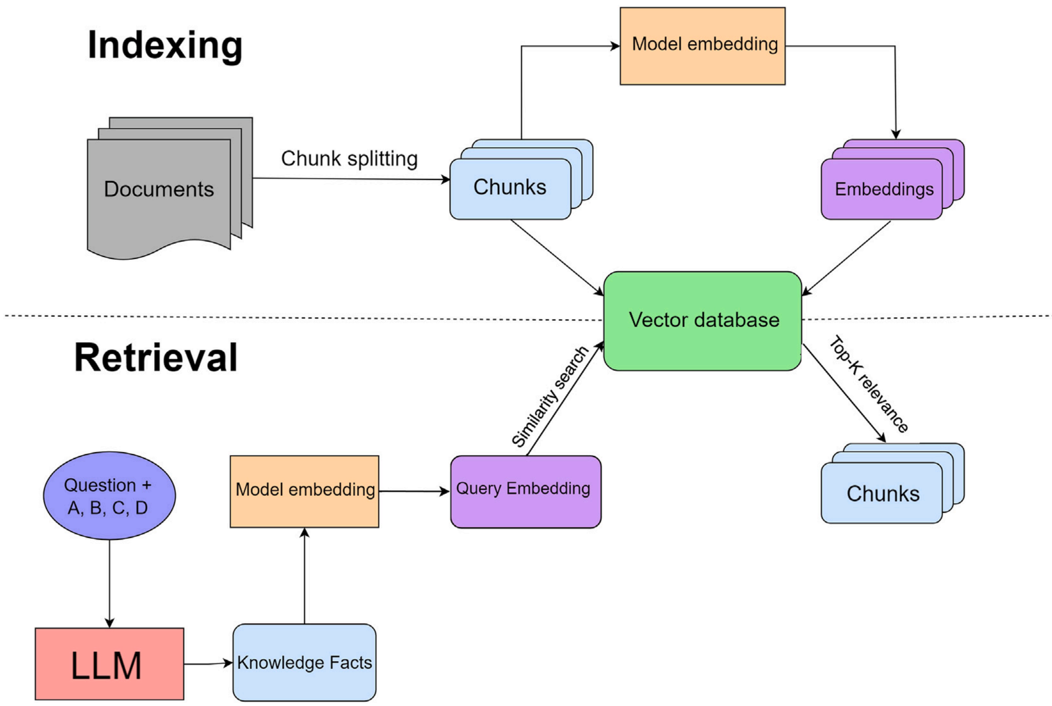

Fig. 9. Context retrieval organization.

Our retrieval system employs an efficient context retrieval mechanism as illustrated in Fig. 9. This system involves the following three key components:

- Document Sources: Textual data is obtained from various sources and segmented into smaller units, typically referred to as *chunks* (e.g., 100 words).
- Text Embedding Models: These models transform each chunk of text into a corresponding vector representation, capturing its semantic meaning within a high-dimensional space.
- Vector Database with Optimized Indexing: The generated embedding vectors are stored within a vector database. Subsequently, indexing strategies are employed to organize and efficiently search this database based on query vectors.

The effectiveness of context retrieval hinges on the organization within the vector database. The quality of retrieved information, and ultimately the model's answers, depends on how data is structured and partitioned within the database. Inspired by the RAG approach (Lewis et al., 2020), our work employs a chunked document representation. Each document is segmented into smaller chunks, and embedding models generate corresponding vectors. These vectors are then indexed within the database, resulting in a collection of approximately 37 million vectors. When a user submits a question as a query, the embedding model generates a query vector. This vector is then compared against indexed vectors, and the most relevant chunks (based on vector similarity) are retrieved. This retrieved set of chunks provides the context necessary for the model to answer the question effectively.

In our approach, once the system retrieves relevant context, it is incorporated into the LLM prompts discussed earlier in Section 4.2. Additionally, a prompt reminding the LLM of available context is included. This reminder, formatted as "As a potential aid to your answer, background context from Wikipedia articles is at your disposal, delimited by ####, even if it might not always be relevant" encourages the LLM to utilize the retrieved context, while acknowledging the possibility of its limited relevance.

4.3.1. LTL prompt template with context retrieval enhancement

LTL Prompt with context retrieval enhancement

Your task is to analyze the question and answer below. If the answer is correct, respond yes, if it is not correct respond no. As a potential aid to your answer, background context from Wikipedia articles is at your disposal, delimited by #### , even if they might not always be relevant.

####
Context: {context}
####

Question: …
Options: …
A. …
B. …
C. …
D. …
Answer: *LLM_completes_here*

An example of LTL Prompt with context retrieval enhancement

Your task is to analyze the question and answer below. If the answer is correct, respond yes, if it is not correct respond no. As a potential aid to your answer, background context from Wikipedia articles is at your disposal, delimited by #### , even if they might not always be relevant.

####
Context: {context}
####

Question: The primary cause of continental drift, earthquakes, and volcanic eruptions is
A. convection currents beneath Earth's crust.
B. the rotation of Earth on its axis.

C. the gradual sinking of Earth's crust.
D. heat from the Sun warming Earth.
Answer:

4.3.2. CoT prompt template with context retrieval enhancement

CoT Prompt with context retrieval enhancement
Your task is to analyze the question and answer options A, B, C, or D below. Rules: - First, write your reasoning step by step. - Do NOT state the final answer inside the reasoning. - After the reasoning is fully complete, on a new line, write ONLY the final answer in the exact format: Answer: [A/B/C/D] - Do not add any words, explanations, or text after the final answer line. — As a potential aid to your answer, background context from Wikipedia articles is at your disposal, delimited by #### , even if they might not always be relevant.
####
Context: {context}
####
Now answer this question:
Question: ...
A. ...
B. ...
C. ...
D. ...

An example of CoT Prompt with context retrieval enhancement
Your task is to analyze the question and answer options A, B, C, or D below. Rules: - First, write your reasoning step by step. - Do NOT state the final answer inside the reasoning. - After the reasoning is fully complete, on a new line, write ONLY the final answer in the exact format: Answer: [A/B/C/D] - Do not add any words, explanations, or text after the final answer line. — As a potential aid to your answer, background context from Wikipedia articles is at your disposal, delimited by #### , even if they might not always be relevant.
####
Context: {context}
####
Now answer this question:
Question: The primary cause of continental drift, earthquakes, and volcanic eruptions is
A. convection currents beneath Earth's crust.
B. the rotation of Earth on its axis.
C. the gradual sinking of Earth's crust.
D. heat from the Sun warming Earth.

4.3.3. STL prompt template with context retrieval enhancement

STL Prompt with context retrieval enhancement
Your task is to analyze the question and answer below. If the answer is correct, respond yes, if it is not correct respond no. As a potential aid to your answer, background context from Wikipedia articles is at your disposal, delimited by #### , even if they might not always be relevant.
####
Context: {context}
####
Question: ...
Proposed answer: ...
Response: *LLM_completes_here*

An example of STL Prompt with context retrieval enhancement
Your task is to analyze the question and answer below. If the answer is correct, respond yes, if it is not correct respond no. As a potential aid to your answer, background context from Wikipedia articles is at your disposal, delimited by #### , even if they might not always be relevant.
####
Context: {context}
####
Question: Which environmental risk is not associated with the disposal of used motor oil from cars?
Proposed answer: It leaks onto the ground and contaminates soil and ground water.
Response:

4.3.4. Knowledge facts generation

In addition, this work introduces a modification to the traditional RAG query approach (Lewis et al., 2020). Instead of directly using the question from the dataset, the question is first processed by an LLM. This LLM generates factual knowledge potentially relevant to answering the question (Liu et al., 2022). It is important to acknowledge that these generated facts might be susceptible to *hallucination*, as described in (Huang et al., 2023). However, they serve as queries for the RAG system. Since these knowledge facts are likely to contain relevant terms and phrases, the similarity search within the vector database becomes more effective.

Overall, the prompt template includes several key requirements:

- **Fact Generation**: where the generated facts must contain relevant information that helps the user answer the question;
- **Relevance**: where the model must ensure that the content provided is appropriate and related to the question's topic; and
- **Multiple Facts (Optional)**: where the model can provide multiple pieces of knowledge to comprehensively address the user's question if necessary.

