SciTaT: A Question Answering Benchmark for Scientific Tables and Text Covering Diverse Reasoning Types

Anonymous ACL submission

Abstract

Scientific question answering (SQA) is an important task aimed at answering questions based on papers. However, current SQA datasets have limited reasoning types and neglect the relevance between tables and text, creating a significant gap with real scenarios. To address these challenges, we propose a QA benchmark for scientific tables and text with diverse reasoning types (SciTaT). To cover more reasoning types, we summarize various reasoning types from real-world questions. To reason on both tables and text, we require the questions to incorporate tables and text as much as possible. Based on SciTaT, we propose a baseline (CaR), which combines various reasoning methods to address different reasoning types and process tables and text at the same time. CaR brings average improvements of 4.1% over other baselines on SciTaT, validating its effectiveness. Error analysis reveals the challenges of SciTaT, such as complex numerical calculations and domain knowledge.

1 Introduction

Scientific Question Answering (SQA) plays a crucial role in addressing research questions based on scientific papers (Tsatsaronis et al., 2015; Lee et al., 2023). Advancing SQA development can significantly accelerate knowledge acquisition (Taylor et al., 2022; AI4Science and Quantum, 2023). The dense technical terms and heterogeneous data representations in papers present challenges for the SQA task (Sun et al., 2024; Pramanick et al., 2024).

To evaluate and enhance the model capabilities in SQA, numerous datasets are proposed (Pampari et al., 2018; Jin et al., 2019; Pappas et al., 2020). However, existing datasets exhibit the following limitations, as shown in Table 1. Firstly, the **reasoning types are relatively narrow**, failing to capture the complexity of real scenarios, such as data analysis, which is frequently encountered in actual queries (Moosavi et al., 2021). Secondly,

Dataset	Reasoning Type L	N	D	T	Evidence Text	Table	TaT
BioRead	◑				✓		
QASA	◑	◔	◕		✓		
SciGen			◕			✓	
SciTab	◑	◕				✓	
SPIQA	◑	◕			✓	✓	
SciTaT	●	●	●	●	✓	✓	✓

Table 1: Comparison of SciTaT to recent SQA datasets, introduced in Appendix A.1. TaT denotes Table and Text. **L**, **N**, **D**, and **T** denote the reasoning type of Look Up, Numerical Reasoning, Data Analysis, and Tabulation, with examples in Figure 1. Pie charts represent the proportion of subtypes compared with SciTaT.

prior works **focus only on split text and tables**, overlooking the relevance between tables and text, thereby limiting their applicability (Wang et al., 2022). To address the limitations, in this paper: *(i)* We introduce a new SQA benchmark, covering diverse real-scenery reasoning types and considering tables and text simultaneously. *(ii)* To enhance the performance on the benchmark, we propose a baseline, which can handle multiple reasoning types and process tables and text simultaneously.

Firstly, we propose a QA benchmark for scientific tables and text (SciTaT), which are collected from papers in arXiv.org. **To incorporate more reasoning types**, we summarize various reasoning types from the real questions raised by researchers (see Figure 1). **To ensure the questions require reasoning on both tables and text**, we require questions to involve tables and text as much as possible. Overall, SciTaT contains 953 questions derived from 871 papers. Data analysis reveals that SciTaT encompasses 4 reasoning types and 13 subtypes, covering the types summarized from the real questions in SparkRA (Wu et al., 2024a) and previous works (Lu et al., 2023; Wu et al., 2024b). SciTaT not only requires the model to look up information and numerical reasoning but also re-

