Enhancing Educational Value in Multiple-Choice QA via a Modular Explanation Generation Framework

Korea University COSE461 Final Project

Myeongchun Kim
Industrial and Management Engineering
Team 1
2020170806

Dongheon Yeon
School of Electrical Engineering
Team 1
2020170902

Seungmin Oh
School of Electrical Engineering
Team 1
2020170984

Jiwoo Park
School of Electrical Engineering
Team 1
2022170944

Abstract

Existing research on LLM-based content generation, particularly explanation generation, has primarily focused on correct answers, overlooking the educational value of distractors. While prior work has concentrated on generating explanations for learnersourced questions, it has not addressed misconceptions or reviewed the correctness of learner-generated answers. Similarly, datasets like PlausibleQA provide distractors ranked by their attractiveness along with explanations, but omit explanations for correct answers, creating an imbalance. Furthermore, generating explanations without access to correctness information increases the risk of hallucination, which is especially problematic in educational contexts.

In this work, we propose a novel framework that generates trustworthy and understandable explanations for both correct and incorrect answers. Our approach introduces a role-separated architecture that decouples answer prediction (classifier) from explanation generation (explainer). The explainer generates explanations through self-reflection on each option. Ultimately, our framework connects explanation generation with effective learning feedback, contributing to more interpretable and educationally useful AI systems.

1 Introduction

Large Language Models (LLMs) have demonstrated strong performance across a range of natural language processing tasks and are increasingly being adopted in educational contexts. In particular, they contribute to enhanced teaching and learning experiences through their ability to interpret and generate human-like language in applications such as question answering, content generation [1], and automated feedback. Among these applications, the generation of explanations for multiple-choice questions (MCQs) has recently gained attention, especially in *learner-sourced settings* where students create their own questions and explanations [2].

However, most existing approaches to explanation generation focus solely on correct answers, overlooking the educational value of plausible distractors that may reveal common misconceptions or learning gaps. This limitation is particularly problematic in learner-sourced platforms, where student-

generated answers are often unverified. As a result, LLMs run the risk of producing hallucinated explanations that may reinforce misunderstandings rather than correct them.

To address this issue, we argue that explanation generation should not rely solely on correctness labels, which are often absent or unreliable in learner-sourced contexts. Instead, explanations should be generated for all answer choices, enabling more balanced and pedagogically effective feedback for learners.

To this end, we propose a novel framework that decouples answer verification from explanation generation. A classification module first predicts the correctness of each candidate option, and then an explanation generator produces justifications that reflect both the plausibility and educational value of each choice. This design enables high-quality explanation generation even in the absence of ground-truth labels, making the system applicable to both expert-curated and learner-sourced datasets.

2 Related Work

2.1 Content Generation for Educational QA

Prior studies have focused on the automated generation of multiple-choice questions (MCQs) that maintain topical alignment, structural diversity, and educational relevance. For example, Xiao et al. [3] fine-tuned LLMs on textbook data to improve coherence in reading comprehension tasks.

Despite these advances, most approaches focus on question and distractor generation, with limited attention to explanations for incorrect answers, which restricts the educational value of such systems.

2.2 Hallucination in Explanation Generation

Hallucinations pose a significant challenge in educational QA. Li et al. [4] mitigated semantic drift by grounding explanations in extractive rationales. Lin et al. [5] introduced TruthfulQA to evaluate models' susceptibility to generating false information. Ji et al. [6] proposed a self-reflection loop to improve consistency by iteratively verifying model outputs.

While these methods enhance reliability, they do not explicitly separate classification from explanation, nor do they address the instructional value of explanations for incorrect answers.

3 Approach

Our goal is to generate pedagogically effective explanations for all answer choices in multiple-choice questions (MCQs), regardless of correctness. To this end, we build on the self-reflective reasoning paradigm of Ji et al. [6] and introduce a role-separated classifier–explainer framework that emphasizes logical consistency and understandability. Specifically, our framework include:

1. Introducing a classifier to explicitly predict answer correctness.
2. Generating knowledge and explanations tailored to each answer choice.
3. Refining generated explanation to improve consistency and understandability.

The overall architecture follows the flow:

Each input consists of a question q and a set of candidate answers $\{a_1, \ldots, a_4\}$. For each option a_i, the system produces: a binary correctness label y_i, background knowledge k_i related to the question, and a natural language explanation e_i that justifies the label.

