Think Both Ways: Teacher-Student Bidirectional Reasoning Enhances MCQ Generation and Distractor Quality

Yimiao Qiu[1], **Yang Deng**[2], **Quanming Yao**[3], **Zhimeng Zhang**[1], **Zhiang Dong**[1], **Chang Yao**[1] and **Jingyuan Chen**[1]

[1]Zhejiang University
[2]Singapore Management University
[3]Tsinghua University
22451052@zju.edu.cn

Abstract

Generating high-quality Multiple Choice Questions (MCQs) remains challenging for educational tools due to the need for contextual relevance and plausible distractors. Existing methods still struggle with these dual requirements, leading to questions that lack depth and distractors that are either too obvious or irrelevant. In this paper, we propose **BiFlow**, a novel framework that integrates bidirectional reasoning perspectives: *teacher reasoning* generates contextually relevant questions and plausible distractors, while *student reasoning* evaluates question clarity and the misleading nature of the distractors. To further enhance reasoning, we introduce **PathFinder**, a mechanism that employs breadth-first search and Chain-of-Thought (CoT) strategies to explore diverse reasoning paths, improving both the quality and diversity of generated questions and distractors. Additionally, we enrich the FairytaleQA dataset to FairytaleMCQ with high-quality distractors, providing a robust benchmark for MCQ generation. Experimental results demonstrate that BiFlow outperforms existing methods, particularly in generating text-grounded questions and high-quality distractors for narrative contexts, highlighting its value in educational applications. Project Page can be found here.

1 Introduction

The task of generating Multiple Choice Questions (MCQs) has garnered significant attention due to its vital role in educational tools, assessments, automated evaluation systems and resource generation (Ha and Yaneva, 2018; Dong et al., 2025; Lv et al., 2025; Dai et al., 2024; Chen et al., 2024a). Over time, question generation has evolved from manual creation by educators (Lindberg et al., 2013; Labutov et al., 2015; Hu et al., 2023) to being automated through AI-driven systems (Du et al., 2017; Yao et al., 2021; Jamshidi and Chali, 2025). This integration of AI holds the potential to drastically reduce the time and costs associated with generating large sets of questions while simultaneously enhancing the diversity and accuracy of assessments.

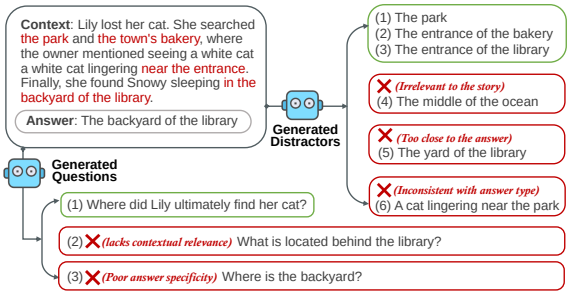

Figure 1: Illustrative examples of MCQs.

As shown in Figure 1, effectively generating MCQs is non-trivial due to two critical challenges: 1) producing questions that are contextually relevant to the given material and 2) generating distractors that are both plausible yet incorrect. Distractors must appear to be reasonable choices, but still be clearly distinguishable from the correct answer. This balance between plausibility and incorrectness requires a sophisticated understanding of both the subject matter and common misconceptions. High-quality distractors not only need to mimic potential mistakes but also contribute to the overall difficulty and validity of the assessment.

Various approaches have been proposed to address these challenges. Modern neural models like sequence-to-sequence architectures (Pyatkin et al., 2021) improve syntactic quality of question generation (QG) but struggle with contextual relevance and generating sufficiently challenging distractors. Additionally, distractor generation (DG) remains under-explored, despite its critical role in MCQ design. Existing methods (Ren and Zhu, 2021; Le Berre et al., 2022) often rely on word-level semantic similarity from lexical resources, prioritizing simple word/phrase options over deeper conceptual relationships. This leads to distractors that are overly obvious, irrelevant, or inadequately misleading, reducing MCQ effectiveness. Another lim-

Findings of the Association for Computational Linguistics: ACL 2025, pages 8240–8253
July 27 - August 1, 2025

itation is that these models typically fail to consider the interaction between the generated question and its distractors, which can lead to incoherent question-distractor pairs.

In this paper, we introduce **Bi**directional Reasoning **Flow** (**BiFlow**), a novel framework that addresses these limitations in MCQ generation by unifying two complementary reasoning perspectives: 1) *Teacher reasoning* leverages deep content understanding to generate candidate questions and distractors, ensuring alignment with learning objectives, contextual relevance and misleading requirements. 2) *Student reasoning* evaluates these questions and their associated distractors through the lens of a learner attempting to solve them, calibrating difficulty and identifying plausible misconceptions. By incorporating both perspectives, BiFlow produces questions that are both pedagogically sound and cognitively challenging, and distractors that are more misleading and contextually relevant. To further enhance the reasoning process, we introduce **PathFinder**, a guiding mechanism that utilizes breadth-first search and Chain-of-Thought (CoT) strategies to explore a wider array of potential solutions. PathFinder ensures that both question generation and distractor generation benefit from diverse reasoning paths, leading to more robust and varied MCQs. The resulting framework provides a significant improvement over prior methods by fostering a more holistic generation process that incorporates multiple layers of reasoning, ultimately improving the quality and diversity of MCQs and their distractors.

Our main contribution are summarized as:

- We propose BiFlow, a novel framework that combines teachers' forward-thinking process with the students' backward reasoning to enhance MCQ generation and distractor quality, ensuring both pedagogical value and contextual relevance.
- We introduce PathFinder to leverage breadth-first search and CoT prompting to systematically explore diverse reasoning paths, improving the quality and diversity of generated content.
- We extend the FairytaleQA dataset (Wang et al., 2022) to FairytaleMCQ by adding high-quality distractors to each question via a human-agent collaborative annotation pipeline. FairytaleMCQ provides a robust benchmark for evaluating multiple-choice question generation models.
- Experiments on authentic datasets reveal that BiFlow surpasses current methods in generating contextually relevant questions and plausible yet incorrect distractors, demonstrating its practical educational value and effectiveness in real-world applications.

2 Related work

Educational Question Generation. Automated educational QG has emerged as a vital tool for enhancing learning outcomes and reducing educators' workloads. The field has evolved significantly from early template-based and syntax-driven methods (Heilman and Smith, 2010; Lindberg et al., 2013; Labutov et al., 2015; Kumar et al., 2024) to advanced approaches utilizing deep neural networks (Du et al., 2017; Xia et al., 2023; Eo et al., 2023). Recent advancements have been driven by the integration of pre-trained and large language models (Wang et al., 2022; Wu et al., 2022; Zeng et al., 2023; Bulathwela et al., 2023; Mucciaccia et al., 2025), via fine-tuning strategies to enhance QG capabilities. These models are widely applied in conversational systems (Gao et al., 2019; Pan et al., 2019) and intelligent tutoring systems (Yao et al., 2021; Ang et al., 2023), showing their versatility across educational applications.