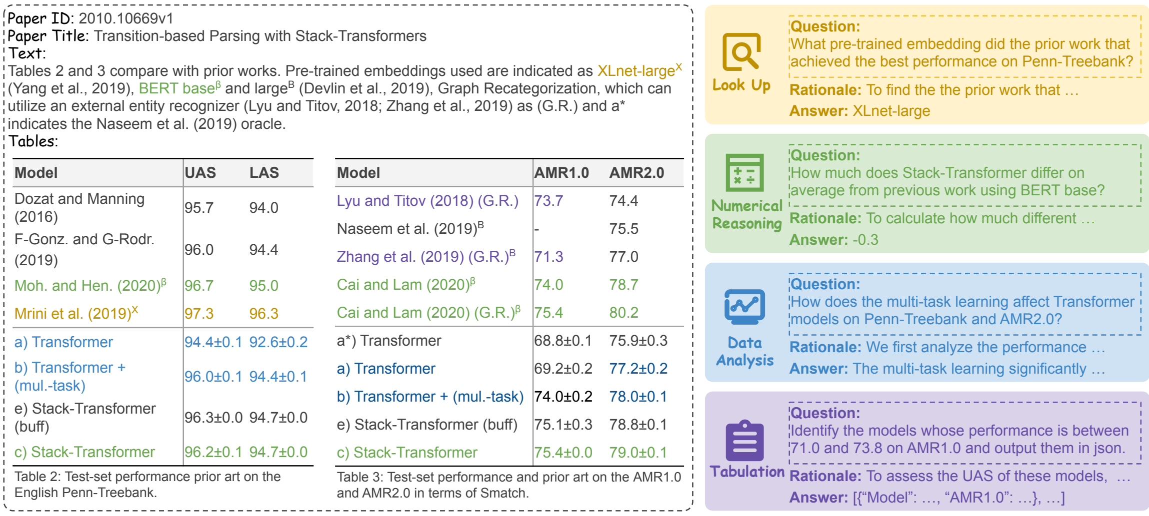


Figure 1: Illustrations of the reasoning types in SCITAT. The tables and text (left) show color-coded spans for question context. The questions (right) are examples of 4 reasoning types, with their rationales and answers.

quires complex data analysis and tabulation, effectively meeting the needs of real-world researchers.

Considering the challenges of SCITAT, we propose a baseline to process scientific data by integrating reasoning methods (CAR). **To handle multiple reasoning types**, CAR includes two modules: Calculator and Reasoner. **To process tables and text**, Calculator extracts and calculates numerical information from tables and text, which is then provided to Reasoner for further reasoning.

We construct a series of baselines on SCITAT. Experimental results reveal that CAR outperforms other baselines with 4.1% on average, proving the effectiveness of the combination of Calculator and Reasoner. However, the Exact Match of CAR using `gpt-4o` is still below 50%, which indicates that SCITAT serves as a challenging benchmark. Error analysis reveals the main challenges of SCITAT, such as context grounding, complex numerical calculation, and the need for domain knowledge.

Our contributions are as follows:

1. To the best of our knowledge, we develop SCITAT, the first QA benchmark for scientific tables and text, covering diverse reasoning types based on real scenarios.

2. We propose CAR, a baseline to solve various reasoning types and process tables and text by integrating reasoning methods.

3. We conduct a series of experiments, providing results and error analysis to highlight the challenges of SCITAT, thereby guiding the direction for future improvements.

2 SCITAT Dataset

The input for our task consists of scientific tables, text, and a question, and the output is the answer. Moreover, we annotate the rationale of each question. For brevity, we refer to each question, its corresponding rationale, and answer, as an instance. We begin by describing the construction process of SCITAT. We employ a framework combining automatic generation with manual annotation to enhance both the quality and efficiency of the annotation process, as illustrated in Figure 2.

2.1 Paper Preparation

Source Data Collection We select papers from the "Artificial Intelligence", "Computation and Language", and "Machine Learning" subfields of "Computer Science" following previous datasets (Lee et al., 2023; Moosavi et al., 2021; Lu et al., 2023). We collect LaTeX code from papers published between January 2020 and July 2023 on arxiv.org[1], using a heuristic method to extract all the tables with their corresponding captions and labels, and text in each paper.

Tables and Text Selection To ensure the inclusion of both tables and text, we filter out papers without tables. Additionally, to guarantee the relevance of the tables and text in the context, the context we provide when annotating the question is a paragraph that mentions tables and the tables mentioned. Specifically, we randomly select at least one paragraph from the paper that mentions tables and the mentioned tables as the context.