3.1 Step 1: Classification Module (Ours)

We introduce a lightweight classifier $\mathcal{C}$ to independently predict the correctness $y_i \in \{0, 1\}$ of each answer a_i given the question q:

$$y_i = \mathcal{C}(q, a_i)$$

By decoupling answer prediction (handled by the classifier) from explanation generation (performed within the generation loop), our framework not only streamlines the reasoning pipeline and enhances

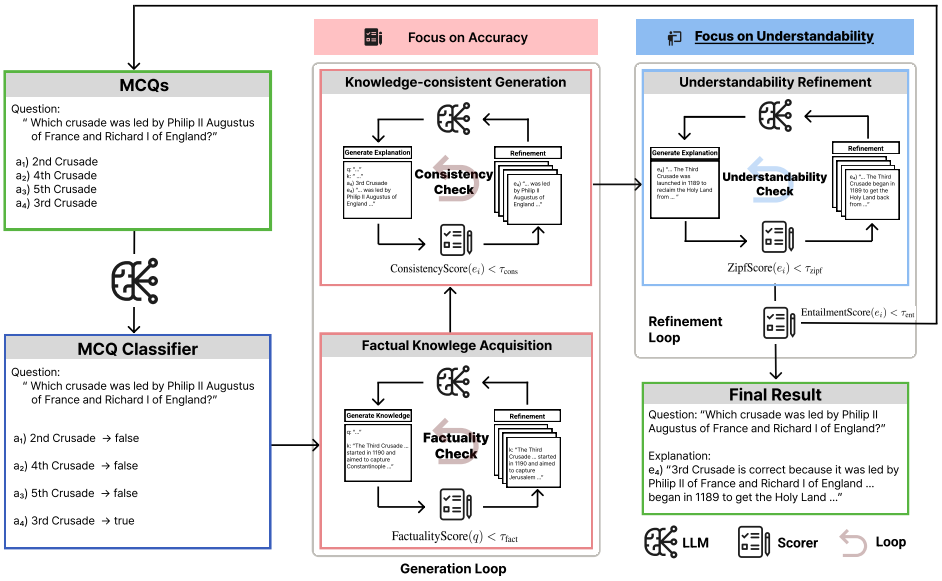

Figure 1: Overview of the proposed architecture.

interpretability, but also enables controlled comparisons between two explanation modes: one in which the generation model is informed of the predicted correctness of each option, and one in which it is not. This allows us to directly assess the impact of correctness signals on factual accuracy, logical coherence, and explanatory capability of the generated responses.

3.2 Step 2: Generation Loop (Baseline)

We adopt the self-reflection-based generation loop from Ji et al. [6] as our baseline. The original framework consisted of three iterative modules. In our revised design, the generation loop focuses on two main modules: (1) *Factual Knowledge Acquisition*, and (2) *Knowledge-Consistent Answer Generation*. The third module, *Question-Entailment Answering*, is applied separately at the final refinement stage. This entailment verification step, which was originally performed within the loop, is thus decoupled to allow post hoc validation of logical consistency.

3.2.1 Factual Knowledge Acquisition Loop

This stage generates background knowledge relevant to the input question. The model iteratively generates and refines background knowledge based on a factuality assessment. Factuality is evaluated using a GPT-2-based GPTScore model [7], which estimates the factual alignment between the generated content and the reference explanation. If the factuality score falls below a predefined threshold, the loop is repeated to improve the factual quality of the generated knowledge.

3.2.2 Knowledge-Consistent Answer Generation Loop

Using the generated background knowledge and the classifier's true/false prediction y_i for each candidate answer a_i, the system constructs an explanation that aligns with the knowledge and justifies why the answer is correct or incorrect. The model ensures consistency between the generated answers and the validated background knowledge. Response consistency is measured using the CTRLEval metric [8], which leverages Inverse Word Frequency (IWF) and topic-aware prompt templates to quantify the alignment between the generated justification and the underlying knowledge.

3.3 Step 3: Refinement Loop (Ours)

We propose an additional refinement loop to address two limitations of the baseline: (1) explanations may still be logically inconsistent or factually incorrect, and (2) explanations may be difficult for

learners to understand. Our refinement loop thus comprises two modules to refine explanation e_i: consistency checking and understandability checking.

3.3.1 Understandability Refinement Loop

To evaluate the readability and learner-friendliness of explanations, we introduce a Zipf-based word frequency scoring mechanism. This approach assumes that explanations composed of simpler words are generally easier for learners to understand.