Distractor Generation. In the field of DG for MCQs, research primarily focuses on retrieval-based and generation-based methods. Retrieval-based approaches (Le Berre et al., 2022; Yu et al., 2024) rely on knowledge bases or question corpora to identify and rank distractors semantically or lexically similar to the correct answer. While effective in specific contexts, these methods require extensive manual effort to design and implement features, limiting their scalability. Generation-based methods (Manakul et al., 2023; Bitew et al., 2022; Qiu et al., 2020) leverage deep learning models, such as sequence-to-sequence architecturesto directly generate distractors. Recent advancements (Luo et al., 2024) have also explored the use of LLMs to combine retrieval and generation techniques, enhancing the relevance and quality of generated distractors.

Chain-of-Thought Reasoning. CoT prompting enhances the reasoning capabilities of LLMs by decomposing complex problems into intermediate steps. This technique has demonstrated effectiveness across various domains, including multimodal reasoning (Chen et al., 2024b; Zhang et al., 2023; Huang et al., 2025), education (Buhnila et al., 2025; Tao et al., 2025), and decision sup-

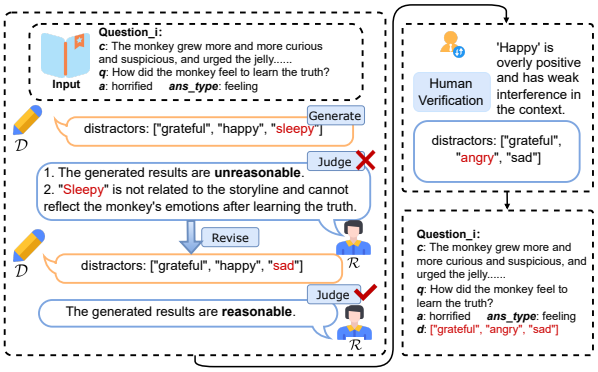

Figure 2: Workflow of FairytaleMCQ construction.

port systems (Cao et al., 2025; Liu et al., 2025). Recent CoT framework has evolved with several prompting strategies. For example, Chain-of-Discussion (Tao et al., 2025) gathers relevant evidence for well-reasoned and helpful responses, while Chain-of-Editions (Zhang et al., 2024) adopts CoT to enhance LLMs' reasoning ability for SQL query generation.

3 FairytaleMCQ Dataset Construction

The task of Multiple Choice Question (MCQ) Generation involves creating a question q based on a given context c and a correct answer ans, along with a set of distractors D'. These distractors should be plausible yet incorrect to challenge the user's understanding, while ensuring the correct answer ans is the most accurate choice.

Narrative comprehension, with its intricate storylines and diverse characters, provides a rich context for complex reasoning tasks. However, current narrative comprehension datasets (Kočiskỳ et al., 2018; Lal et al., 2021) focus mainly on question-answer pairs without distractors, making it difficult to evaluate models designed for MCQs generation. At the same time, current mainstream MCQ datasets have relatively short contexts, making them unsuitable for complex reasoning questions. To address this gap, we extend the existing FairytaleQA (Wang et al., 2022) dataset by generating distractors for each question. This enables a robust evaluation framework for our model. Below, we describe the process for constructing the extended dataset.

Initial Dataset. The original FairytaleQA is from real-world reading comprehension corpora. Each instance includes a passage of context c, a question q, a correct answer ans, the type of the answer `ans_type` and an associated question `difficulty`.

Distractor Generation Pipeline. Creating high-quality distractors presents a significant challenge, which must meet two essential criteria: 1) *Deceptiveness*: They should be syntactically and semantically similar to the correct answer, making them plausible yet incorrect. 2) Consistency: They must not only be contextually relevant but also consistent with the `ans_type` to ensure alignment with the question's structure and type.

To achieve this, we introduce a two-step Distractor Generation Pipeline involving two agents: a *Distractor Generator* and a *Result Checker*, both implemented using GPT-4.

`Distractor Generator` $\mathcal{D}$: This agent is responsible for producing three plausible but incorrect distractors based on the context of the question (Appendix A.1). These distractors are derived from the context c, ensuring they are syntactically and semantically aligned with the correct answer ans, yet sufficiently misleading to challenge the user's understanding of the material.

`Result Checker` $\mathcal{R}$: This agent acts as a supervisory layer, ensuring that the distractors generated by $\mathcal{D}$ meet specific quality standards . It evaluates the distractors according to three key criteria: syntactic correctness, consistency with `ans_type`, and plausibility as an incorrect option (Appendix A.2).

As illustrated in Figure 2, the process is iterative: $\mathcal{D}$ generates distractors, and $\mathcal{R}$ provides feedback. If any distractor is unsuitable, $\mathcal{R}$ suggests revisions. This feedback loop continues until all distractors meet the quality standards.

Human Verification. To ensure the distractors meet human-level quality standards, we incorporate a final human verification step. After the automatic generation phase, two graduate students in computer science perform two rounds of evaluations to assess the distractors' relevance, accuracy, and overall quality (Appendix A.3). Any distractors identified as unsuitable are revised accordingly.

4 Method

We introduce BiFlow, a novel framework designed to enhance the generation of MCQs and distractors by integrating dual perspectives of teacher and student reasoning. As shown in Figure 3, BiFlow, which consists of a question generator (§ 4.1) and a distractor generator (§ 4.2), aims to leverage both the forward-thinking approach of a teacher and the backward reasoning typical of students.

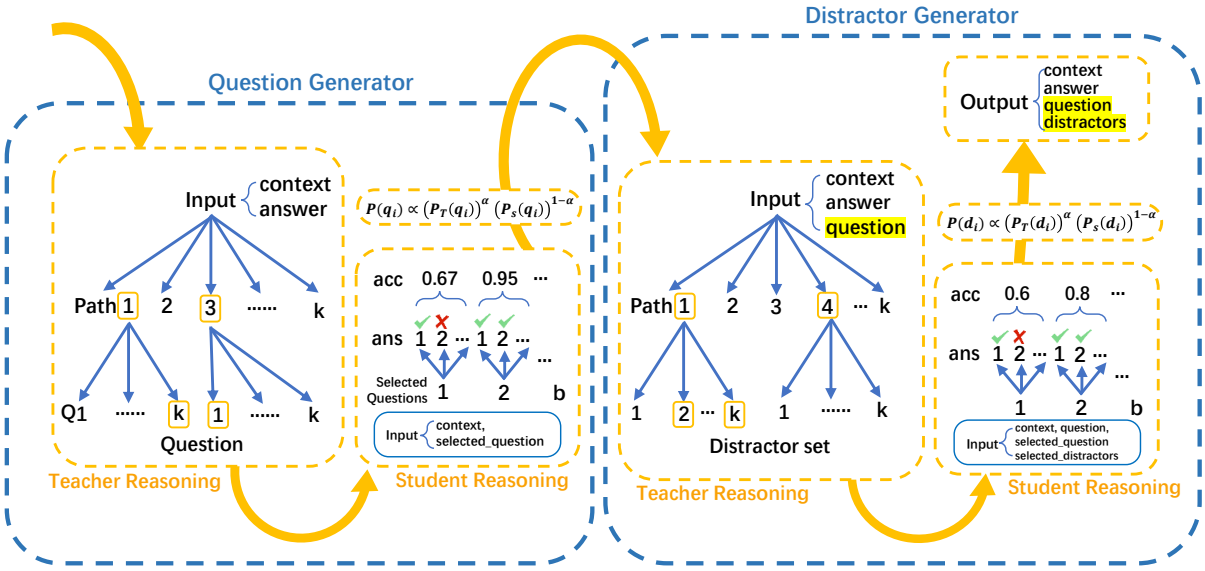

Figure 3: BiFlow comprises a question generator (QG) and a distractor generator (DG), reflecting dual teacher-student reasoning. In QG, teacher reasoning uses PathFinder (combining breadth-first search and CoT) to generate potential questions from context c and answer ans, while student reasoning verifies question clarity using LLM accuracy. The final question is selected via a weighted geometric mean of teacher evaluation and student accuracy. Similarly, in DG, teacher reasoning generates distractor sets, and student reasoning evaluates their plausibility and misleadingness, with the final set chosen by balancing teacher evaluation and inverse student accuracy.