[1] https://info.arxiv.org/help/bulk_data/index.html

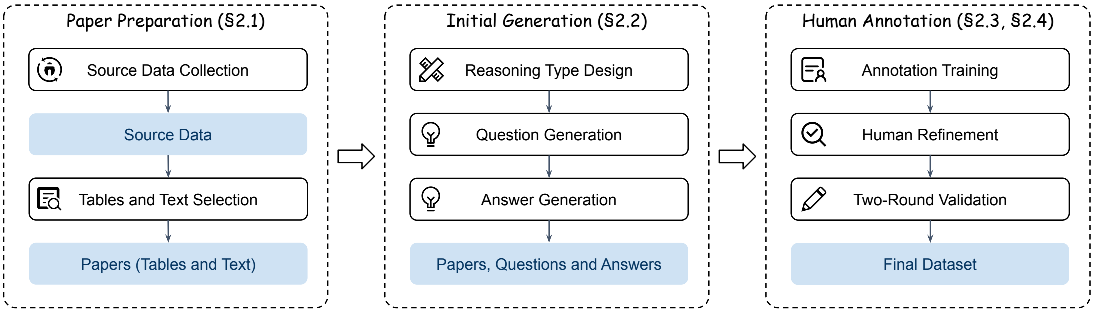


Figure 2: The overview of our annotation process. The blue boxes represent the data and the white boxes with solid lines represent the annotation procedures.

The requirements for generating questions

1. The question must meet the reasoning type.
2. The question is best answered by referring to both the tables and the text simultaneously.
3. The question should be with more reasoning and calculation.

Figure 3: The requirements for generating questions.

2.2 Initial Generation

Reasoning Type Design To observe the reasoning types that researchers might query, we select SParkRA (Wu et al., 2024a), a platform specifically designed to provide QA services for researchers in the context of scientific papers. We hypothesize that the reasoning types observed in these questions are comparable to those found in real-world inquiries. We randomly select 650 questions and categorize their reasoning types. To account for potentially unobserved reasoning types, we also incorporate reasoning types from previous datasets (Lu et al., 2023; Wu et al., 2024b). Finally, we summarize 4 reasoning types and 13 subtypes for SciTaT, as shown in Table 2.

Question and Answer Generation We assign a reasoning type to each context, including a paragraph and mentioned tables. Manually annotating scientific questions and answers is time-consuming and prone to introducing annotation artifacts since it requires substantial domain expertise and a deep understanding of the paper (Bender and Friedman, 2018; Pramanick et al., 2024). To address these challenges, we leverage the extensive knowledge and powerful instruction-following capabilities of `gpt-4o` (OpenAI et al., 2024) following previous works (Zhang et al., 2023, 2024; Yu et al., 2025). We guide the LLM to generate questions aligned with the reasoning type based on the context using three-shot prompt, with requirements outlined in Figure 3. We guide the LLM to generate the rationale and answer for each question according to the context with a three-shot prompt. Detailed prompts are provided in Appendix B.1.

2.3 Human Refinement

Since LLMs cannot guarantee the reasonableness of the questions or the correctness of the answers, we employ manual checks and refinement. (*i*) For the context, annotators are tasked with verifying that the extracted tables and text are consistent with the original paper and removing any incorrect ones. (*ii*) For the questions, annotators should refine them following the guidelines in Figure 3. (*iii*) For the rationales and answers, annotators are required to verify their correctness and correct any errors. Due to the diverse reasoning types in SciTaT, our answers include both *short-form* and *free-form* types. Annotators are instructed to extract one or more tokens for short-form answers and use complete sentences for free-form answers. (*iv*) For the answer source, annotators are prompted to select the source of the answer, which may include *Text*, *Table*, or *Table and Text*, and identify the relevant tables. Annotators are compensated $1 per instance.

2.4 Quality Control

To ensure the quality of SciTaT, we implement rigorous quality control strategies.