1. First, we extract word frequency statistics from the Brown Corpus, a widely-used general-purpose English corpus.
2. Each explanation e_i is tokenized into individual words, and each word is assigned a *Zipf score* based on its corpus frequency rank.
3. The average Zipf score across all tokens in e_i is computed:

$$\text{ZipfScore}(e_i) = \frac{1}{|T|} \sum_{t \in T} \text{Zipf}(t)$$

   where T is the set of tokens in the explanation.

Higher Zipf scores indicate the use of more common words (e.g., "eat", "run"), which are considered easier to understand. In contrast, explanations dominated by rare or technical terms (e.g., "mitochondrial", "geopolitical") yield lower Zipf scores. If the average score is below a threshold, we trigger rephrasing to increase accessibility.

3.3.2 Question-Entailment Answering Loop

This module verifies whether each generated explanation e_i logically follows from the input (q, a_i). We relocate this entailment check from the original generation loop [6] to the refinement stage. Semantic entailment is evaluated using a pretrained sentence embedding model (`multi-qa-MiniLM-L6-cos-v1`) from SentenceTransformers, which computes dot product similarity between the question and explanation embeddings. If the entailment score falls below a threshold τ_1, the system considers the explanation invalid and returns to the initial *Factual Knowledge Acquisition Loop* to restart the generation process.

4 Experiment

4.1 Data

We base our experiments on the publicly available **PlausibleQA** dataset [9], which is constructed from existing QA corpora including TriviaQA [10], Natural Questions [11], and WebQuestions [12]. The dataset consists of 10,000 multiple-choice questions, each paired with one correct and ten incorrect candidate answers. For each incorrect answer, human annotators provide a natural language explanation, a plausibility score, and pairwise preference comparisons.

However, the dataset lacks justifications for correct answers, which limits its educational applicability. To address this, we augment each correct answer with a concise, LLM-generated justification using `deepseek-ai/deepseek-llm-7b-chat`. Prompts are designed to elicit brief, focused explanations that do not reference distractors and aim for pedagogical clarity (see Appendix A).

We also remove pairwise ranking metadata, filter out entries with emoji or non-textual characters, exclude questions involving identifiable individuals, and convert all items to a standard four-choice (one correct, three incorrect) format. After filtering and restructuring, the final dataset comprises 6,071 questions, each with four labeled options and aligned explanations. This format supports our classifier–explainer architecture by enabling binary correctness prediction and explanation generation for each choice.

4.2 Evaluation Methods

To assess the quality of generated explanations, we employ traditional quantitative metrics that capture different aspects of textual similarity between ground truth and model-generated justifications:

BERTScore evaluates semantic similarity by comparing contextualized token embeddings from BERT-base-uncased between reference and candidate justifications. We compute pairwise cosine similarities between all tokens, and aggregate them using F1 metrics.

ROUGE-L evaluates the longest common subsequence between reference and candidate texts, capturing structural similarity and word overlap patterns while accounting for word order.

All metrics are computed at the candidate level and aggregated across the dataset to provide overall performance statistics.

4.2.1 Keyword Match Score

To assess the factual alignment of generated explanations beyond semantic similarity and structural overlap, we introduce **Keyword Match Score**, an evaluation metric designed to quantify the retention of essential informational content. Unlike embedding-based metrics such as BERTScore and ROUGE-L, which focus on semantic similarity and allow for paraphrasing, Keyword Match assesses whether specific factual terms are explicitly preserved. This makes it particularly useful in educational QA, where missing key information can significantly reduce the instructional value of an explanation.

For each (q, a_i) pair, we extract up to three keywords w_1, w_2, w_3 representing the essential content of the reference explanation. The Keyword Match is defined as the number of these keywords that are found in the corresponding generated justification for a_i, yielding a score from 0 to 3. This provides a simple yet interpretable signal for how well the explanation preserves key factual elements.

The keyword extraction pipeline is implemented using the `spaCy` NLP library and includes the following steps:

- Only tokens labeled as `NOUN` or `PROPN` are considered, along with named entities up to three words in length.
- Common stopwords (as defined in NLTK) and generic QA terms (e.g., *answer*, *true*, *choice*) are excluded.
- Tokens are normalized by lowercasing and removing punctuation, possessives, and the article "the".
- Redundant or overlapping candidates are filtered using `difflib.SequenceMatcher`, which applies Ratcliff/Obershelp pattern matching. Keyword pairs with a similarity ratio greater than 0.8 are treated as duplicates.