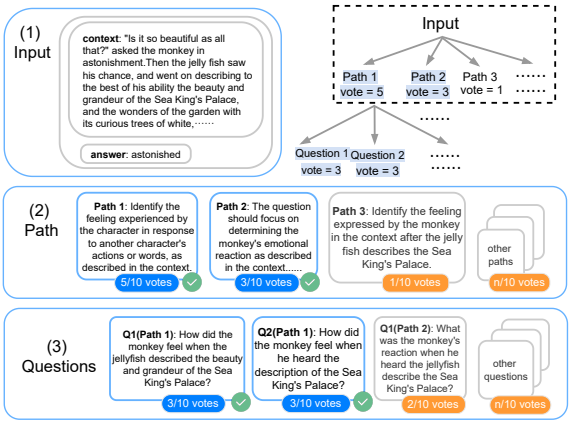

Figure 4: Workflow of PathFinder.

4.1 Question Generator

The goal of the question generator is to generate relevant questions q based on a given context c and the correct answer ans. This process involves teacher reasoning and student reasoning, which work collaboratively to ensure the generated question is both reliable and pedagogically effective.

Teacher Reasoning. Teacher reasoning reasons from the perspective of a teacher and simulates teacher's question-formulation behavior, that is, to generate questions with teaching plan.

QG as an open-ended and exploratory task, a given answer can lead to multiple possible questions. To this end, we propose a mechanism named **PathFinder**, as illustrated in Figure 4. Take c and ans as inputs, PathFinder frames QG task as a process of breadth-first search (with a depth of 2) combined with a CoT reasoning strategy. It enables the generation of intermediate reasoning paths, with subsequent content creation following these paths, ensuring a more structured and exploratory problem-solving process. In PathFinder, each intermediate search result represents a state that includes both a partial solution and the sequence of reasoning steps taken up to that point. It operates as follows:

1. Reasoning Path Generation: With CoT prompting, teacher reasoning component analyzes context c and ans. It then generates k intermediate reasoning paths for question formulation. Each path represents a potential way of generating a question based on context and answer.
2. Path Evaluation: To heuristically evaluate the quality of generated paths, a simple zero-shot evaluation prompt is used to assess the options during the path-finding stage. This process is repeated for v iterations. The likelihood of each path is calculated based on its relative score:

$$P_T(p_i) = \frac{E(p_i)}{\sum_{j=1}^{k} E(p_j)}, \quad (1)$$

where $E(p_i)$ represents the evaluation score of path p_i and k is the total number of generated

paths. The top b paths with the highest scores are selected as $P^s = [p_1, p_2, \ldots, p_b]$.

3. Question Generation: For each selected path, the teacher reasoning component generates k candidate questions. These questions are then evaluated using the same prompt with path evaluation stage. Similarly, the likelihood of each generated question q_i is calculated as:

$$P_T(q_i) = \frac{E(q_i)}{\sum_{j=1}^{k} E(q_j)}, \tag{2}$$

where $E(q_i)$ is the evaluation score of question q_i, and k is the total number of generated questions. Finally, the top b questions with the highest evaluation scores are selected as the final candidate questions $Q^s = [q_1, q_2, \ldots, q_b]$, which are then used as inputs for student reasoning.

Following the process above, teacher reasoning component uses the breadth-first search strategy combined with CoT to explore multiple paths and generate question candidates.

Student Reasoning. Student reasoning starts from a student's perspective to verify the candidate questions Q^s by simulating a problem-solving process. This is opposite to teacher reasoning and ensures that the questions are not only well-formed but also pedagogically effective.

Specifically, for each candidate question q_i, we mask the answer and provide the context c and the question q_i to an LLM in the role of a student. The LLM answers the question for k times, and the accuracy of these responses is calculated as:

$$\text{Accuracy}(q_i) = \frac{\text{Correct}(q_i)}{k}, \tag{3}$$

where $\text{Correct}(q_i)$ is the number of times the LLM's answer matches the correct answer ans. Higher accuracy indicates that the question is clear and precise, signifying higher quality.

Teacher-Student Combination. To fully leverage the dual perspectives of teacher and student, we combine their results to select questions that are more precise, effective and aligned with learning objectives. This integration ensures that the generated questions are not only well-structured from the teacher's perspective, but also validated through the student's problem-solving process. Each candidate question q_i is assigned a weight based on both the teacher's and student's evaluations:

- Teacher Weight: The weight from teacher reasoning is proportional to the evaluation score of the question, $P_T(q_i)$, as questions that receive higher evaluation scores are supposed to be of higher quality.
- Student Weight: The weight from student reasoning is proportional to the accuracy of the LLM's answers:

$$P_S(q_i) \propto \text{Accuracy}(q_i), \tag{4}$$

which based on the assumption that questions with higher accuracy in the LLM's responses are likely to be more effective and precise.

To integrate the two weights, we adopt weighted geometric mean for verification, which balances the influence of teacher and student reasoning while ensuring that both perspectives contribute meaningfully to the final decision.

$$P(q_i) \propto (P_T(q_i))^{\alpha} (P_S(q_i))^{1-\alpha}, \tag{5}$$

where $\alpha \in [0, 1]$ is a weight that balances the influence of teacher and student reasoning.

Finally, the question with the highest combined probability is selected as the final question:

$$q_{\text{final}} = \arg\max_{q_i \in Q^s} P(q_i). \tag{6}$$

The integration method is computationally efficient, as it relies on simple probability calculations without additional training or data collection.

4.2 Distractor Generation

In the distractor generation process, the final question q_{final} along with c and ans are fed as inputs. Similar to the QG process, the DG part also consists of two stages: Teacher Reasoning and Student Reasoning, though the focus is on generating distractors instead of questions.

Teacher Reasoning. Similar to QG, the teacher reasoning component generates multiple distractor sets with PathFinder. Specifically, it first generates k potential paths, evaluates them to select the best b paths, then generates distractor sets $D^s = [d_1, d_2, ...d_b]$ along the best paths. The likelihood of selecting each distractor set d_m is calculated as:

$$P_T(d_i) = \frac{E(d_i)}{\sum_{j=1}^{k} E(d_j)}, \tag{7}$$

where $E(d_i)$ is the evaluation score d_i received.