Competent Annotators The annotators we employ are all graduate students majoring in artificial intelligence. Initially, they undergo **annotation training** sessions to learn the task and the annotation interface (see Appendix C.1) and are required to annotate 20 questions. We retain those with

Reasoning Type	Subtypes	Description	%
Look Up	Table Look Up	Search for specific tables	2.7
	Span Look Up	Search for spans in tables or paragraphs	2.0
Numerical Reasoning	Arithmetic Calculation	Numerical calculations	11.1
	Comparison	Comparison of values	8.2
	Aggregation	Combines multiple data points into a single metric	3.9
	Ranking	Arranges items in a specific order	7.0
	Counting	Counting occurrences	9.2
	Domain Knowledge Calculation	Calculations requiring domain knowledge	6.5
Data Analysis	Descriptive Analysis	Summarize or interpret to spot patterns and trends	23.3
	Anomaly Detection	Detect deviations and their causes	7.0
	Causal Analysis	Investigate cause-and-effect relationships	10.6
Tabulation	-	Standardizing the formats of tables/subtables	8.4

Table 2: The reasoning types, the description of their subtypes, and their proportion in SciTaT. Look Up, Numerical Reasoning, Data Analysis, and Tabulation account for 4.7%, 46.0%, 40.9%, and 8.4% respectively.

Statistics	Long-context	Short-context
Questions	953	953
Papers	871	871
Avg. Tables	5.2	1.1
Avg. Cells	60.8	56.6
Avg. Paragraphs	80.2	1.0
Avg. |Paragraph|	83.4	113.0

Table 3: The statistics of SciTaT. Avg. Tables and Avg. Cells indicate the average number of tables and cells per table. Avg. Paragraphs and Avg. |Paragraph| indicate the average number of paragraphs and the average length of each paragraph.

Statistics	Table	Text	TaT	Total
Short-form answers	234	13	93	340
Free-form answers	308	67	238	613
Total	542	80	331	953

Table 4: Question distribution over different answers and sources in SciTaT.

Exact Match $\geq 95\%$ and provide constructive feedback on their mistakes. Detailed annotation information is provided in Appendix C.2.

Two-round Validation After the instances are submitted by the annotator, a two-round validation is implemented, consisting of manual verification and revision, following the previous work (Zhu et al., 2021). (*i*) In the first round, a verifier examines each instance to ensure that the annotations adhere to the guidelines. If errors are found, the verifier communicates with the annotator and requests the corresponding corrections. (*ii*) In the second round, a different annotator reviews the instances again. Any identified errors are discussed with the verifier annotator and revised as needed.

2.5 Data Analysis

Basic Statistics To better evaluate the reasoning ability across different context lengths, we construct two settings: *long-context* and *short-context*. In the long-context setting, the model should answer questions based on the whole paper. In the short-context setting, the model is required to answer questions based on a paragraph and the tables referenced in that paragraph. We present the statistics of SciTaT in the two settings in Table 3. We also show the question distribution over different answers and sources in Table 4. Notably, over $1/3$ of the questions in SciTaT require reasoning that involves both tables and text simultaneously.

Reasoning Types We analyze the distribution of reasoning types in SciTaT, as shown in Table 2. It can be found that SciTaT has a variety of reasoning types evenly distributed. Among these types, Data Analysis and Tabulation are identified as common patterns based on observations of real queries and are rarely represented in existing datasets.

3 CaR

CaR is designed to address the questions on the context of scientific tables and text. Given that SciTaT combines diverse reasoning types, CaR is composed into two modules: Calculator and Reasoner, as illustrated in Figure 4, which focus on different reasoning types. To process tables and text simultaneously, Calculator extracts and computes the numerical information from the context and Reasoner derives the final answer based on the calculated information. The prompts we use are presented in Appendix B.2.