4.2.2 TOEIC vocabulary based Understandability Evaluation

We design a readability metric based on TOEIC vocabulary difficulty levels to evaluate the accessibility and comprehensibility of generated justifications. This allows us to objectively measure text complexity in relation to standardized English proficiency levels.

Vocabulary Difficulty Weighting We establish a weighted scoring system based on five TOEIC vocabulary difficulty categories:

- **Basic TOEIC Words**: +2.0 (fundamental vocabulary, positive contribution to readability)
- **Core Frequent Words**: +1.0 (intermediate vocabulary, moderate positive contribution)
- **Standard Vocabulary**: 0.0 (neutral words)
- **TOEIC 800-level Words**: -1.0 (advanced vocabulary, negative impact on readability)
- **TOEIC 900-level Words**: -2.0 (highly advanced vocabulary, significant negative impact)

This weighting scheme reflects the intuition that texts containing predominantly basic vocabulary are more accessible to a broader audience, while those with advanced vocabulary may present comprehension challenges.

Scoring Methodology We assess readability through a systematic five-step process. First, the output sentences are tokenized into individual word tokens. Second, part-of-speech (POS) tagging is applied to identify the grammatical roles of the words. Third, lemmatization is performed to convert words into their canonical forms based on their POS tags. Fourth, each lemmatized word is compared

against a TOEIC vocabulary dataset, with predefined weights assigned to matched terms and zero to unmatched ones. Finally, the readability score is computed by normalizing the sum of all assigned weights by the total word count.

The TOEIC vocabulary based readability metric provides a normalized, interpretable score that complements traditional semantic similarity measures by assessing the comprehensibility of justifications: positive values indicate high readability with simple vocabulary, negative values reflect lower readability due to advanced word usage, and scores near zero suggest moderate difficulty aligned with standardized language proficiency levels.

4.3 Experimental details

For our experiments, we used the vLLM library[13] for efficient LLM inference. We used `deepseek-ai/deepseek-llm-7b-chat` model and set the `tensor_parallel_size` to 2 for parallel decoding. Generation was performed with sampling parameters: `temperature=1.0`, `top_p=1.0`, `top_k=1`, `num_beams=1`, and `max_new_tokens=128` to control randomness and output length. Our method consists of three sequential generation-refinement loops: knowledge acquisition, explanation generation, and understandability enhancement. At each step, outputs are re-evaluated using predefined thresholds—factuality (-1), consistency (-5), entailment (0.8), and understandability (3)—and the loop continues until these criteria are met or the iteration limit is reached. Due to hardware limitations, we randomly sampled 200 questions from the final dataset of 6,071 questions for experiment.

All prompts were formatted using structured templates designed specifically for classification and explanation generation tasks, following a system/user/assistant format compatible with instruction-tuned chat models. Initial prompts elicit background knowledge, followed by justification generation and an optional refinement step to enhance human understandability. The full set of templates, covering all stages from dataset construction to explanation generation, is provided in Appendices A.

4.4 Results

To assess the effectiveness of the classifier module independently, we evaluated its performance over a held-out set of 200 question–answer cases. The classifier achieved an accuracy of 84.5%, correctly identifying the true/false labels for 169 out of 200 candidate answers.

Classifier	Generation Loop	Refinement Loop	BERT Score	ROUGE-L	Keyword match	TOEIC Score*
×	✓	×	0.7571	0.2121	1.299	0.7006
✓	✓	×	0.7638	0.2211	1.313	0.7567
✓	✓	✓	0.7588	0.2073	1.095	0.8111

Table 1: Evaluation results for the ablation study on our framework. *TOEIC Score indicates the readability metric based on TOEIC vocabulary difficulty levels.

Table 1 presents the results of an ablation study evaluating the impact of three key components: the Classifier, Generation Loop, and Refinement Loop. We analyze the results based on four evaluation metrics: BERT Score, ROUGE-L, Keyword Match, and TOEIC vocabulary based readability score. Our baseline corresponds to the use of the Generation Loop alone, while our full architecture incorporates all three components: the Classifier, Generation Loop, and Refinement Loop.

Introducing the classifier component leads to noticeable improvements in both BERT Score (from 0.7571 to 0.7638) and ROUGE-L (from 0.2121 to 0.2211), indicating that explicitly predicting the correctness of each option prior to explanation generation helps the model produce more semantically aligned and structurally accurate justifications. Keyword Match also increases slightly (1.299 to 1.313), further supporting the idea that the classifier helps ground the explanations in content that is more relevant to the question and answer candidates.