Student Reasoning. Different from QG, in DG task, the objective of student reasoning is to measure the degree of plausibility and misleadingness of the distractors.

Student reasoning first combines the b most promising candidate distractor sets D^s with the input $(q_{\text{final}}, c, ans)$, forms b complete MCQs, then

simulates a student's problem-solving process, that is, to prompt the LLM to answer while disrupting the order of options. The accuracy of LLM's responses is calculated as:

$$\text{Accuracy}(d_i) = \frac{\text{Correct}(d_i)}{k}, \tag{8}$$

where $\text{Correct}(d_i)$ is the number of times LLM's answer matches the correct answer ans.

Teacher-Student Combination. The weight assignment criteria in DG are different, too. For a candidate distractor set d_i:

- Teacher Weight is defined proportional to the evaluation score of the question $P_T(d_i)$.
- Student Weight is defined inversely proportional to the LLM's accuracy:

$$P_S(d_i) \propto \frac{1}{\text{Accuracy}(d_i)}. \tag{9}$$

  This is based on the assumption that, if the distractors are of higher quality (*e.g.*, more plausible and misleading), the LLM's accuracy in selecting the correct answer should decrease.

For teacher-student combination, as in the question generator, the final score for each candidate distractor set d_i is calculated as a weighted geometric mean of the teacher weight $P_T(d_i)$ and the student weight $P_S(d_i)$:

$$P(d_i) \propto (P_T(d_i))^{\alpha}(P_S(d_i))^{1-\alpha}, \tag{10}$$

where $\alpha \in [0, 1]$ is the weight that balances the influence of dual reasoning. The distractor set with the highest combined score is selected as the final generated distractors:

$$d_{\text{final}} = \arg\max_{d_i \in D^s} P(d_i). \tag{11}$$

5 Experiment

5.1 Experimental Setups

Datasets. We conduct the experiments on the FairytaleMCQ dataset (§3). Detailed description of the dataset is provided in Appendix B.1. Moreover, when analyzing experimental results, we utilize the difficulty labels ex_or_im (the answer is explicit or implicit) in the original FairytaleQA dataset, which are annotated by experts.

Evaluation Metrics. We adopt both automatic evaluation and human metrics to evaluate the experimental results of our framework. For **QG** task, consistent with previous studies, we adopt automatic evaluation metrics including BLEU (Papineni et al., 2002), ROUGE-L (Lin, 2004) to evaluate token-level similarity, and adopt BertScore (Zhang et al., 2019) to evaluate semantic similarity. BertScore is imported since metrics based on n-gram overlap like BLEU and ROUGE-L mainly focus on evaluating the exact match between the generated text and the groundtruth, and do not guarantee text quality (Zhang et al., 2019). For **DG** task, we choose BLEU and ROUGE-L as automatic evaluation metrics.

Baselines. Due to the limited research on few-shot automatic MCQ generation, we selected several general prompting methods in the few-shot setting as baselines. Specifically, we compare BiFlow with Standard Prompt (Brown et al., 2020), Random-CoT (Wei et al., 2022), Manual-CoT (Wei et al., 2022), Self-Consistency (Wang et al., 2023), Complex-CoT (Fu et al., 2022) and Least-to-Most Prompting (Zhou et al., 2023). Among these, Standard Prompt, Random-CoT, and Manual-CoT are tested under 1-shot and 3-shot scenarios, while the others are evaluated in a zero-shot setting. Detailed information about the baseline configurations is provided in Appendix B.2.

5.2 Evaluation on Question Generation

We evaluate the performance of our method in QG task through both automatic evaluation and human evaluation.

Automatic Evaluation. We evaluate our method and various baselines using the test set of FairytaleQA dataset, categorizing the results by question difficulty (explicit and implicit). Table 1 presents the comparison of the automatic evaluation results. Some observations are as follows.

First, compared with other methods, our approach achieve overall better results on the whole dataset, which proves the effectiveness of BiFlow. Second, among questions of different difficulty levels, all the baselines perform better in explicit questions than implicit questions. It confirms that explicit questions contain directly relevant information about the answer, making it easier to infer the corresponding paragraph from the context. Third, our method provides varying degrees of improvement for problems of different difficulty levels, with greater improvement for implicit problems. This proves that BiFlow has greater advantages in implicit and complex reasoning tasks, as it can grasp the hidden relationships in the text. Forth, the performance of CoT steadily improves with more examples in the context learning setting, and com-

Method	BLEU-1 ↑			BLEU-2 ↑			BLEU-4 ↑			ROUGE-L ↑			BertScore ↑		
	All	Explicit	Implicit	All	Explicit	Implicit	All	Explicit	Implicit	All	Explicit	Implicit	All	Explicit	Implicit
Standard Prompt 1 shot	28.74	30.74	22.78	19.04	20.58	14.45	9.18	10.00	6.74	33.75	35.57	28.33	87.51	88.01	86.01
Standard Prompt 3 shot	29.43	30.92	24.98	19.62	20.83	16.01	10.52	11.58	7.36	34.69	36.70	28.70	87.66	88.20	86.05
Random-CoT 1 shot	29.97	31.47	25.50	20.15	21.36	16.54	11.01	12.12	7.70	35.76	37.43	30.78	87.66	88.15	86.19
Random-CoT 3 shot	30.66	32.10	26.36	21.51	22.40	18.86	11.80	13.13	7.84	37.68	38.53	35.15	88.62	89.07	87.28
Manual-CoT 1 shot	30.64	31.74	27.36	21.23	22.74	16.73	11.67	12.96	7.83	38.65	39.16	32.51	88.98	89.44	87.61
Manual-CoT 3 shot	32.32	33.11	29.97	22.06	23.20	18.66	11.83	13.22	7.69	39.27	41.05	33.98	89.42	90.13	87.30
Self-Consistency	34.36	35.84	29.95	22.45	24.31	16.91	12.16	13.58	7.92	44.22	42.28	34.10	89.66	90.15	88.21
Complex CoT	37.23	39.22	31.28	25.45	27.63	18.96	10.72	12.71	4.79	42.12	44.64	34.59	89.44	89.54	89.15
Least-to-Most	35.74	36.88	32.34	23.66	25.07	19.45	12.18	13.60	7.90	40.31	42.54	33.66	89.88	90.24	88.81
BiFlow	**39.38**	**40.95**	**34.70**	**27.54**	**29.16**	**22.73**	**12.27**	**13.72**	**7.94**	**44.94**	**46.80**	**39.38**	**90.45**	**90.63**	**89.91**

Table 1: Automatic evaluation results on the Fairytale dataset for the task of QG.

Method	Read.↑	Rele.↑	Cog.↑	overall.↑
Standard Prompt	3.10	3.43	2.86	3.13
Random-CoT	3.76	3.54	3.25	3.52
Manual-CoT	3.81	3.86	3.23	3.63
Self-Consistency	4.47	4.06	3.97	4.17
Complex-CoT	4.45	4.28	4.30	4.34
Least-to-Most	4.52	4.36	4.24	4.37
BiFlow (- PathFinder)	4.51	4.46	4.24	4.40
BiFlow	**4.55**	**4.62**	**4.36**	**4.51**
Ground truth	4.76	4.91	4.49	4.72

Table 2: Human evaluation results of QG.