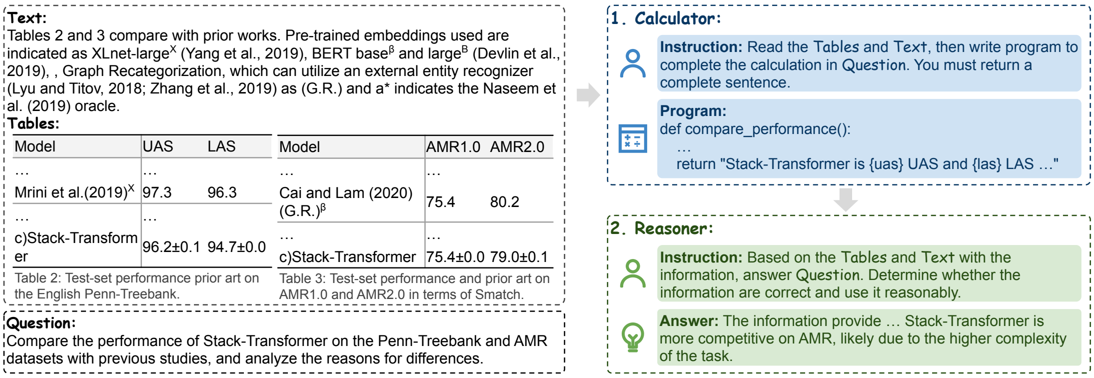


Figure 4: The overview of CAR, which consists of two modules: (*i*) Calculator generates code to compute the numerical information required for solving the question. (*ii*) Reasoner continues the reasoning process based on the information provided by the Calculator to answer the question.

3.1 Calculator

The input to the Calculator consists of a question and the scientific context (including tables and text), and the output provides the numerical information necessary to answer the question. Specifically, we prompt the LLM to generate a program function based on the context to answer the question. Unlike other Program-of-Thought (PoT) methods (Gao et al., 2023; Chen et al., 2023) that require the program to return the answer directly, the function is designed to return a sentence explicitly describing the numerical information, as illustrated in Figure 4. Once the function is obtained, it is executed to extract the numerical information.

3.2 Reasoner

The Reasoner takes as input a question, the scientific context, and the numerical information obtained from the Calculator to produce the final answer. Specifically, we utilize a CoT prompt (Wei et al., 2022) to guide the LLM through a step-by-step reasoning process based on the context and information, leading to the final answer. However, since the information may not always be accurate or helpful, we further prompt the LLM to engage in reflection, evaluating the correctness and relevance of the extracted information during reasoning.

4 Experiments

4.1 Settings

Metrics Due to the significant difference in token counts between free-form answers and short-form answers (see Table 12), we evaluate the two types of answers separately. For short-form answers, we use Exact Match (EM) to assess correctness, while for free-form answers, we use F1 and BERTScore F1 (BERTScore) (Zhang* et al., 2020), following previous studies (Zhu et al., 2021; Moosavi et al., 2021). EM measures the proportion of the predicted result that exactly matches the gold answer. F1 calculates the overlap between predicted and gold answers based on their bag-of-words representation. BERTScore evaluates the similarity between predicted and gold answers by calculating the cosine similarity of their embeddings.

Models We employ the open-source LLM Llama3.1-Instruct (Dubey et al., 2024), Qwen2.5-Coder-Instruct (Hui et al., 2024) and the closed-source LLM `gpt-4o` (OpenAI et al., 2024) to evaluate SCITAT. Llama3.1 and Qwen2.5-Coder are among the top-performing open-source general models and code models, while `gpt-4o` is one of the leading closed-source models.

Baselines We compare CAR with the following baselines, with prompts in Appendix B.2.

- Direct QA (Pramanick et al., 2024) prompts the LLM to directly answer the questions.
- CoT (Wei et al., 2022) prompts the LLM to perform the step-by-step reasoning process and then get the final answer.
- PoT (Gao et al., 2023; Chen et al., 2023) prompts the LLM to generate a program that can be executed to obtain the answer.
- Three-Agent (Fatemi and Hu, 2024) is currently the state-of-the-art method for reasoning on tables and text without considering fine-tuning models, which consists of three agents: an analyst agent that looks up relevant information and performs calculations, and