The addition of the Refinement Loop introduces a notable trade-off. While BERT Score, ROUGE-L, and Keyword Match slightly decrease to 0.7588, 0.2073, and 1.095 respectively, the TOEIC-based readability score significantly improves from 0.7567 to 0.8111. This indicates that the refinement

process enhances the *readability* and *lexical simplicity* of the generated explanations by promoting the use of simpler expressions and more general vocabulary.

In particular, the drop in Keyword Match suggests a potential loss of specific terms containing core concepts, which may be replaced by easier alternatives during refinement. Nevertheless, the increase in TOEIC score implies that the resulting explanations become easier to understand for non-expert or general users. This is especially beneficial in practical scenarios such as education, tutoring, or automated feedback, where user-friendliness is a key priority. Thus, the refinement process plays an essential role in improving the accessibility and acceptability of explanations, even if it slightly compromises certain semantic alignment metrics.

The observed results reveal a fundamental trade-off. The Generation Loop and Classifier contribute more to semantic accuracy (BERT and ROUGE), while the Refinement Loop enhances readability and vocabulary appropriateness (TOEIC Score). Notably, the configuration with all three components yields the highest TOEIC Score, but slightly reduced alignment metrics. This suggests that careful balancing is required depending on the target application—whether factual precision or user comprehension is prioritized.

Overall, these results highlight the complementary contributions and limitations of each component. The classifier improves factual grounding and generation focus, while the refinement module promotes accessibility. However, both modules exhibit sensitivity to upstream prediction quality and lack robust fallback mechanisms. These findings underscore the importance of balancing factual precision, semantic similarity, and educational clarity in the design of explanation-generation systems.

5 Analysis

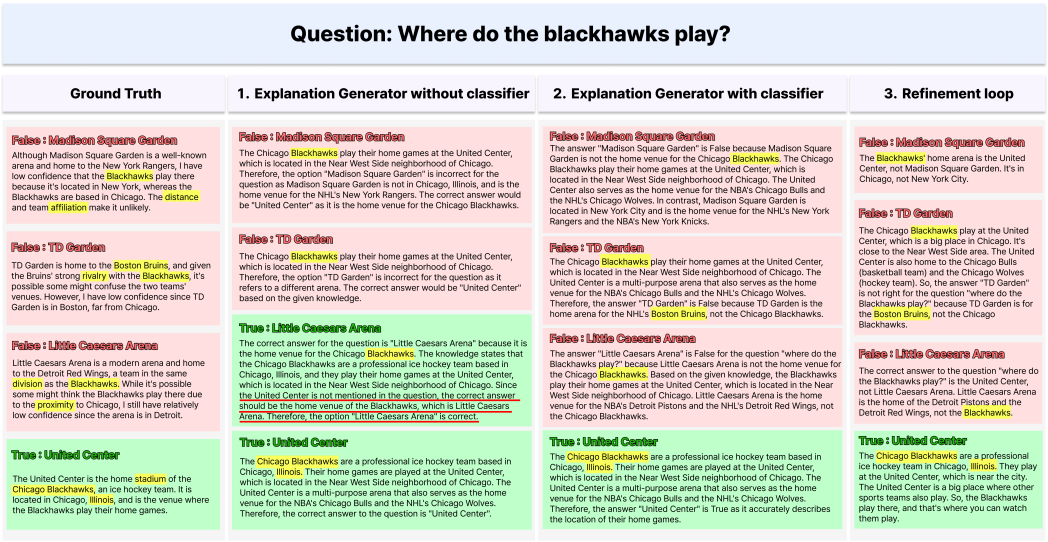

Figure 2: Example of Generated Explanation

Figure 2 illustrates how the output evolves through each stage of our pipeline. Without the classifier (Column 2), the model correctly rejects some distractors like *Madison Square Garden*, but mistakenly accepts *Little Caesars Arena*, showing a tendency to overgeneralize when distractors are semantically plausible. This illustrates the model's difficulty in grounding judgments without explicit correctness signals.

With the classifier (Column 3), explanations become more accurate and focused. The model now correctly marks *Little Caesars Arena* as incorrect, and uses the classifier's binary prediction to frame more decisive justifications. For instance, the output now highlights venue misalignment and team associations more clearly, anchoring its explanation in the predicted label.It also catches more keywords highlighted in Yellow for *TD Garden*.