Method	B-1↑	B-2↑	B-4↑	R-L↑
Standard Prompt 1 shot	23.85	9.64	2.03	28.12
Standard Prompt 3 shot	24.11	10.47	2.22	28.87
Random-CoT 1 shot	26.70	11.31	2.36	29.06
Random-CoT 3 shot	31.59	13.65	3.92	34.46
Manual-CoT 1 shot	28.96	13.22	3.92	31.94
Manual-CoT 3 shot	33.18	14.80	3.90	36.01
Self-Consistency	33.41	13.65	3.27	34.26
Complex-CoT	33.82	14.27	3.88	36.52
Least-to-Most Prompting	33.73	15.64	**3.96**	36.21
BiFlow	**35.60**	**15.72**	3.94	**38.36**

Table 3: Automatic evaluation results on the Fairytale dataset for the task of DG.

pared with Random-CoT, Manual-CoT achieves better results through manually set high-quality samples, Which is consistent with our expected results. With the experimental results above, BiFlow achieves strong results in complex question generation as well as showcases its pioneering potential.

Human Evaluation. We evaluate the quality of generated questions by randomly sampling 50 examples from the FairytaleQA test set. Three annotators with strong English backgrounds rate the questions from 1 to 5 based on three metrics: (1) Readability, (2) Relevance, and (3) Cognitive Complexity. The average score from all annotators is calculated. As shown in Table 2, while ground truth questions receive the highest scores, our method achieves results closely aligned with them, outperforming other baselines. Detailed scoring guidelines are represented in Appendix B.3.

5.3 Evaluation on Distractor Generation

Next, we evaluate the performance of BiFlow in DG task for multiple-choice questions of narrative comprehension, which also includes both automatic evaluation and human evaluation.

Automatic Evaluation. Table 3 summarizes the automatic evaluation results for DG.

First, BiFlow ranks in the top two of the overall baseline in all metrics, indicating its overall effectiveness in generating diverse and high-quality distractors. Second, We observe that BiFlow achieves similar scores in **BLEU-4** (3.94) compared to Least-to-Most Prompting (3.96). This minor gap may because that Least-to-Most Prompting focuses more on local consistency during its step-by-step reasoning process, while BiFlow prioritizes global semantic coherence. Moreover, studies (Ch and Saha, 2018) indicate that comparing gold standard and system-generated distractors is unreliable, as a gold standard may not include all valid distractors, leading to valid options being misclassified as incorrect. Thus, we further conducted a manual evaluation.

In summary, the high performance consistency BiFlow across varying metrics further demonstrates its universality.

Human Evaluation. For human evaluation, we also adopt other 4 metrics rating from 1 to 5 to evaluate the quality of generated distractors, including (1) Relevance, (2) Distractiveness, (3) Diversity and (4) Overlap. Detailed standard is shown in Appendix B.4. Table 4 represents the results. Overall, the distractors generated by BiFlow perform well in all baselines across different metrics. In addition, since PathFinder with CoT strategy allows inherently analyzes and infers from context, the generated distractors are strongly correlated with the context's topic.

Method	Rele.↑	Dis.↑	Diver.↑	Overl.↓	overall↑
Standard Prompt	4.42	4.38	3.03	2.05	3.70
Random-CoT	4.62	4.52	3.56	1.37	4.08
Manual-CoT	4.68	4.54	3.82	1.34	4.18
Self-Consistency	4.69	4.53	4.49	0.68	4.51
Complex-CoT	4.72	4.55	4.52	0.66	4.56
Least-to-Most	4.76	4.62	4.56	0.82	4.53
BiFlow (- PathFinder)	4.75	4.58	4.56	0.85	4.44
BiFlow	**4.83**	**4.64**	**4.62**	**0.57**	**4.63**
Ground truth	4.95	4.87	4.78	0.09	4.88

Table 4: Human evaluation results of DG.

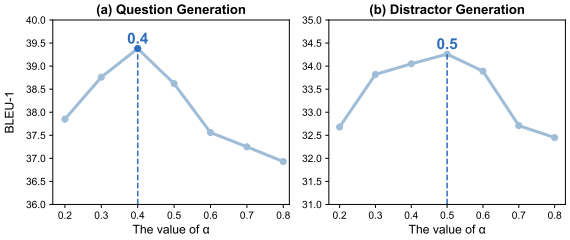

Figure 5: The figure shows the BLEU-1 scores of BiFlow on QG and DG with different numbers of α.

5.4 Influence of the Weight Parameter

We investigate the influence of the weight parameter α on both question generation (QG) and distractor generation (DG). By varying the value of α within the range $[0, 1]$, we evaluate its impact on model performance, as illustrated in Figure 5. From the line chart, we observe that $\alpha = 0.4$ yields the best performance for QG, while $\alpha = 0.5$ is optimal for DG. This demonstrates that the combination of teacher and student reasoning significantly affects the quality of generated questions and distractors. These findings not only validate the effectiveness of our approach but also provide valuable insights for educational research, suggesting that integrating multiple reasoning perspectives can enhance the quality of educational content generation.

5.5 Ablation Study

We further conducted ablation experiments to investigate the effects of prompt, CoT, PathFinder and teacher and student reasoning (TS) part in our method. Table 5 shows the ablation results for question and distractor generation. First, we remove TS, the final selected results are the ones with highest evaluation scores. After removal, QG performance dropped by BLEU-4 8.39%, while DG dropped by BLEU-1 3.71%, showing that TS effectively selects those of higher quality. Second, we remove PathFinder and replace it with CoT with the same prompting process. The significant performance drop indicates that BiFlow helps expand the search space and generate higher-quality text. Finally, we remove CoT and replace it with a simple prompt. The significant performance drop highlights the importance of CoT for large models' reasoning, consistent with prior research.

Method	Question Gen.			Distractor Gen.	
	B-4 ↑	R-L↑	BertScore↑	B-1↑	R-L↑
+BiFlow	12.27	44.94	90.45	33.19	35.28
(1)w/o T-S	11.24	44.00	89.97	31.96	35.04
(2)w/o PathFinder	10.71	38.58	88.04	28.77	32.89
(3)w/o CoT	9.08	35.05	87.26	26.69	29.06

Table 5: Ablation study results on QG and DG.

Story Name: How Molo Stole the Lovely Rose Red
Two years passed, and the youth no longer thought of any danger. ..... Said the prince: "The whole blame rests on Rose-Red. I do not reproach you. Yet since she is now your wife I will let the whole matter rest. But Molo will have to suffer for it!" So the prince ordered a hundred armored soldiers, with bows and swords, to surround the house of the youth, and under all circumstances to take Molo captive. But Molo drew his dagger and flew up the high wall. Thence he looked about him like a hawk. The arrows flew as thick as rain, but not one hit him...