Below is a one-shot prompt used to instruct this LLM to generate knowledge from an MCQ:

Prompt
System
You are a resourceful information assistant dedicated to generating knowledgeable facts for the RAG system. Utilize your understanding of Wikipedia concepts and the provided user prompt (usually a multiple-choice question) to craft informative text chunks. Remember, your goal is to provide relevant information from the RAG system to assist users in finding answers.
user
Your task is to generate knowledge facts that help answer user questions, which are presented as multiple-choice questions.
Key considerations:

- Fact Generation: Create text containing relevant information that can potentially answer the user question.
- Relevance: Ensure that the generated facts are pertinent to the topic and context of the user question.
- Multiple Facts (Optional): If necessary, provide multiple facts to cover various aspects of the user prompt or related topics.

Example User Question:
Question: An astronomer observes that a planet rotates faster after a meteorite impact. What is the most likely effect of this increase in rotation?
A. Planetary density will decrease.
B. Planetary years will become longer.
C. Planetary days will become shorter.
D. Planetary gravity will become stronger.
Assistant Response:
Faster rotation of a planet following a meteorite impact can lead to shorter planetary days.
Increased rotation speed after a meteorite impact might result in changes to planetary gravity.
Planetary rotation acceleration due to a meteorite impact could influence various planetary characteristics, including its rotational period.
User Question:
Question: *question*
A. *Option A*
B. *Option B*
C. *Option C*
D. *Option D*
Assistant Response:
assistant

5. Experimental setup

5.1. Models

Selecting an appropriate LLM is critical for effective experimentation, performance evaluation, and baseline method comparisons. The chosen model should exhibit strong performance on the ARC dataset while remaining computationally feasible within the research timeframe. This study opted to utilize two models: Mistral 7B (Jiang et al., 2023) and LLaMa2 7B (Touvron et al., 2023):

- LLaMa2 7B: This well-established model, released in 2023, was a frontrunner in LLM advancements at the time, boasting the top-rated inference capabilities upon its introduction.
- Mistral 7B: Despite being smaller than LLaMa2 13B, Mistral 7B demonstrates exceptional performance with superior inference capabilities. This recently introduced model has been evaluated on a range of datasets, notably ARC Easy and ARC Challenge (Clark et al., 2018).

It is worth noting that LLaMA2 7B and Mistral 7B are among the most popular open-source LLMs, offering the advantage of efficient and effective fine-tuning. Additionally, these fine-tuned models can be tested with our prompting techniques without incurring significant costs.

5.2. MCQ datasets

We evaluate our method on three prominent multiple-choice science question- answering benchmarks: the AI2 Reasoning Challenge (ARC), Scientific Question Answering (SciQ), and OpenBookQA.

- The ARC dataset (Clark et al., 2018) serves as a prominent benchmark for evaluating the performance of LLMs in handling MCQ formats. This publicly accessible dataset, available on the Hugging Face platform (Allenai, 2018a), comprises a total of 7787 natural science questions, divided into two categories: ARC-Easy (5197 questions) and ARC-Challenge (2590 questions). ARC-Easy questions are relatively easier and can be solved using simple reasoning, while ARC-Challenge questions require deeper reasoning and abstract thinking. The questions in the ARC dataset vary in their target student grade level, ranging from 3rd to 9th grade (ages 8 to 13). However, it is important to note that there is substantial overlap in difficulty among grade levels, as each grade level contains a mixture of easy and difficult questions.
- **SciQ** (Welbl et al., 2017): A collection of 13,679 multiple-choice science exam questions covering Physics, Chemistry, and Biology, which is also available on the Hugging Face platform (Allenai, 2017).
- **OpenBookQA** (Mihaylov et al., 2018b): Comprising 5957 elementary-level science questions, this dataset is modeled after open-book exams and requires multi-step reasoning, commonsense knowledge, and rich text comprehension. The dataset is also public on the Hugging Face platform (Allenai, 2018b).

Table 1 summarizes the sizes of the train, development (dev), and test partitions across all three datasets. These partitions are used for training, validating, and testing the model, respectively. For the ARC dataset, the question vocabulary consists of 6329 distinct words (stemmed) (Clark et al., 2018).

For reference, Table 2 presents a table extracted from the original Mistral 7B research paper (Jiang et al., 2023), showcasing the performance of both Mistral 7B and LLaMa2 7B on several prominent multiple-choice datasets. Based on the Table 2, the key baseline measures can be summarized as shown in Table 3.

5.3. Context retrieval

Indexing: To facilitate the context retrieval process, Facebook AI Similarity Search (FAISS) (Johnson et al., 2019) is employed for vector organization and indexing. This technique enables efficient retrieval of the top-k most similar vectors to a query vector, based on Euclidean distance. FAISS is particularly adept at handling large-scale educational materials due to its use of parallel computation, hash-based search, and k-nearest neighbor algorithms.

Table 1
Number of samples in the ARC Easy and ARC Challenge in the ARC, SciQ, and OpenBookQA datasets.

Dataset	Train	Dev	Test
ARC-Easy	2251	570	2376
ARC-Challenge	1119	299	1172
ARC (Summary)	3370	869	3548
SciQ	11,679	1000	1000
OpenBookQA	4957	500	500

Table 2
Results of Mistral 7B and variants of LLaMa2 on ARC datasets (Jiang et al., 2023).

Model	Modality	ARC easy	ARC challenge
LLaMa 2 7B	Pretrained	68.7%	43.2%
LLaMa 2 13B	Pretrained	75.2%	48.8%
Code-LLama 7B	Finetuned	59.4%	34.5%
Mistral 7B	Pretrained	80.0%	55.5%

Table 3
Reported results of Mistral 7B and LLaMa2 7B from (Jiang et al., 2023) and (Touvron et al., 2023) on ARC, SciQ, and OpenBookQA datasets.

Dataset	LLaMa2 7B	Mistral 7B
ARC Easy	68.7	80.0
ARC Challenge	43.2	55.5
ARC (Summary)	60.27	71.9
SciQ	N/A	N/A
OpenBookQA	41.0	60.5

Documents: A large and diverse document source is used, with the primary source being the English version of the Wikipedia dataset (Foundation, 2024), which comprises approximately 6.5 million articles.

Text Embedding Models: Text embedding is crucial for enhancing information retrieval through vector databases that store the embeddings generated by the model. We opted for the pre-trained *bge-small-en* model (Xiao et al., 2024), which offers a favorable balance between embedding dimension (384) and computational resource requirements, ensuring efficient organization of embeddings within the vector database.

Generating Knowledge Facts: To generate knowledge facts for use as query contexts with the language model, we selected the quantized version of the Mistral 7B model, fine-tuned with the OpenOrca dataset (TheBloke, 2023) specifically designed for the question-answering tasks. This choice expedites processing and minimizes computational resource consumption.

5.4. Results

This section details the two LLM-based approaches employed for MCQs. The first approach is the baseline method, Labels Token Logits (LTL), derived from the Hendrycks' framework (Hendrycks, 2020). The second approach is the Chain-of-Thought (CoT) prompting approach. Our proposed approach is the Single Token Logits (STL) method. Three options are considered for providing additional context to the LLMs: no context, context from the standard RAG model, and context from a RAG model enhanced with our group's query improvement techniques. This combination of two LLM approaches with three context options yields a total of six distinct experimental conditions for each dataset. To assess performance under different information availability conditions, we evaluate these methods across multiple science-oriented question-answering datasets, both with and without retrieval-augmented context.

5.4.1. Main results

Table 4 presents a comprehensive comparison of the three methods (LTL, CoT, and STL) across four science-oriented question-answering datasets: ARC-Challenge (ARC-C), ARC-Easy (ARC-E), OpenBookQA, and SciQ. These results represent core performance relying solely on the models' abilities.