After applying the Refinement Loop (Column 4), explanations are rephrased to enhance clarity and reduce redundancy. For example, verbose statements such as "The Chicago Blackhawks play their home games at the United Center..." are distilled into concise forms like "The Blackhawks' home arena is the United Center." While these edits improve accessibility, they may also remove specific phrases (e.g., "Near West Side neighborhood"), slightly reducing keyword match and factual density.

Despite minor information loss, the refinement loop produces explanations that are more accessible and readable—especially beneficial in educational contexts. This qualitative observation aligns with the rise in TOEIC Score seen in Table 1, supporting the loop's effectiveness at simplifying complex or redundant outputs without significantly compromising content accuracy. Specifically:

Sentence length and redundancy reduction In the pre-refinement example, sentences such as "The Chicago Blackhawks play their home games at the United Center, which is located in the Near West Side neighborhood of Chicago. The United Center also serves as…" repeat the same facts multiple times. The post-refinement example instead states "The Blackhawks' home arena is the United Center, not Madison Square Garden." in a single, concise sentence.

Balance between keyword retention and clarity improvement Post-refinement retains essential terms ("United Center," "Chicago Blackhawks," etc.) and removes superfluous wording—improving readability and clarity without affecting quantitative keyword-match metrics.

Potential drawback of background removal While post-refinement omits extensive venue histories to sharpen focus, this simplification can be a limitation. Explaining why a distractor is incorrect often requires some background context. For learners, providing concise but sufficient information about each false option may be more instructive than maximal brevity—thus, simplification may not be the optimal choice for educational clarity.

6 Conclusion

In this work, we introduce a three-stage framework comprising a classifier, a generation loop, and a refinement loop to produce high-quality, factually grounded explanations for multiple-choice QA. Incorporating the classifier yields consistent gains in semantic overlap and factual precision, while the generation loop enhances structural coherence. The refinement loop, though introducing a minor trade-off in overlap-based metrics, substantially improves user-centric readability. Taken together, these components constitute a flexible and effective framework for generating educationally valuable explanations.

While our proposed framework demonstrates meaningful improvements in explanation generation for multiple-choice questions, several directions remain open for future exploration. First, although the refinement loop improves readability, it occasionally simplifies content at the expense of factual specificity. A promising extension would be a controllable refinement mechanism that balances lexical simplicity with information density based on the learner's proficiency level or task context. This could involve integrating user-adaptive models or fine-grained readability constraints.

Second, our framework currently treats each answer explanation independently. Future work could explore cross-option reasoning to enhance contrastive explanations—explicitly stating why one answer is more valid than others, which may help learners better internalize conceptual distinctions.

These extensions would help broaden the applicability and robustness of explanation generation systems in real-world educational environments.

References

[1] Shen Wang, Tianlong Xu, Hang Li, Chaoli Zhang, Joleen Liang, Jiliang Tang, Philip S Yu, and Qingsong Wen. Large language models for education: A survey and outlook. *arXiv preprint arXiv:2403.18105*, 2024.

[2] Qiming Bao, Juho Leinonen, Alex Yuxuan Peng, Wanjun Zhong, Gaël Gendron, Timothy Pistotti, Alice Huang, Paul Denny, Michael Witbrock, and Jiamou Liu. Exploring iterative enhancement for improving learnersourced multiple-choice question explanations with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 28955–28963, 2025.

[3] Changrong Xiao, Sean Xin Xu, Kunpeng Zhang, Yufang Wang, and Lei Xia. Evaluating reading comprehension exercises generated by llms: A showcase of chatgpt in education applications. In *Proceedings of the 18th workshop on innovative use of NLP for building educational applications (BEA 2023)*, pages 610–625, 2023.

[4] Chenliang Li, Bin Bi, Ming Yan, Wei Wang, and Songfang Huang. Addressing semantic drift in generative question answering with auxiliary extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 942–947, 2021.

[5] Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*, 2021.

[6] Ziwei Ji, Tiezheng Yu, Yan Xu, Nayeon Lee, Etsuko Ishii, and Pascale Fung. Towards mitigating hallucination in large language models via self-reflection. *arXiv preprint arXiv:2310.06271*, 2023.

[7] Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. Gptscore: Evaluate as you desire, 2023.