Difficulty: explicit
Gold Question: What will Molo do when the prince wants to capture him?
Answer: draw his dagger and fly up the high wall.
Gold Distractor: run into the house and hide; surrender to the soldiers; swim across the river to escape

Standard Prompt QG: What did Molo do when the armored soldiers surrounded the youth's house?
CoT QG: What did Molo do when a hundred armored soldiers surrounded the youth's house?
Self-Consistency: What did Molo do when the prince's soldiers surrounded the house of the youth?
Complex-CoT: What did Molo do when the prince's soldiers surrounded the youth's house?
Least-to-Most: What did Molo do when the prince ordered his soldiers to surround the house and take him captive?

BiFlow Question: What did Molo do when the prince's soldiers surrounded the house to capture him?
BiFlow Distractor: hide under the floorboards of the house; surrender peacefully to the prince's soldiers; disguise himself as one of the prince's soldiers to escape.

Figure 6: The figure presents a case study from a representative example on the extended Fairytale dataset. We highlight important semantic phrases in the generated questions and distractors, along with their corresponding context, using different colors.

5.6 Case Study

Figure 6 present a case study to visually analyze the experimental effect of BiFlow, where the question and distractors generated by different methods are presented. See Appendix B.5 for detailed analysis.

6 Conclusions

In this paper, we introduced BiFlow, a novel framework for generating high-quality multiple-choice questions (MCQs) and distractors by integrating teacher and student bidirectional reasoning. Our approach leverages PathFinder, a mechanism combining breadth-first search and Chain-of-Thought strategies, to enhance reasoning diversity and quality. Experiments on the FairytaleMCQ dataset demonstrated that BiFlow significantly out-

performed existing methods, particularly in generating complex questions and high-quality distractors for long-text scenarios. The results highlight the effectiveness of our bidirectional reasoning approach in producing pedagogically sound and contextually relevant MCQs.

Limitations

Although BiFlow demonstrates strong performance in MCQ and distractor generation, there are a few limitations to consider. First, the use of PathFinder to explore reasoning paths, while effective, introduces some additional complexity, which might not be necessary for simpler tasks or shorter texts. In these cases, the extra reasoning layers could be seen as overkill, without providing significant improvements. The computational overhead introduced by combining teacher and student reasoning might not be ideal for real-time applications where speed is crucial, although this issue may be less pronounced for smaller-scale use cases.

7 Acknowledgement

We thank the anonymous reviewers for their valuable suggestions and acknowledge the support of our funders. This research was supported by grants from the "Pioneer" and "Leading Goose" R&D Program of Zhejiang (2025C02022), National Natural Science Foundation of China (No. 62307032), and Shanghai Rising-Star Program (23QA1409000).

References

Beng Heng Ang, Sujatha Das Gollapalli, and See Kiong Ng. 2023. Socratic question generation: A novel dataset, models, and evaluation. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 147–165.

Semere Kiros Bitew, Amir Hadifar, Lucas Sterckx, Johannes Deleu, Chris Develder, and Thomas Demeester. 2022. Learning to reuse distractors to support multiple-choice question generation in education. *IEEE Transactions on Learning Technologies*, 17:375–390.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Ioana Buhnila, Georgeta Cislaru, and Amalia Todirascu. 2025. Chain-of-MetaWriting: Linguistic and textual analysis of how small language models write young students texts. In *Proceedings of the First Workshop on Writing Aids at the Crossroads of AI, Cognitive Science and NLP (WRAICOGS 2025)*, pages 1–15, Abu Dhabi, UAE. International Committee on Computational Linguistics.

Sahan Bulathwela, Hamze Muse, and Emine Yilmaz. 2023. Scalable educational question generation with pre-trained language models. In *International Conference on Artificial Intelligence in Education*, pages 327–339. Springer.

Yupeng Cao, Haohang Li, Yangyang Yu, and Shashidhar Reddy Javaji. 2025. Capybara at the financial misinformation detection challenge task: Chain-of-thought enhanced financial misinformation detection. In *Proceedings of the Joint Workshop of the 9th Financial Technology and Natural Language Processing (FinNLP), the 6th Financial Narrative Processing (FNP), and the 1st Workshop on Large Language Models for Finance and Legal (LLMFinLegal)*, pages 321–325, Abu Dhabi, UAE. Association for Computational Linguistics.

Dhawaleswar Rao Ch and Sujan Kumar Saha. 2018. Automatic multiple choice question generation from text: A survey. *IEEE Transactions on Learning Technologies*, 13(1):14–25.

Jingyuan Chen, Jiaxin Shi, Zirun Guo, Yichen Zhu, Zehan Wang, Tao Jin, Zhou Zhao, Fei Wu, Shuicheng Yan, Hanwang Zhang, et al. 2024a. Action imitation in common action space for customized action image synthesis. *Advances in Neural Information Processing Systems*, 37:12195–12218.

Yangyi Chen, Karan Sikka, Michael Cogswell, Heng Ji, and Ajay Divakaran. 2024b. Measuring and improving chain-of-thought reasoning in vision-language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 192–210, Mexico City, Mexico. Association for Computational Linguistics.

Zhenlong Dai, Chang Yao, WenKang Han, Ying Yuan, Zhipeng Gao, and Jingyuan Chen. 2024. Mpcoder: Multi-user personalized code generator with explicit and implicit style representation learning. *arXiv preprint arXiv:2406.17255*.

Zhiang Dong, Jingyuan Chen, and Fei Wu. 2025. Knowledge is power: Harnessing large language models for enhanced cognitive diagnosis. *arXiv preprint arXiv:2502.05556*.

Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to ask: Neural question generation for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Sugyeong Eo, Hyeonseok Moon, Jinsung Kim, Yuna Hur, Jeongwook Kim, Songeun Lee, Changwoo

Chun, Sungsoo Park, and Heuiseok Lim. 2023. Towards diverse and effective question-answer pair generation from children storybooks. *arXiv preprint arXiv:2306.06605*.

Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. 2022. Complexity-based prompting for multi-step reasoning. In *The Eleventh International Conference on Learning Representations*.

Yifan Gao, Piji Li, Irwin King, and Michael R Lyu. 2019. Interconnected question generation with coreference alignment and conversation flow modeling. *arXiv preprint arXiv:1906.06893*.

Le An Ha and Victoria Yaneva. 2018. Automatic distractor suggestion for multiple-choice tests using concept embeddings and information retrieval. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, page 389–398.

Michael Heilman and Noah A Smith. 2010. Good question! statistical ranking for question generation. In *Human language technologies: The 2010 annual conference of the North American Chapter of the Association for Computational Linguistics*, pages 609–617.

Liya Hu, Zhiang Dong, Jingyuan Chen, Guifeng Wang, Zhihua Wang, Zhou Zhao, and Fei Wu. 2023. Ptadisc: a cross-course dataset supporting personalized learning in cold-start scenarios. *Advances in Neural Information Processing Systems*, 36:44976–44996.

Jinfa Huang, Jinsheng Pan, Zhongwei Wan, Hanjia Lyu, and Jiebo Luo. 2025. Evolver: Chain-of-evolution prompting to boost large multimodal models for hateful meme detection. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 7321–7330, Abu Dhabi, UAE. Association for Computational Linguistics.