Model	Scale	Method	Long-context			Short-context		
			EM	F1	BERTScore	EM	F1	BERTScore
Llama3.1	8B	Direct QA	0.0	30.6	66.4	0.0	30.7	66.5
		CoT	13.0	29.5	65.6	20.6	41.4	71.6
		PoT	4.4	21.7	54.0	17.1	21.5	49.9
		Three-Agent	12.6	34.3	68.5	22.1	36.5	67.6
		CaR	**24.8**	**37.5**	**69.7**	**24.2**	**44.3**	**73.2**
	70B	Direct QA	0.0	31.6	67.5	0.0	33.6	68.7
		CoT	30.3	36.8	69.9	32.1	44.1	73.1
		PoT	5.3	28.8	61.7	36.8	35.6	64.0
		Three-Agent	30.3	40.0	71.3	40.3	40.0	71.1
		CaR	**35.9**	**41.7**	**71.8**	**40.7**	**46.2**	**74.4**
Qwen2.5-Coder	7B	Direct QA	0.0	27.7	64.8	1.8	35.2	68.9
		CoT	15.3	36.6	69.3	21.5	44.2	73.2
		PoT	4.4	6.9	35.4	19.7	14.3	42.2
		Three-Agent	10.6	35.9	68.8	20.3	41.8	72.5
		CaR	**16.8**	**41.3**	**71.5**	**25.9**	**44.5**	**73.7**
`gpt-4o`	-	Direct QA	0.0	29.8	67.3	0.0	39.6	71.5
		CoT	31.3	41.3	72.7	32.2	46.8	75.5
		PoT	5.0	15.0	44.4	28.3	31.2	59.8
		Three-Agent	34.7	40.3	72.3	27.1	44.9	74.6
		CaR	**37.5**	**41.8**	**73.1**	**43.7**	**47.1**	**75.7**

Table 5: Performance comparison of different models and methods. The best results of each model under each setting are annotated in **bold**.

two critic agents that provide feedbacks on extraction and calculation and correct errors.

Given the long-context setting, we adopt zero-shot prompts in main experiments to prevent exceeding the context limit, with the few-shot results in the short-context setting in Appendix E.1.

4.2 Main Experiments

The results of comparing CaR with other baselines on SciTaT are shown in Table 5. The results reveal that: (*i*) CaR significantly outperforms other baselines across different models and settings, achieving an average improvement of 4.1% on all metrics, highlighting its effectiveness. (*ii*) Despite improvement, CaR demonstrates suboptimal performance, as EM and F1 remain below 50.0, and while BERTScore is relatively high (Moosavi et al., 2021; Zhao et al., 2024a), it remains under 80.0, reflecting the challenge of SciTaT. We also observe:

Baselines (*i*) CaR outperforms Three-Agent, demonstrating the diversity of reasoning types in SciTaT. It encompasses not only Look Up and Numerical Reasoning, and complex computations in SciTaT cannot be solved solely by the models themselves. Three-Agent is more pronounced on larger-scale models, as their stronger critic capabilities allow for more precise information extraction and calculation (Pan et al., 2024; Tian et al., 2024; Lin et al., 2024). (*ii*) Among other baselines, CoT achieves higher performance, while Direct QA exhibits lower EM, and PoT shows lower F1 and BERTScore. Considering that diverse reasoning types in SciTaT, CoT is relatively better at handling these types of questions (Wei et al., 2022; Wu et al., 2024b; Pramanick et al., 2024). Direct QA, due to its lack of reasoning, is prone to computational errors and longer answers for short-form answers, resulting in an EM score of zero (Snell et al., 2024). Since the program typically returns shorter answers (see Appendix E.2), the PoT method is less effective at answering free-form questions.

Context Settings CaR demonstrates a more significant improvement in the long-context setting than the short-context setting. Due to the dense knowledge presented in the paper, directly answering questions based on the entire paper may confuse the model, preventing it from focusing on the relevant tables and text (Lee et al., 2023; Pramanick et al., 2024). In contrast, CaR uses the Calculator to extract and compute useful numerical information from the paper, effectively guiding the Reasoner and avoiding the need to search for answers directly within the whole paper.