[8] Pei Ke, Hao Zhou, Yankai Lin, Peng Li, Jie Zhou, Xiaoyan Zhu, and Minlie Huang. Ctrleval: An unsupervised reference-free metric for evaluating controlled text generation, 2022.

[9] Jamshid Mozafari, Abdelrahman Abdallah, Bhawna Piryani, and Adam Jatowt. Wrong answers can also be useful: Plausibleqa–a large-scale qa dataset with answer plausibility scores. *arXiv preprint arXiv:2502.16358*, 2025.

[10] Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[11] Chris Alberti, Kenton Lee, and Michael Collins. A BERT baseline for the natural questions. *CoRR*, abs/1901.08634, 2019.

[12] Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. Semantic parsing on Freebase from question-answer pairs. In David Yarowsky, Timothy Baldwin, Anna Korhonen, Karen Livescu, and Steven Bethard, editors, *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544, Seattle, Washington, USA, October 2013. Association for Computational Linguistics.

[13] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

A Appendix: Prompt Inputs Used in Experiments

A.1 Classifier Prompt

```
You are a helpful assistant.  You will be given a question and one
proposed answer.
Reason about it if necessary, then clearly conclude your response
with:
Answer:  True or Answer:  False.
Only the final line should contain this answer keyword.


Question:  {question}
Proposed Answer:  {candidate_answer}

Is this the correct answer?
Answer:
```

A.2 Dataset Prompt

```
You are a concise assistant.  Given the following question and
explanation,
provide a short justification (1-2 sentences) for why the provided
answer is correct.
Focus only on the correct answer and how it matches the question.
Do not mention other options.


Question:  {question}
Explanation:  {combined justifications of all incorrect answers}

Why is 'correct answer' the correct answer?
```

A.3 Knowledge Generation Prompt

```
<|system|>
You are a helpful assistant that provides background knowledge to
help answer questions.
<|user|>
Provide background knowledge to answer the given question:
``{question}''
```

A.4 Explanation Generation Prompt

```
<|system|>
You are an expert tutor who explains MCQ choices using background
knowledge.
<|user|>
Based on the following knowledge:  ``{knowledge}'', explain why the
answer ``{candidate}'' is {True/False} for the question:  ``{question}''
in a single paragraph.

Example 1:
Question:  What is the capital of France?
Choice:  Paris
Is Correct?:  True
Knowledge:  France's capital is Paris.
Explanation:  Paris is the capital of France, so this answer is
correct.

Example 2:
Question:  What is the capital of France?
Choice:  Berlin
Is Correct?:  False
Knowledge:  Berlin is the capital of Germany, not France.
```

```
Explanation:  Berlin is the capital of Germany, not France.
Therefore, it is not the correct answer.
```

A.5 Understandability Refinement Prompt

```
<|system|>
You are an expert at simplifying complex text.  Your task is to
rewrite explanations using simple, common English words that are
easy to understand.

<|user|>
Please simplify this explanation by:
1.  Replacing difficult words with common alternatives
2.  Breaking long sentences into shorter ones
3.  Using everyday vocabulary instead of technical terms
4.  Keeping the same meaning but making it much easier to read


Original explanation:  {justification}
Simplified explanation:
```

B Appendix: Data example

ID: trivia254
Question: Which crusade was led by Philip II Augustus of France and Richard I of England?
Candidate: 2nd Crusade **Predicted:** False **Ground Truth:** False
Justification: *Although the 2nd Crusade was a major crusade, it was led by King Louis VII of France and King Conrad III of Germany, not Philip II Augustus or Richard I. My confidence is low because the leaders do not match, but it's not zero since it's a plausible time period.*
Candidate: 4th Crusade **Predicted:** False **Ground Truth:** False
Justification: *The 4th Crusade was led by Boniface I, Marquess of Montferrat, and Doge Enrico Dandolo of Venice, not Philip II Augustus or Richard I. My confidence is very low because the leaders and the direction of the crusade (against Constantinople) do not align with the given information.*
Candidate: 5th Crusade **Predicted:** False **Ground Truth:** False
Justification: *The 5th Crusade was led by King Andrew II of Hungary, Duke Leopold VI of Austria, and John of Brienne, not Philip II Augustus or Richard I. My confidence is very low because the leaders and the time period do not match.*
Candidate: 3rd Crusade **Predicted: True Ground Truth: True**
Justification: *The 3rd Crusade was led by Philip II Augustus of France and Richard I of England, making it the correct answer to the given question.*