The results presented in Table 4 offer a compelling comparison of the LTL, CoT, and STL methods, revealing distinct performance patterns. For the LLaMa2 7B model, the proposed STL method demonstrates clear superiority across all datasets. It consistently establishes a performance hierarchy of **STL** > **CoT** > **LTL**. On the ARC-Challenge dataset, for instance, STL achieves an accuracy of 58.11%, surpassing CoT by over 5.5 percentage points and the baseline LTL by more than 11 points. This strong and consistent outperformance suggests that for models in this performance tier, the STL method, which mitigates the MCSB problem through independent verifications, is substantially more effective than both standard multi-option prompting (LTL) and generative reasoning (CoT).

With the Mistral 7B model, the performance gap between the methods narrows, and the hierarchy becomes **STL** ≈ **CoT** > **LTL**. CoT shows a slight increase in performance, achieving the highest score on ARC-Challenge (67.83%), a dataset known for its complex reasoning questions, while STL proves most effective on the ARC-Easy dataset (84.30%).

A phenomenon is observed on the OpenBookQA dataset, where the baseline LTL method is the top performer (69.04%). This can be attributed to the characteristics of the OpenBookQA dataset, which focuses on factual questions where the answer options are typically very short (often 1–2 words) and distinct. In such a context, a powerful model like Mistral 7B might benefit more from the directness of the LTL method over the processes of STL or the more complex reasoning of CoT. However, a granular analysis of how performance varies by specific dataset characteristics or question categories is beyond the scope of this paper and remains a priority for future investigation.

Table 4
Core comparison of methods across models and datasets.

Model	Method	ARC-C	ARC-E	OpenBookQA	SciQ
LLaMa2 7B	LTL	46.76	65.15	50.14	71.50
	CoT	52.47	71.13	50.82	78.20
	STL	**58.11**	**77.02**	**60.55**	**81.10**
Mistral 7B	LTL	63.99	81.19	**69.04**	81.60
	CoT	**67.83**	82.45	67.95	**84.80**
	STL	64.68	**84.30**	67.95	84.60

5.4.2. Evaluation with external context

To understand how different methods perform when supplemented with external knowledge, we conducted additional experiments using the ARC dataset with three context conditions: no context (Empty), standard RAG retrieval, and our enhanced query RAG with the knowledge facts generation technique described in Section 4.3.

The following Tables 5 and 6 present the experimental results for both the ARC-Easy and ARC-Challenge datasets, categorized by LLM (LLaMa2 and Mistral). The first row in each table displays the performance of the LTL baseline method on MCQs without additional context, as reported in the original Mistral 7B paper. The "ARC (Summary)" column aggregates the accuracy scores from both the ARC-Easy and ARC-Challenge datasets to provide an overall performance metric.

For research purposes and to facilitate integration with our retrieval system's additional context, a reimplementation of the LTL baseline was undertaken, distinct from the original framework. This reimplementation, which included the creation of new prompts, resulted in slight deviations in individual dataset performance compared to the original paper's data. However, the overall accuracy across the entire ARC dataset remains comparable. The "ARC (Summary)" column, representing aggregated performance across both datasets, serves as a basis for comparing the effectiveness of different LLM approaches under varying context scenarios.

Table 5
Experimental results on the LLaMa2 7B dataset.

LLaMa2 7B	Context	ARC easy	ARC challenge	ARC (summary)
Paper	Empty	68.7	43.2	60.27
LTL		65.15	46.76	59.07
CoT		71.13	52.47	64.96
STL		77.02	58.11	70.77
LTL	RAG	63.22	49.32	58.62
CoT		66.54	50.17	61.13
STL		76.47	58.28	70.46
LTL	Enhance query RAG	72.05	54.10	66.12
CoT		71.89	52.73	65.56
STL		83.80	62.63	76.80

Table 6
Experimental results on the Mistral 7B dataset.

Mistral 7B	Context	ARC easy	ARC challenge	ARC (Summary)
Paper	Empty	80.0	55.5	71.9
LTL		75.55	63.99	71.7
CoT		82.45	67.83	77.62
STL		84.30	64.68	77.81
LTL	RAG	81.19	66.89	76.46
CoT		82.41	66.98	77.31
STL		82.62	65.19	76.86
LTL	Enhance query RAG	84.55	70.05	79.76
CoT		85.73	70.82	80.8
STL		87.29	68.43	81.06

5.4.3. Analysis

Across all experimental conditions, the proposed STL method generally outperforms the baseline LTL approach while remaining competitive with—or slightly outperforming—the established CoT prompting strategy, as evidenced in Tables 5 and 6. For example, on the LLaMa2 7B model with no context, STL achieves 70.77% accuracy on the combined ARC dataset, surpassing LTL (59.07%) and CoT (64.96%). On Mistral 7B under the same setting, STL attains 77.81%, exceeding LTL (71.7%) and matching CoT (77.62%). These gains highlight STL's efficacy in addressing MCSB limitations through independent semantic evaluation of answer choices, free from positional biases.

To further evaluate robustness, we assess performance under varying levels of context richness: no context (baseline intrinsic reasoning), standard RAG (potentially noisy retrieval), and enhanced query RAG (refined, high-value information via knowledge fact generation). This framework simulates real-world applications where external knowledge may range from imperfect to optimized. Notably, STL not only maintains its relative advantages across these conditions but also amplifies them in richer contexts, underscoring its practical utility in knowledge-augmented scenarios.

It is essential to reiterate that the results from the original paper pertain solely to the LTL baseline under no context and do not incorporate additional retrieval. To maintain focus on the intrinsic capabilities of each prompting method within the scope of this research, few-shot techniques were deliberately omitted across all methods. These findings are further corroborated by statistical significance testing in Section 5.4.5.

5.4.4. Cost and latency analysis

While the STL approach demonstrates superior accuracy compared to LTL, it is important to evaluate the computational overhead associated with this improvement. The STL method requires processing k distinct prompts for a single sample, which are currently evaluated as a single batch of size k. In this study, we conducted comprehensive experiments across multiple models and datasets to quantify the resource requirements of STL relative to LTL. This comparison focuses on STL versus LTL as both operate in the same computational paradigm (single-pass logit extraction). CoT is excluded because its autoregressive multi-token generation places it in a fundamentally different and substantially higher cost regime, which would obscure the STL-vs-LTL differences at this scale.

Our analysis reveals the following computational characteristics (illustrated in Figs. 10–13):

- **Latency:** STL exhibits $1.58 \pm 0.09\times$ slower execution compared to LTL. This relatively modest increase is attributed to parallel evaluation of prompts within each batch, which mitigates the overhead of processing multiple prompts per question (illustrated in Fig. 10).
- **GPU Memory:** Memory utilization increases by $3.72 \pm 0.34\times$, which is the most significant resource constraint when deploying STL in production environments (illustrated in Fig. 11).

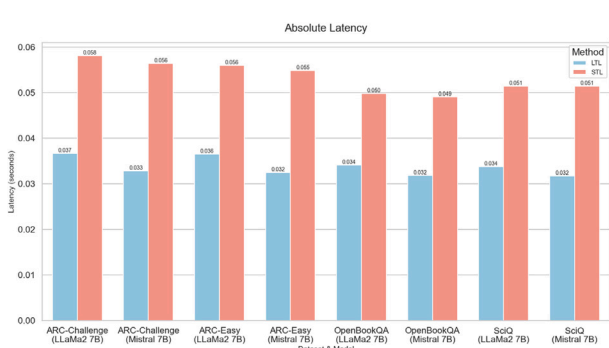

Fig. 10. Absolute latency (seconds per question) for LTL and STL across all model–dataset combinations.

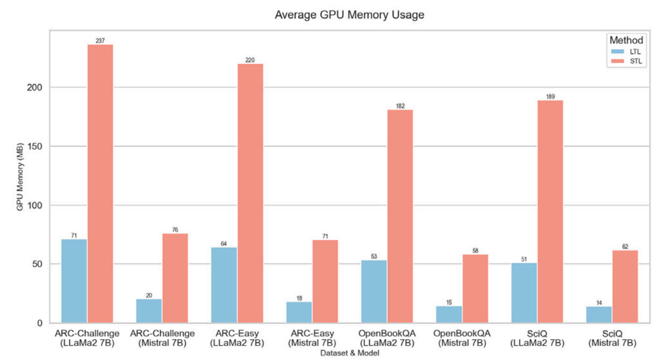

Fig. 11. Average GPU memory usage (MB) for LTL and STL.