Samin Jamshidi and Yllias Chali. 2025. GNET-QG: Graph network for multi-hop question generation. In *Proceedings of the Workshop on Generative AI and Knowledge Graphs (GenAIK)*, pages 20–26, Abu Dhabi, UAE. International Committee on Computational Linguistics.

Tomáš Kočiskỳ, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The narrativeqa reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328.

Archana Praveen Kumar, Ashalatha Nayak, Manjula Shenoy K, Chaitanya, and Kaustav Ghosh. 2024. A novel framework for the generation of multiple choice question stems using semantic and machine-learning techniques. *International Journal of Artificial Intelligence in Education*, 34(2):332–375.

Igor Labutov, Sumit Basu, and Lucy Vanderwende. 2015. Deep questions without deep understanding. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*.

Yash Kumar Lal, Nathanael Chambers, Raymond Mooney, and Niranjan Balasubramanian. 2021. Tellmewhy: A dataset for answering why-questions in narratives. *arXiv preprint arXiv:2106.06132*.

Guillaume Le Berre, Christophe Cerisara, Philippe Langlais, and Guy Lapalme. 2022. Unsupervised multiple-choice question generation for out-of-domain q&a fine-tuning. In *60th annual meeting of the association for computational linguistics*, volume 2, pages 732–738. Association for Computational Linguistics.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

David Lindberg, Fred Popowich, JohnC. Nesbit, and PhilipH. Winne. 2013. Generating natural language questions to support learning on-line. *Natural Language Generation,Natural Language Generation*.

Xinpeng Liu, Bing Xu, Muyun Yang, Hailong Cao, Conghui Zhu, Tiejun Zhao, and Wenpeng Lu. 2025. A chain-of-task framework for instruction tuning of LLMs based on Chinese grammatical error correction. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 8623–8639, Abu Dhabi, UAE. Association for Computational Linguistics.

Haohao Luo, Yang Deng, Ying Shen, See-Kiong Ng, and Tat-Seng Chua. 2024. Chain-of-exemplar: enhancing distractor generation for multimodal educational question generation. ACL.

Xiangwei Lv, Guifeng Wang, Jingyuan Chen, Hejian Su, Zhiang Dong, Yumeng Zhu, Beishui Liao, and Fei Wu. 2025. Debiased cognition representation learning for knowledge tracing. *ACM Transactions on Information Systems*.

Potsawee Manakul, Adian Liusie, and Mark JF Gales. 2023. Mqag: Multiple-choice question answering and generation for assessing information consistency in summarization. *arXiv preprint arXiv:2301.12307*.

Sérgio Silva Mucciaccia, Thiago Meireles Paixão, Filipe Wall Mutz, Claudine Santos Badue, Alberto Ferreira de Souza, and Thiago Oliveira-Santos. 2025. Automatic multiple-choice question generation and evaluation systems based on LLM: A study case with university resolutions. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 2246–2260, Abu Dhabi, UAE. Association for Computational Linguistics.

Boyuan Pan, Hao Li, Ziyu Yao, Deng Cai, and Huan Sun. 2019. Reinforced dynamic reasoning for conversational question generation. *arXiv preprint arXiv:1907.12667*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Valentina Pyatkin, Paul Roit, Julian Michael, Reut Tsarfaty, Yoav Goldberg, and Ido Dagan. 2021. Asking it all: Generating contextualized questions for any semantic role. *arXiv preprint arXiv:2109.04832*.

Zhaopeng Qiu, Xian Wu, and Wei Fan. 2020. Automatic distractor generation for multiple choice questions in standard tests. *arXiv preprint arXiv:2011.13100*.

Siyu Ren and Kenny Q Zhu. 2021. Knowledge-driven distractor generation for cloze-style multiple choice questions. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 4339–4347.

Mingxu Tao, Dongyan Zhao, and Yansong Feng. 2025. Chain-of-discussion: A multi-model framework for complex evidence-based question answering. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 11070–11085, Abu Dhabi, UAE. Association for Computational Linguistics.

Xu Wang, Simin Fan, Jessica Houghton, and Lu Wang. 2022. Towards process-oriented, modular, and versatile question generation that meets educational needs. *arXiv preprint arXiv:2205.00355*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Zichen Wu, Xin Jia, Fanyi Qu, and Yunfang Wu. 2022. Enhancing pre-trained models with text structure knowledge for question generation. *arXiv preprint arXiv:2209.04179*.

Zehua Xia, Qi Gou, Bowen Yu, Haiyang Yu, Fei Huang, Yongbin Li, and Cam-Tu Nguyen. 2023. Improving question generation with multi-level content planning. *arXiv preprint arXiv:2310.13512*.

Bingsheng Yao, Dakuo Wang, Tongshuang Wu, Zheng Zhang, Toby Jia-Jun Li, Mo Yu, and Ying Xu. 2021. It is ai's turn to ask humans a question: Question-answer pair generation for children's story books. *arXiv preprint arXiv:2109.03423*.

Han-Cheng Yu, Yu-An Shih, Kin-Man Law, Kai-Yu Hsieh, Yu-Chen Cheng, Hsin-Chih Ho, Zih-An Lin, Wen-Chuan Hsu, and Yao-Chung Fan. 2024. Enhancing distractor generation for multiple-choice questions with retrieval augmented pretraining and knowledge graph integration. *arXiv preprint arXiv:2406.13578*.

Hongwei Zeng, Bifan Wei, Jun Liu, and Weiping Fu. 2023. Synthesize, prompt and transfer: Zero-shot conversational question generation with pre-trained language model. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8989–9010.

Hanchong Zhang, Ruisheng Cao, Hongshen Xu, Lu Chen, and Kai Yu. 2024. CoE-SQL: In-context learning for multi-turn text-to-SQL with chain-of-editions. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6487–6508, Mexico City, Mexico. Association for Computational Linguistics.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, et al. 2023. Instruction tuning for large language models: A survey. *arXiv preprint arXiv:2308.10792*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, et al. 2023. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*.

A Dataset Construction Details

A.1 Prompt of Distractor Generator Agent

```
You are a language education expert. Your task
↪ is to generate three plausible but
↪ incorrect distractors for each narrative
↪ comprehension question. The generated
↪ distractors should meet the following
↪ criteria:

**Criteria**:
(1) Deceptiveness: The distractors should be
↪ syntactically and semantically similar to
↪ the correct answer, making them plausible
↪ yet incorrect.
(2) Consistency: The distractors must be
↪ contextually relevant and consistent with
↪ the answer type (ans\_type) to ensure
↪ alignment with the question's structure
↪ and type.

**Steps to Follow**:
(1) Analyze the provided context (c), question
↪ (q), correct answer (ans), and its type
↪ (ans\_type).
```

```
(2) Identify key elements in the context that
↪  are related to the correct answer but can
↪  be modified or misinterpreted to create
↪  misleading options.
(3) Generate distractors that are:
   a) Of the same category and grammatical
   ↪  structure as the correct answer.
   b) Plausible within the context but
   ↪  factually incorrect.
   c) Challenging enough to test the user's
   ↪  understanding of the material.

**Examples**:
**Context**: Thus they went along, the jelly
↪  fish skimming through the waves with the
↪  monkey sitting on his back. When they were
↪  about half-way, the jelly fish, who knew
↪  very little of anatomy, began to wonder if
↪  the monkey had his liver with him or not!
↪  "Mr. Monkey, tell me, have you such a
↪  thing as a liver with you?" The monkey was
↪  very surprised at this queer question, and
↪  asked what the jelly fish wanted with a
↪  liver. "That is the most important thing
↪  of all," said the stupid jelly fish, "so
↪  as soon as I recollected it, I asked you
↪  if you had yours with you?" "Why is my
↪  liver so important to you?" asked the
↪  monkey. "Oh! you will learn the reason
↪  later," said the jelly fish.
**Question**: How did the monkey feel when he
↪  was asked about his liver?
**Answer**: surprised
......

**Response**: Distractors: ["confused",
↪  "curious", "amused"]

Here is the target question:

**Context**: {context}
**Question**: {question}
**Answer**: {answer}

Your output:
{distractors: ["xxx", "xxx", ...]}
```