Answer Types CaR shows more significant improvements in short-form answers than free-form answers. For short-form answers, the Reasoner

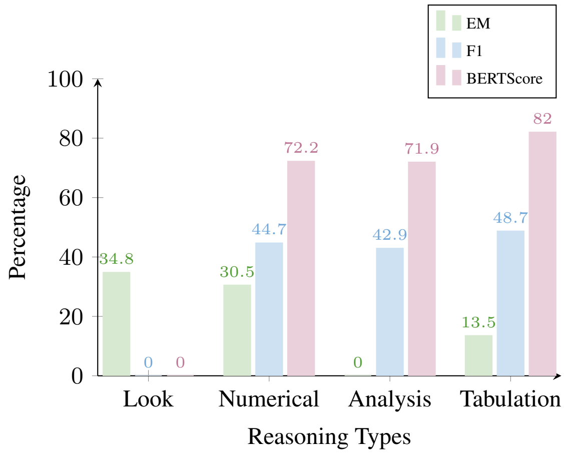


Figure 5: The average performance of CAR across four models on four reasoning types. Look denotes Look Up, Numerical denotes Numerical Reasoning, and Analysis denotes Data Analysis.

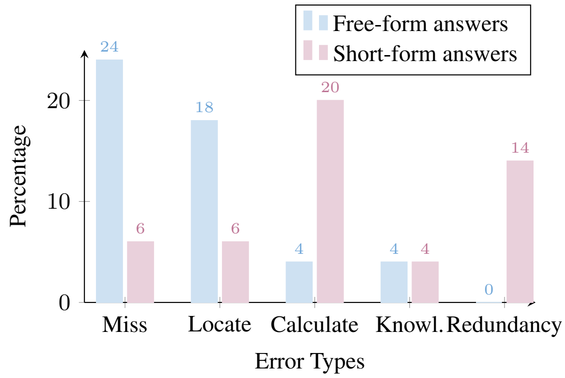


Figure 6: The distribution of error types of CAR. Knowl. denotes Knowledge.

Scale	Method	EM	F1	BERTScore
8B	CAR	**24.2**	**44.3**	**73.2**
	w/o Calculator	20.6	41.4	71.6
	w/o Reasoner	0.3	30.8	62.8
70B	CAR	**40.7**	**46.2**	**74.4**
	w/o Calculator	32.1	44.1	73.1
	w/o Reasoner	0.0	42.0	70.4

Table 6: The ablation results of CAR using Llama3.1 in the short-context setting, compared with removing Calculator (*w/o Calculator*) and removing Reasoner (*w/o Reasoner*).

typically only needs to verify the correctness of the result of the Calculator and extract the answer. In contrast, for free-form answers, the Reasoner often needs to perform additional analysis based on the numerical information provided by the Calculator.

Reasoning Types We present the average performance of four models across reasoning types in Figure 5. Specifically, the F1 and BERTScore for the type of Look Up are 0 as all corresponding answers are short-form. The EM for Data Analysis is 0, as all the answers to this reasoning type are free-form. We can observe that: (*i*) The models perform worst on Data Analysis, which requires more comprehensive capabilities, such as numerical computation, logical reasoning, and summarization (Wu et al., 2024b). (*ii*) The F1 and BERTScore on Tabulation are the highest, but the EM is the lowest, indicating the difficulty of this reasoning type. While the predicted result may be close to the gold answer, achieving an exact match remains challenging. This highlights the need for more effective evaluation metrics. (*iii*) There is still significant room for improvement on all the types, underscoring the challenges of SCITAT.

4.3 Ablation Experiments

To demonstrate the effectiveness of CAR, we perform an ablation study by removing each module, with results presented in Table 6. Specifically, when removing the Calculator, it is the same as the CoT baseline. The significant performance drop confirms the validity of CAR. The results suggest that relying on a single reasoning method is insufficient to derive accurate answers due to the diverse reasoning types in SCITAT. Especially, the EM of removing Reasoner is low since we prompt the program to return the entire numerical information instead of the simple answer.