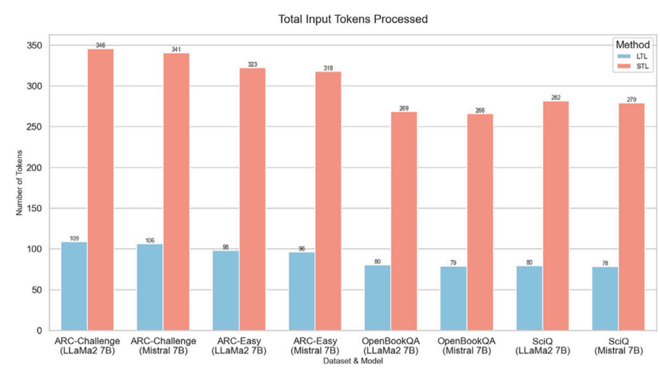

Fig. 12. Total input tokens processed per question for LTL and STL.

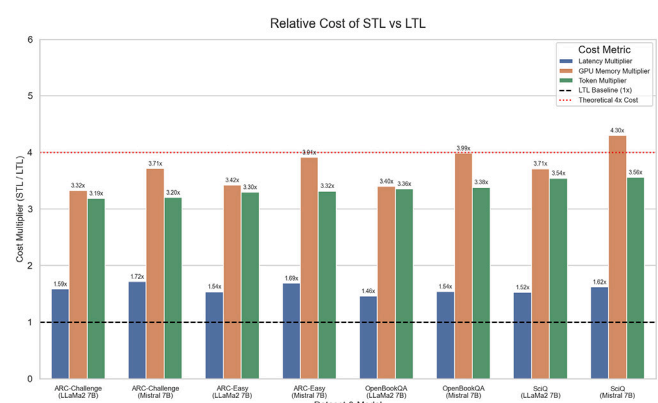

Fig. 13. Relative cost multipliers of STL over LTL across latency, GPU memory, and token consumption.

- **Token Consumption:** The method requires $3.35 \pm 0.14\times$ more input tokens per question, reflecting the need to generate and process k distinct reasoning paths (illustrated in Fig. 12).

It is important to note that these benchmarks were conducted using a standard implementation within the Hugging Face Transformers library. The exact performance multipliers may differ on other inference frameworks, particularly those with advanced optimizations such as flash attention or same-prefix caching, which could potentially reduce both memory footprint and latency.

Despite the increased computational requirements, the accuracy gains achieved by STL may justify the additional resource consumption in applications where prediction quality is paramount. However, practitioners should carefully consider the trade-off between accuracy improvements and resource constraints when selecting between LTL and STL approaches for their specific use cases.

5.4.5. Statistical significance testing

To validate the observed performance improvements, we conducted statistical significance testing using McNemar's Test. This test is specifically designed for paired, sample-by-sample analysis, examining the specific outcomes for each individual question to determine if there is a statistically significant directional change in correctness between two methods. Instead of merely comparing aggregate accuracy scores, this approach directly confirms whether the observed performance gains represent consistent improvements rather than random variation.

Our methodology operates on paired comparisons at the sample level. For each question in the test set, we compare the predictions of two methods (e.g., STL and LTL) and categorize the outcome into one of four cases: (1) both methods correct, (2) both methods incorrect, (3) STL correct while the compared method is incorrect (recorded as "STL wins"), or (4) the compared method correct while STL is incorrect (recorded as "LTL wins" or "CoT wins"). McNemar's Test then evaluates whether the discordant pairs—cases where one method succeeds and the other fails—show a statistically significant asymmetry, indicating genuine performance differences rather than random variation.

We performed two sets of comparisons: (1) STL versus the baseline LTL approach, and (2) STL versus the Chain-of-Thought (CoT) prompting strategy. All tests were conducted at the $\alpha = 0.05$ significance level across four benchmark datasets (ARC-Challenge, ARC-Easy, OpenBookQA, and SciQ) using both Mistral 7B and LLaMa2 7B models. Based on the p-values obtained from McNemar's Test, we classify the results into three categories: "STL is Better" ($p < 0.05$ with STL wins > comparison method wins), "Equivalent" ($p \geq 0.05$), or "LTL/CoT is Better" ($p < 0.05$ with comparison method wins > STL wins).

The statistical analysis reveals two key findings:

- **STL is Statistically Superior to LTL:** The results confirm our primary hypothesis, with STL demonstrating statistically significant improvements ($p < 0.05$) over the LTL baseline in most conditions (see Table 7). The remaining two conditions show equivalent performance, with no cases where LTL significantly outperforms STL.
- **STL is Highly Competitive with CoT:** When compared against the established Chain-of-Thought prompting strategy, STL proves to be a powerful alternative. It achieved statistically significant improvements in 4 of the 8 conditions while performing equivalently in the other 4 (see Table 8). Crucially, STL was not outperformed by CoT, establishing it as a highly reliable method. This makes STL a particularly compelling choice, as it delivers comparable accuracy, whereas CoT's approach is inherently more costly due to its requirement for generating long, token-intensive reasoning chains.

In summary, this rigorous statistical analysis provides strong evidence to support our paper's central claims: STL represents a statistically significant improvement over the standard LTL baseline and stands as a competitive alternative to other advanced prompting techniques like CoT.

Table 7
McNemar's test results: STL vs. LTL.

Model	Dataset	STL wins	LTL wins	p-value	Result
Mistral 7B	ARC-Challenge	161	154	0.7353	Equivalent
Mistral 7B	ARC-Easy	221	145	0.0001	STL is Better
Mistral 7B	OpenBookQA	86	94	0.6018	Equivalent
Mistral 7B	SciQ	84	54	0.0136	STL is Better
LLaMa2 7B	ARC-Challenge	259	143	0.0000	STL is Better
LLaMa2 7B	ARC-Easy	479	205	0.0000	STL is Better
LLaMa2 7B	OpenBookQA	166	90	0.0000	STL is Better
LLaMa2 7B	SciQ	161	65	0.0000	STL is Better

Table 8
McNemar's test results: STL vs. CoT.

Model	Dataset	STL wins	CoT wins	p-value	Result
Mistral 7B	ARC-Challenge	173	210	0.0658	Equivalent
Mistral 7B	ARC-Easy	234	190	0.0368	STL is Better
Mistral 7B	OpenBookQA	114	114	0.9472	Equivalent
Mistral 7B	SciQ	84	86	0.9389	Equivalent
LLaMa2 7B	ARC-Challenge	253	187	0.0019	STL is Better
LLaMa2 7B	ARC-Easy	407	267	0.0000	STL is Better
LLaMa2 7B	OpenBookQA	178	107	0.0000	STL is Better
LLaMa2 7B	SciQ	124	95	0.0585	Equivalent

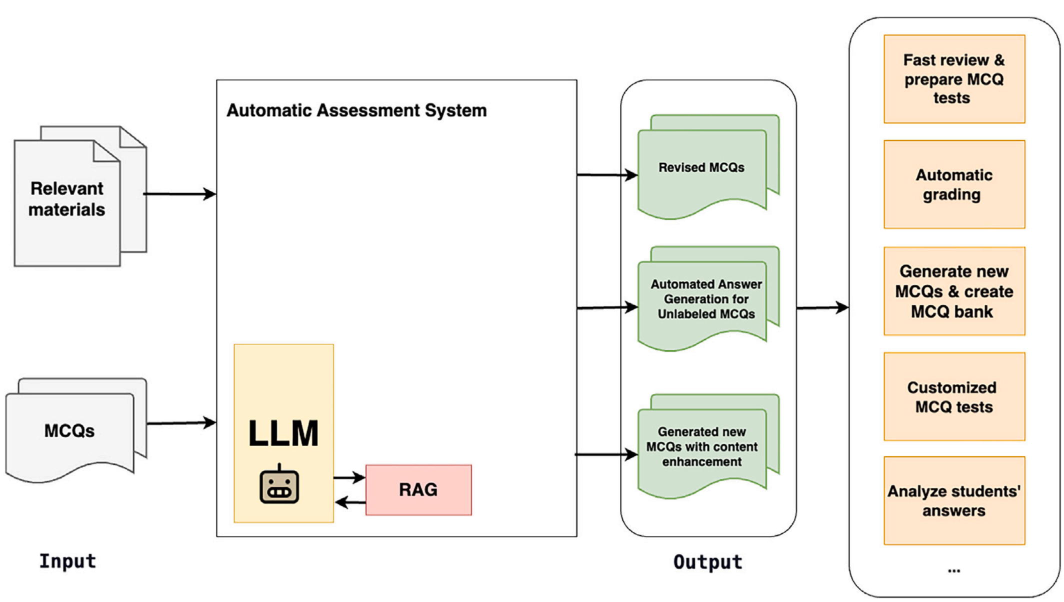

Fig. 14. Automatic assessment system powered by our LLM-based enhancements for MCQA.