A.2 Prompt of Result Checker Agent

```
You are a Result Checker. Your task is to
↪  evaluate the quality of distractors
↪  generated by the Distractor Generator
↪  Agent and to ensure that the distractors
↪  meet the following quality standards:

**Quality Standards**:
(1). Syntactic Correctness: The distractor
↪  must be grammatically sound and free from
↪  errors.
(2). Consistency with ans\_type: The
↪  distractor must align with the expected
↪  answer type (ans\_type), ensuring it is
↪  structurally similar to the answer and
↪  relevant to the type of question.
(3). Plausibility as an Incorrect Option: The
↪  distractor must be a reasonable, albeit
↪  incorrect, alternative to the correct
↪  answer, making it deceptive yet
↪  contextually appropriate.

**Steps to Follow**:
(1). Review each distractor generated by the
↪  Distractor Generator Agent.
(2). Evaluate the distractor based on the
↪  three criteria above.
(3). If any distractor fails any of the
↪  criteria, provide feedback to the
↪  Distractor Generator Agent for revision.
(4). If all the distractors pass all criteria,
↪  approve them for inclusion in the dataset.

**Examples**:
**Context**: Thus they went along, the jelly
↪  fish skimming through the waves with the
↪  monkey sitting on his back. When they were
↪  about half-way...
**Problem**: How did the monkey feel when he
↪  was asked about his liver?
**Answer**: surprised
**Answer type**: feeling
**Generated distractors**: ["Curious",
↪  "Confused", "Frustrated"]
**Response**: Distractors are unreasonable.
↪  "Frustrated" is grammatically correct and
↪  aligns with the expected answer type but
↪  does not fit the context of the monkey's
↪  reaction to an unusual question.
......

Here is the target question:

#Context: {context}
#Problem: {problem}
#Answer: {answer}
#Answer Type: {answer type}
#Generated Distractors: [{distractor 1},
↪  {distractor 2}, {distractor 3}]

Your output:

#The distractors are
↪  {reasonable/unreasonable}.
#Reason(If inappropriate): {reason}
```

A.3 Criteria Details

The evaluators assess and modify the automatically generated distractors based on the following criteria:

- **Linguistic accuracy**: The distractor should be grammatically correct and uses appropriate words. Evaluators should correct inappropriate wording.
- **Relevance**: The distractor must be appropriate in the context of the question. Evaluators should regenerate distractors that do not align with the article.
- **Reasonableness**: The distractor must be a plausible but incorrect alternative to the correct answer.

Evaluators should regenerate interference items with poor interference

B Experiment Details

B.1 Dataset FairytaleQA

FairytaleQA is a high-quality dataset focusing on children's storybook learning and assessment, which consists of 10,580 explicit and implicit questions derived from 278 children-friendly stories, covering seven types of narrative elements or relations. The training, validation and test sets contain 8,548, 1,025, and 1,007 QA pairs. By expanding the dataset (Section 3), we have added a set of distractor answers D' for each QA pair in the test set. Therefore, the QA dataset can be converted into a multiple-choice question (MCQs) dataset for deeper experiments.

B.2 Detailed Baseline Configurations

Selected baselines and their detailed configurations are shown below:

- **Standard Prompt** (Brown et al., 2020) is a simple few-shot prompting method with in-context exemplars. we randomly select several examples for few-shot settings.
- **Random-CoT** (Wei et al., 2022) is a naive baseline with up to three examples randomly selected from the training set. We adhere to the design principles outlined in their study to design our CoT template, while each of the selected examples is formatted the same as our preprocessed data in the FairytaleQA dataset.
- **Manual-CoT** (Wei et al., 2022) is similar to Random-CoT unless the examples are manually created to ensure high quality and diversity of examples. We try different variants with minor modifications of our CoT template and choose the one with the best performance in experiments.
- **Self-Consistency** (Wang et al., 2023) is a method that involves generating multiple reasoning paths through CoT prompting with sampled decoding, then aggregating the results by selecting the most consistent answer from the final set. We use the same prompt of Manual-CoT in Self-consistency.
- **Complex-CoT** (Fu et al., 2022) employs a strategy where multiple reasoning chains and corresponding answers are generated for the same test question, ranked in descending order by the length of the reasoning chain, and the top-ranked chains are retained to determine the final prediction through a voting mechanism. We apply it to QG and DG tasks and perform a similar process.
- **Least-to-Most Prompting** (Zhou et al., 2023) decompose the problem into a series of simpler subproblems and then solve them in sequence. To apply it to DQG tasks, we decompose the task into three steps corresponding to our CoT template, then sequentially prompt LLM to complete these steps. Each step is generated based on the previous step.

B.3 Human Evaluation Criteria for QG

We conduct human evaluation to evaluate the quality of the generated questions. We randomly sampling 50 examples from the test set of FairytaleQA, then employ three annotators with good English background to rate them from 1 to 5 based on 3 metrics, where 1 denotes poor and 5 denotes perfect:

- **Readability** measures the generated question easy to read and understand, suitable for students in the target grade level;
- **Relevance** measures whether the generated problem is closely related to the given reading comprehension context;
- **Cognitive Complexity** measures whether the generated questions are suitable for the cognitive level of the target grade students and whether they can stimulate students' thinking, encouraging them to reason, analyze, and reflect on the given context.

B.4 Human Evaluation Criteria for DG

we adopt other 4 metrics rating from 1 to 5 to evaluate the quality of generated distractors, including:

- **Relevance** measures whether the distractor is related to the background and topic of the article.
- **Distractiveness** measures whether the distractor can effectively confuse students and cause them to hesitate between the correct answer and the interference item.
- **Diversity** measures whether there are differences between distractors and whether they can cover different types or angles of errors
- **Overlap** measures whether there is complete or partial overlap between the interference term and the correct answer (such as semantic repetition, keyword repetition, etc.).

B.5 Case Study Analysis

As shown in Figure 6, the question generated by our method aligns more closely with the gold question as BiFlow explores multiple questioning plans

and selects higher-quality text via teacher-student bidirectional reasoning. Besides, distractors generates by our method align closely with the contextual theme. When focusing solely on the correct answer and the interference, their similar grammatical structures create stronger distractions for students therefore stimulate thinking.