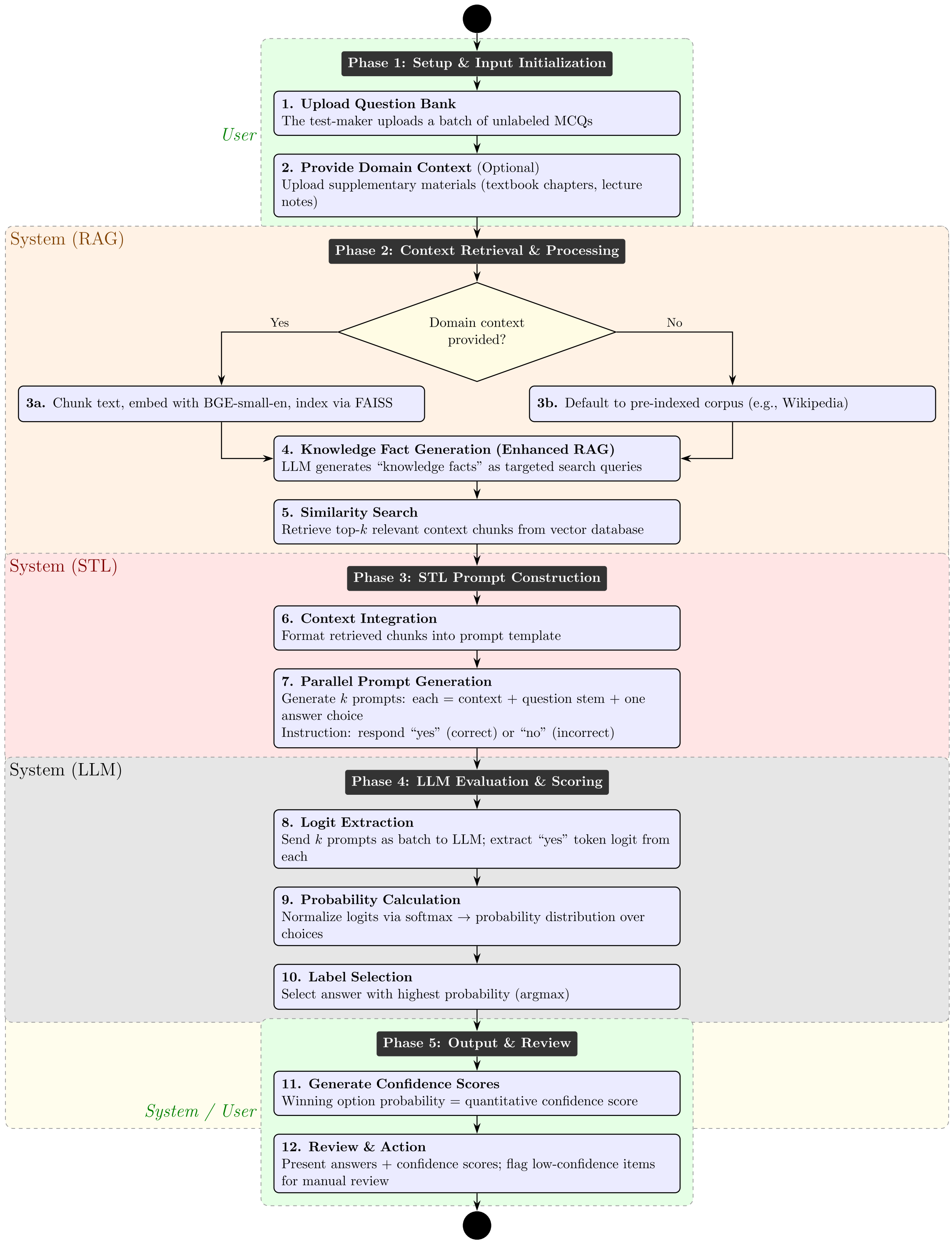

Fig. 15. UML activity diagram of the Automated answer Generation workflow, illustrating the end-to-end data flow from user input through RAG context retrieval, STL prompt construction, LLM evaluation, to final output and review.

6. Application scenario

Potential Application: Automated Answer Generation for Unlabeled MCQs.
Context: In educational settings, test-makers often handle large volumes of MCQs that require manual grading. Many items also lack pre-labeled answers, further complicating quality assurance.
Scenario: A test-maker is preparing a new set of assessments and has developed a list of MCQs. However, these questions do not have labeled answers, and the test-maker wants to ensure they are accurate and aligned with the provided course materials.
Application: *The automated assessment system (as illustrated in Fig. 14).*

Setup: The test-maker uploads a list of MCQs to the automated assessment system. Alongside each question, the test-maker may attach relevant educational materials (e.g., textbook chapters, lecture notes). Alternatively, when no materials are supplied, the system's retrieval-augmented generation (RAG) component can automatically find relevant context to help answer the questions.

System process: The platform applies the proposed methods to the questions and their associated context. The LLM generates answers by leveraging single-token logit scoring together with contextual information retrieved by the improved RAG module.

Output: The system provides a suggested answer for each question, along with a confidence score to aid review. It can also assist in revising MCQs to identify and correct overlooked errors. Furthermore, upon request, the system can generate new MCQs with enhanced content.

Detailed implementation workflow

To illustrate the practical deployment of our Automatic Assessment System, we outline the step-by-step activity workflow for the Automated Answer Generation for Unlabeled MCQs scenario. This workflow bridges the user interface with our underlying, empirically validated STL and Retrieval-Augmented Generation (RAG) architectures:

Phase 1: Setup and Input Initialization (User Action)

1. **Upload Question Bank:** The test-maker uploads a batch of unlabeled Multiple-Choice Questions (MCQs) into the system.
2. **Provide Domain Context (Optional):** The test-maker uploads relevant supplementary educational materials (e.g., textbook chapters, lecture notes) to refine the system's knowledge base.

Phase 2: Context Retrieval & Processing (System Action)

3. **Document Indexing:** If domain context is provided, the system chunks the text, applies text embedding models (e.g., BGE-small-en), and indexes the vectors using FAISS within a vector database. If no materials are provided, the system defaults to a pre-indexed corpus, such as Wikipedia.
4. **Knowledge Fact Generation (Enhanced RAG):** For each MCQ, the system queries the LLM to generate "knowledge facts" based on the question, which serve as highly relevant, targeted search queries.
5. **Similarity Search:** The system uses these generated knowledge facts to retrieve the top-k most relevant context chunks from the vector database.

Phase 3: STL Prompt Construction (System Action)

6. **Context Integration:** The retrieved context chunks are formatted into the prompt template.
7. **Parallel Prompt Generation:** For a question with k answer choices, the system generates k separate prompts. Each prompt combines the retrieved context, the question stem, and exactly one of the proposed answer choices, instructing the model to respond "yes" if the answer is correct and "no" if it is not.

Phase 4: LLM Evaluation and Scoring (System Action)

8. **Logit Extraction:** The k prompts are sent to the LLM (e.g., Mistral 7B or LLaMA2 7B) as a batch. The system specifically extracts the raw logit value of the "yes" token from the LLM's first output position for each of the k prompts.
9. **Probability Calculation:** The system normalizes the "yes" logits and applies a softmax function to calculate a comparative probability distribution across all answer choices.
10. **Label Selection:** The system selects the answer label with the highest probability using an argmax function.

Phase 5: Output and Review (System / User Action)

11. **Generate Confidence Scores:** The calculated probability of the winning option serves as a quantitative confidence score.
12. **Review & Action:** The system presents the suggested answers and confidence scores to the test-maker. Questions with low confidence scores are automatically flagged for the test-maker to manually review or may trigger the system to assist in revising the MCQ for clarity.

By detailing the exact data handoffs between the user, the RAG vector database, and the STL logit-extraction module, this workflow demonstrates that the application is fully realizable using the exact algorithms validated in Section 5.

Fig. 15 presents the corresponding UML activity diagram, which visualizes the end-to-end data flow across these five phases.

Benefits:

- **Rapid review & test preparation:** Test-makers can review suggested answers and make necessary adjustments. The accuracy of the proposed methods reduces the need for extensive manual checking, saving time and effort when working with large item sets.
- **Automated grading:** The system can generate answers for unlabeled MCQs, extending applicability across a wide range of assessment scenarios.
- **MCQ generation & item bank creation:** With RAG-based content enhancement, the system can compose new MCQs and help maintain an up-to-date MCQ bank.
- **Customized MCQ tests:** By analyzing student responses, the system can help educators tailor MCQs to specific needs and learning profiles, enabling more personalized instruction.

Impact: By implementing this application, educational institutions can leverage LLM-based automation to generate answers for MCQs. This automation saves time for test-makers and helps ensure consistency with course materials. As a result, test-makers can devote more effort to interactive teaching and personalized student support, ultimately improving the overall learning experience.

7. Conclusion

Large Language Models (LLMs) demonstrate significant potential in the field of education, particularly in enhancing intelligent assessment systems. Although LLMs have shown some limitations in understanding MCQA tasks, our research presents notable advancements in this area. The conducted experiments convincingly demonstrate that our proposed Single-Token Logit (STL) methodology outperforms the standard baseline and demonstrates highly competitive performance against strong alternatives like Chain-of-Thought (CoT). Key takeaways from the experimental evaluation of these methods include:

- **Single-Token Logit Prompting:** Our proposed Single-Token Logit prompting technique effectively addresses the limitations of MCSB, improving LLM performance in MCQA tasks.
- **Robustness in Rich-Context Scenarios:** We demonstrate that STL's performance gains are robust across different informational conditions. The method maintains its superiority over the baseline even when provided with rich external context via Retrieval-Augmented Generation (RAG), highlighting its practical utility in real-world, knowledge-intensive applications.
- **Competitive Performance Against Chain-of-Thought:** Through rigorous, sample-by-sample statistical testing, we confirm that STL is not only superior to the standard baseline but also highly competitive with the strong Chain-of-Thought (CoT) method. This positions STL as a potent alternative, especially considering that CoT often incurs significant computational and latency costs due to its verbose reasoning generation. Moreover, a key advantage of STL lies in its fully automatic integration within systems: it leverages direct logit-based evaluation for prediction. In contrast, CoT relies on human-designed prompting structures and format-based evaluation, which can introduce variability and additional engineering

overhead. STL's streamlined design is significantly more cost-efficient than CoT—requiring no multi-token generation—while incurring only a moderate overhead relative to the single-pass LTL baseline (see Section 5.4.4). This balance makes it particularly suitable for scalable, automated deployment in production environments where both accuracy and computational budget are important considerations.

Our findings underscore the potential of LLMs to revolutionize educational assessment by providing accurate and efficient tools for generating and grading MCQs. While this study has yielded promising results, it is crucial to scrutinize the applicability of these methods to a wider range of subjects that require more logical reasoning and natural language production, or tasks demanding high reasoning abilities, such as complex mathematics, particularly given that all current benchmarks are drawn from science-oriented datasets. Moreover, STL's accuracy gains come at a measurable computational cost over the single-pass LTL baseline (1.58× latency, 3.72× GPU memory, as detailed in Section 5.4.4), a trade-off that practitioners should weigh against the accuracy improvements for their specific use cases (see Section 8 for a full discussion of these limitations).

8. Future work and limitations

While our study demonstrates the effectiveness of the STL method, we acknowledge several limitations that present clear avenues for future research.

8.1. Limitations

- **Increased Computational Cost Over Baseline (LTL):** The STL method, by its design, requires a separate forward pass through the model for each of the k answer choices. This results in a computational cost approximately k times greater than the single-pass LTL baseline. While this trade-off yields significant accuracy improvements, it must be considered for applications with computational budgets. It is important to note, however, that this cost remains substantially lower than that of verbose reasoning methods like Chain-of-Thought.
- **Threats to Validity:** Even though our experimental results demonstrate the superiority of STL across multiple datasets, a potential threat to validity remains that our method could be inadvertently tuned to the specific style of academic science questions. Furthermore, while in our experiments, the base models may carry inherent biases or have been exposed to these datasets during pretraining, a factor that affects all evaluated methods but warrants consideration.

8.2. Future work

- **Granular Error Analysis:** To gain a deeper understanding of our method's strengths and weaknesses, future works can perform a detailed error analysis. This will involve categorizing questions by type (e.g., factual recall, multi-step reasoning, commonsense) to identify specific areas where STL excels or fails, guiding further refinements.
- **Exploration of Few-Shot Prompting:** Building on our zero-shot baseline, future works should investigate the impact of incorporating few-shot examples. This will help determine how STL's performance scales with in-context learning and how it compares to methods like CoT in a few-shot regime. More broadly, evaluating STL's interaction with adaptive prompting strategies (e.g., dynamic exemplar selection) and post-hoc calibration techniques would clarify whether these complementary approaches can further improve the confidence estimates that STL natively produces.
- **Broader Domain and Task Generalization:** To further validate the robustness of our findings, further evaluation should be more conducted on diverse and challenging benchmarks, including those from non-science domains (e.g., humanities, law) and tasks requiring higher-order reasoning. This direction is critical for addressing the domain-bias threat to validity identified in this study, as all current benchmarks are drawn from science-oriented question sets.
- **Prompt Sensitivity Analysis:** A systematic study of how variations in the STL verification prompt (e.g., alternative phrasing of the yes/no question, different target tokens) affect performance would provide important evidence of the method's robustness and help establish best practices for prompt design across different models and domains.
- **STL-Enabled Voting for Robust Decision-Making:** A promising future direction is to generalize the STL principle for use in a voting ensemble. In this setup, a single prompt input designed for verification is sent to diverse models simultaneously—for example, a lightweight model for speed and a specialized one for domain expertise. Each model performs just a single forward pass on the input to generate a logit score for a target token (e.g., "yes"), which quantifies its confidence in the validity or quality of the provided information. A central evaluator then aggregates these individual confidence logits from across the ensemble. This approach enables far more sophisticated and reliable decision-making than relying on a single model's judgment, allowing for weighted voting or confidence-based filtering. This logit-driven fusion is highly efficient and could be used to create robust applications, from educational tutors that "vote" on the quality of a generated explanation to uncertainty-aware systems for high-stakes domains like medical diagnostics or legal reasoning.

CRediT authorship contribution statement

Quoc Phu Dang: Methodology, Investigation, Conceptualization. **Phat T. Tran-Truong:** Writing – original draft, Investigation, Data curation. **Duc-Ly Vu:** Writing – review & editing, Validation, Investigation, Conceptualization. **Long S. T. Nguyen:** Writing – review & editing, Data curation. **Quynh T. N. Vo:** Writing – review & editing, Data curation. **Tho Quan:** Writing – review & editing, Validation, Supervision, Methodology, Investigation, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

We acknowledge Ho Chi Minh City University of Technology (HCMUT), VNU-HCM for supporting this study.

Appendix A. Sample questions with improved predictions

To illustrate the effectiveness of the proposed methods, we present several sample questions that have been reformulated to improve answer retrieval. These examples utilize either the adjusted RAG system or the LLM approach.

First sample

Question: Each body cell of a goldfish contains 94 chromosomes. How many chromosomes are contained in a goldfish sex cell?

A. 23
B. 47
C. 94
D. 188

LTL:

Prediction: D (Wrong)
STL:
Prediction: B (Correct)

Second sample
Question: A potential negative impact of building a dam on a river is that the dam
A. prevents sediment from flowing downstream.
B. increases the amount of water available to farms.
C. prevents seasonal downstream flooding.
D. increases the rate of water loss from a lake.
LTL & RAG:
Prediction: C (Wrong)
LTL & Enhanced Query RAG:
Generated knowledge facts: Building a dam on a river can have several negative impacts. One potential negative effect is that the dam prevents sediment from flowing downstream, which can lead to changes in river ecosystems and sedimentation issues. Additionally, the dam may prevent seasonal downstream flooding, which can disrupt natural habitats and ecosystems.
Prediction: A (Correct)

Third sample
Question: A group of science students used water, filter paper, and a hot plate to separate a mixture of salt and sand. Which of the following did the students most likely do first?
A. shake the mixture until the sand settles at the bottom.
B. pour the mixture through the filter paper.
C. heat the mixture on the hot plate.
D. dissolve the salt by adding water to the mixture.
STL & RAG:
Prediction: D (Correct)
STL & Enhanced Query RAG:
Generated knowledge facts: The students most likely began by dissolving the salt by adding water to the mixture. This step would have allowed them to separate the salt from the sand more easily. Next, the students used the filter paper to separate the remaining mixture, allowing the sand to be retained on the paper while the water passed through. Finally, the students shook the mixture until the sand settled at the bottom, ensuring that the sand was separated from the remaining water. In summary, the students first dissolved the salt by adding water to the mixture, then used filter paper to separate the remaining components, and finally shook the mixture to settle the sand.
Prediction: C (Wrong)

The third example highlights a scenario where the LLM successfully generated text containing the correct explanation. However, the crucial answer itself resided solely within the first sentence of the explanation. Utilizing the entire generated text as the query for the RAG system resulted in the retrieval of a significant amount of irrelevant information. This extraneous data overwhelmed the LLM, ultimately causing the model to provide incorrect answers compared to the standard RAG approach.

References

Allenai. (2017). SciQ dataset. https://huggingface.co/datasets/allenai/sciq [Accessed 2025–October–01].
Allenai. (2018a). ai2-arc dataset. https://huggingface.co/datasets/allenai/ai2_arc [Accessed 2024–June–30].
Allenai. (2018b). OpenBookQA dataset. https://huggingface.co/datasets/allenai/openbookqa [Accessed 2025–October–01].
Alomran, M., & Chai, D. (2018). Automated scoring system for multiple choice test with quick feedback. *International Journal of Information and Education Technology*, *8*, 538–545.
Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, *5*, 135–146.
Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., et al., On the opportunities and risks of foundation models, arXiv preprint arXiv:2108.07258, 2021.
Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, *33*, 1877–1901.
Chen, J., Lin, H., Han, X., & Sun, L. (2024a). Benchmarking large language models in retrieval-augmented generation. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 38, pp. 17754–17762).
Chen, L., Wang, W., Bai, Z., Xu, P., Fang, Y., Fang, J., Wu, W., Zhou, L., Zhang, R., Xia, Y., et al., Pharmgpt: Domain-specific large language models for bio-pharmaceutical and chemistry, arXiv preprint arXiv:2406.18045, 2024b.
Chen, M., Tworek, J., Jun, H., Yuan, Q., de Oliveira Pinto, H. P., Kaplan, J., Edwards, H., Burda, Y., Joseph N., Brockman, G., Ray, A., Puri, R., Krueger, G., Petrov, M., Khlaaf, H., Sastry, G., Mishkin, P., Chan, B., Gray, S., Ryder, N., Pavlov, M., Power, A., Kaiser, L., Bavarian, M., Winter, C., Tillet, P., Such, F. P., Cummings, D., Plappert, M., Chantzis, F., Barnes, E., Herbert-Voss, A., Guss, W. H., Nichol, A., Paino, A., Tezak, N., Tang, J., Babuschkin, I., Balaji, S., Jain, S., Saunders, W., Hesse, C., Carr, A. N., Leike, J., Achiam, J., Misra, V., Morikawa, E., Radford, A., Knight, M., Brundage, M., Murati, M., Mayer, K., Welinder, P., McGrew, B., Amodei, D., McCandlish, S., Sutskever, I., & Zaremba, W., Evaluating large language models trained on code, arXiv preprint arXiv:2107.03374, 2021.
Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (Dec 2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 workshop on deep learning*.
Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., & Tafjord, O., Think you have solved question answering? Try arc, the ai2 reasoning challenge, arXiv preprint arXiv:1803.05457, 2018.
Devlin, J., Chang, M., Lee, K., & Toutanova, K. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In J. Burstein, C. Doran, & T. Solorio (Eds.), *Proceedings of the 2019 conference of the north American chapter of the association for computational linguistics: Human language technologies, NAACL-HLT 2019, June 2–7, 2019, (long and short papers)* Vol. 1, pp. 4171–4186. Minneapolis, MN, USA: Association for Computational Linguistics. https://doi.org/10.18653/v1/n19-1423
Dominguez-Olmedo, R., Hardt, M., & Mendler-Dünner, C. (2024). Questioning the survey responses of large language models. In *ICLR 2024 workshop on reliable and responsible foundation models*. https://openreview.net/forum?id=cvy6DMGnqP.
Foundation, W. (2024). Wikimedia downloads. https://huggingface.co/datasets/wikimedia/wikipedia [Accessed 30-June-2024].
Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.
He, H., Zhang, H., & Roth, D., Rethinking with retrieval: Faithful large language model inference, arXiv preprint arXiv:2301.00303, 2022.
Hendrycks. (2020). Measuring massive multitask language understanding framework. https://github.com/hendrycks/test [Accessed 2024–July–30].
Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., & Steinhardt, J. (2020). Measuring massive multitask language understanding. In *International conference on learning representations*.
Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*, 1735–1780.
Huang, L., Yu, W., Ma, W., Zhong, W., Feng, Z., Wang, H., Chen, Q., Peng, W., Feng, X., Qin, B., et al., A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions, arXiv preprint arXiv:2311.05232, 2023.
Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., de Las Casas, D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L. R., Lachaux, M., Stock, P., Scao, T. L., Lavril, T., Wang, T., Lacroix, T., & Sayed, W. E. arXiv preprint arXiv:2310.06825. Mistral 7b, 2023.
Johnson, J., Douze, M., & Jégou, H. (2019). Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, *7*, 535–547.
Karpukhin, V., Oguz, B., Min, S., Lewis, P. S. H., Wu, L., Edunov, S., Chen, D., & Yih, W. 2020. Dense passage retrieval for open-domain question answering. In B. Webber, T. Cohn, Y. He, & Y. Liu (Eds.), *Proceedings of the 2020 conference on empirical methods in natural language processing, EMNLP 2020, online, November 16–20, 2020* (pp. 6769–6781). Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.emnlp-main.550
Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., & Iwasawa, Y. (2022). Large language models are zero-shot reasoners. *Advances in neural information processing systems*, *35*, 22199–22213.
Lai, G., Xie, Q., Liu, H., Yang, Y., & Hovy, E. H. 2017. RACE: Large-scale reading comprehension dataset from examinations. In M. Palmer, R. Hwa, & S. Riedel (Eds.), *Proceedings of the 2017 conference on empirical methods in natural language processing, EMNLP 2017, September 9–11, 2017* (pp. 785–794). Copenhagen, Denmark: Association for Computational Linguistics. https://doi.org/10.18653/v1/d17-1082
Levine, Y., Wies, N., Jannai, D., Navon, D., Hoshen, Y., & Shashua, A. (2022). The inductive bias of in-context learning: Rethinking pretraining example design. In *International conference on learning representations*. https://openreview.net/forum?id=lnEaqbTJIRz.
Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-T., Rocktäschel, T., et al. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems*, *33*, 9459–9474.

Li, Z., Wang, Z., Wang, W., Hung, K., Xie, H., & Wang, F. L. (2025). Retrieval-augmented generation for educational application: A systematic survey. *Computers and Education: Artificial Intelligence*, *8*, 100417.

Liu, J., Liu, A., Lu, X., Welleck, S., West, P., Bras, R. L., Choi, Y., & Hajishirzi, H. 2022. Generated knowledge prompting for commonsense reasoning. In S. Muresan, P. Nakov, & A. Villavicencio (Eds.), *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: Long papers), ACL 2022, May 22–27, 2022* (pp. 3154–3169). Dublin, Ireland: Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.acl-long.225

Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., & Neubig, G. (2023). Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, *55*, 1–35.

Lyu, C., Wu, M., & Aji, A. (2024). Beyond probabilities: Unveiling the misalignment in evaluating large language models. In S. Li, M. Li, M. J. Zhang, E. Choi, M. Geva, P. Hase, & H. Ji (Eds.), *Proceedings of the 1st workshop on towards knowledgeable language models (KnowLLM 2024)* (pp. 109–131). Bangkok, Thailand: Association for Computational Linguistics. https://aclanthology.org/2024.knowllm-1.10.

Mihaylov, T., Clark, P., Khot, T., & Sabharwal, A. (2018a). Can a suit of armor conduct electricity? A new dataset for open book question answering. In E. Riloff, D. Chiang, J. Hockenmaier, & J. Tsujii (Eds.), *Proceedings of the 2018 conference on empirical methods in natural language processing, brussels, belgium, October 31 - November 4, 2018* (pp. 2381–2391). Association for Computational Linguistics. https://doi.org/10.18653/v1/d18-1260

Mihaylov, T., Clark, P., Khot, T., & Sabharwal, A. (2018b). Can a suit of armor conduct electricity? A new dataset for open book question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing* (pp. 2381–2391). Association for Computational Linguistics.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. 2013. Efficient estimation of word representations in vector space. In Y. Bengio, & Y. LeCun (Eds.), *1st international conference on learning representations, ICLR 2013, Workshop track proceedings, May 2–4, 2013*, Scottsdale, Arizona, USA. http://arxiv.org/abs/1301.3781.

Mitkov, R., Le An, H., & Karamanis, N. (2006). A computer-aided environment for generating multiple-choice test items. *Natural language engineering*, *12*, 177–194.

Pan, X., Sun, K., Yu, D., Chen, J., Ji, H., Cardie, C., & Yu, D. (2019). Improving question answering with external knowledge. In A. Fisch, A. Talmor, R. Jia, M. Seo, E. Choi, & D. Chen (Eds.), *Proceedings of the 2nd workshop on machine reading for question answering* (pp. 27–37). Hong Kong, China: Association for Computational Linguistics. https://aclanthology.org/D19-5804.

Pennington, J., Socher, R., & Manning, C. (2014). GloVe: Global vectors for word representation. In A. Moschitti, B. Pang, & W. Daelemans (Eds.), *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532–1543). Doha, Qatar: Association for Computational Linguistics. https://aclanthology.org/D14-1162.

Perkowski, E., Pan, R., Nguyen, T. D., Ting, Y.-S., Kruk, S., Zhang, T., O'Neill, C., Jablonska, M., Sun, Z., Smith, M. J., et al. (2024). Astrollama-Chat: Scaling astrollama with conversational and diverse datasets. *Research Notes of the AAS*, *8*, 7.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. In M. Walker, H. Ji, & A. Stent (Eds.), *Proceedings of the 2018 conference of the north American chapter of the association for computational linguistics: Human language technologies, volume 1 (long papers)* (pp. 2227–2237). New Orleans, Louisiana: Association for Computational Linguistics. https://aclanthology.org/N18-1202.

Pezeshkpour, P., & Hruschka, E. (2024). Large language models sensitivity to the order of options in multiple-choice questions. *Findings of the Association for Computational Linguistics: NAACL 2024*, 2006–2017.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, *1*, 9.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, *21*, :140:1–:140:67. http://jmlr.org/papers/v21/20-074.html.

Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using siamese BERT-networks. In K. Inui, J. Jiang, V. Ng, & X. Wan (Eds.), *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)* (pp. 3982–3992). Hong Kong, China: Association for Computational Linguistics. https://aclanthology.org/D19-1410.

Robertson, S., & Zaragoza, H. (2009). The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends in Information Retrieval*, *3*, 333–389. https://doi.org/10.1561/1500000019

Robinson, J., & Wingate, D. 2023. OpenReview.net. Leveraging large language models for multiple choice question answering. In *The eleventh international conference on learning representations, ICLR 2023, May 1–5, 2023*. Kigali, Rwanda. https://openreview.net/pdf?id=yKbprarjc5B.

Roziere, B., Gehring, J., Gloeckle, F., Sootla, S., Gat, I., Tan, X. E., Adi, Y., Liu, J., Remez, T., Rapin, J., et al., Code llama: Open foundation models for code, arXiv preprint arXiv:2308.12950, 2023. https://arxiv.org/abs/2308.12950.

Santurkar, S., Durmus, E., Ladhak, F., Lee, C., Liang, P., & Hashimoto, T. (2023). Whose opinions do language models reflect? In *International conference on machine learning* (pp. 29971–30004). PMLR.

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, *27*.

TheBloke. (2023). Mistral 7b openorca - gptq. https://huggingface.co/TheBloke/Mistral-7B-OpenOrca-GPTQ [Accessed 2024–June–30].

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al., Llama 2: Open foundation and fine-tuned chat models, arXiv preprint arXiv:2307.09288, 2023.

Wang, L., Ma, C., Feng, X., Zhang, Z., Yang, H., Zhang, J., Chen, Z., Tang, J., Chen, X., Lin, Y., et al. (2024a). A survey on large language model based autonomous agents. *Frontiers of Computer Science*, *18*, 186345.

Wang, P., Li, L., Chen, L., Cai, Z., Zhu, D., Lin, B., Cao, Y., Kong, L., Liu, Q., Liu, T., & Sui, Z. (2024b). Large language models are not fair evaluators. In L.-W. Ku, A. Martins, & V. Srikumar (Eds.), *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 9440–9450). Bangkok, Thailand: Association for Computational Linguistics. https://aclanthology.org/2024.acl-long.511.

Wang, X., Ma, B., Hu, C., Weber-Genzel, L., Röttger, P., Kreuter, F., Hovy, D., & Plank, B. 2024c. My answer is c": First-token probabilities do not match text answers in instruction-tuned language models. In L. Ku, A. Martins, & V. Srikumar (Eds.), *Findings of the association for computational linguistics, ACL 2024, bangkok, Thailand and virtual meeting, August 11–16, 2024* (pp. 7407–7416). Association for Computational Linguistics. https://aclanthology.org/2024.findings-acl.441.

Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., et al. (2022a). Emergent abilities of large language models. *Transactions on Machine Learning Research*, 08/2022, 1–30. https://openreview.net/forum?id=yzkSU5zdwD.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., Zhou, D., et al. (2022b). Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, *35*, 24824–24837.

Welbl, J., Liu, N., & Gardner, M. (2017). Crowdsourcing multiple choice science questions. In *Proceedings of the 3rd workshop on noisy user-generated text* (pp. 94–106). Association for Computational Linguistics.

Wu, S., Irsoy, O., Lu, S., Dabravolski, V., Dredze, M., Gehrmann, S., Kambadur, P., Rosenberg, D., & Mann, G., Bloomberggpt: A large language model for finance, arXiv preprint arXiv:2303.17564, 2023.

Xiao, S., Liu, Z., Zhang, P., Muennighoff, N., Lian, D., & Nie, J.-Y. (2024). C-pack: Packed resources for general Chinese embeddings. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval* (pp. 641–649).

Xu, K., Tin, J., & Youn, J. (2019). A BERT based model for multiple-choice reading comprehension. https://cs229.stanford.edu/proj2019spr/report/72.pdf.

Yenduri, G., Ramalingam, M., Selvi, G. C., Supriya, Y., Srivastava, G., Maddikunta, P. K. R., Raj, G. D., Jhaveri, R. H., Prabadevi, B., Wang, W., Vasilakos, A. V., & Gadekallu, T. R. (2024). GPT (generative pre-trained transformer)-a comprehensive review on enabling technologies, potential applications, emerging challenges, and future directions. *IEEE Access*, *12*, 54608–54649. https://doi.org/10.1109/ACCESS.2024.3389497

Zheng, C., Zhou, H., Meng, F., Zhou, J., & Huang, M. (2024). Large language models are not robust multiple choice selectors. In *The twelfth international conference on learning representations*. https://openreview.net/forum?id=shr9PXz7T0.