4.4 Error Analysis

To present the challenges of SCITAT, we analyze the error instances of CAR using Llama3.1-70B. Specifically, we randomly select 25 error instances with BERTScore below 60 from the results corresponding to free-form answers and another 25 instances with EM of 0 from the results corresponding to short-form answers. We manually categorize the error types, as illustrated in Figure 6, with examples of error types in Appendix D. It can be observed that the distribution of error types for free-form answers and short-form answers differs significantly. We proceed with a detailed analysis.

(*i*) **Miss** refers to the omission of part of the answer, such as when only some sub-questions are addressed, or when data analysis is limited to summarizing phenomena without providing conclusions or insights. (*ii*) **Locate** refers to locating incorrect relevant context according to the question. (*iii*) **Calculate** denotes errors in applying formulas, programming mistakes, or computational inaccuracies. (*iv*) **Knowledge** refers to errors in responses due to the lack of domain-specific knowledge. (*v*) **Redundancy** refers to the generation of unnecessary responses that result in an EM of zero.

Compared to previous datasets, SCITAT presents the following challenges. (*i*) Free-form and short-form answers are associated with different error types, necessitating the design of distinct methods. (*ii*) SCITAT requires integration of various reasoning types and the processing of both tables and text, demanding the model to have strong domain-specific knowledge in the scientific field. We outline these challenges to inspire future work in addressing these issues, aiming to enhance model performance in SQA on tables and text.

5 Related Works

5.1 Scientific QA Datasets

Early SQA datasets were designed in a cloze-style format, limiting their difficulty (Pampari et al., 2018; Pappas et al., 2018). To address this issue, PubMedQA (Jin et al., 2019), QASPER (Dasigi et al., 2021), and QASA (Lee et al., 2023) employ humans to annotate questions and answers over papers, and SciInstruct (Zhang et al., 2024) collects questions from sources like textbooks and synthesizes answers using LLMs. However, these works primarily focus on text, without considering the tables appearing in papers. Therefore, SciGen (Moosavi et al., 2021) focuses on generating descriptions based on tables in papers, SciTab (Lu et al., 2023) concentrates on the table fact verification, and SPIQA (Pramanick et al., 2024) is designed for QA based on tables and images.

Nevertheless, the reasoning types of existing datasets are relatively limited, since they do not involve diverse reasoning types, such as Data Analysis and Tabulation, that frequently occur in real scenarios. Moreover, they overlook the relevance between tables and text, limiting their application (Chen et al., 2020; Wang et al., 2022). Therefore, we propose SCITAT, a QA benchmark for scientific tables and text with diverse reasoning types.

5.2 QA Datasets for Tables and Text

Previous QA datasets for tables and text mainly focus on Look Up and Numerical Reasoning in the Wikipedia and financial domains. For example, HybridQA (Chen et al., 2020) annotates QA pairs over Wikipedia tables and text, which primarily focuses on look up spans in the context. TAT-QA (Zhu et al., 2021), FinQA (Chen et al., 2021), DocMath-Eval (Zhao et al., 2024c), and FinanceMATH (Zhao et al., 2024b) primarily address the numerical reasoning task in the financial domain. However, previous datasets focus on limited reasoning types, mainly Look Up and Numerical Reasoning, which differ significantly from the SQA scenarios in real-world applications. Furthermore, these datasets do not require models to possess domain-specific knowledge in science. A detailed comparison SCITAT with previous datasets for tables and text is shown in Appendix A.2.

Considering the reasoning types of existing datasets, previous works introduce programs to obtain the final answer (Gao et al., 2023; Chen et al., 2023; Zhang et al., 2023). For instance, BlendSQL (Glenn et al., 2024) prompts the LLM to generate a superset of SQL that can query information from both tables and text to obtain answers. Hpropro (Shi et al., 2024) provides commonly used program functions, allowing the LLM to directly call them during program generation. However, these methods can not apply directly to SCITAT, as SCITAT also involves reasoning types, such as Data Analysis, which is challenging to be solved by the program alone (Wu et al., 2024b). Therefore, we propose CAR, which combines multiple reasoning types to enhance performance on SCITAT.

6 Conclusion

To address the limitations of previous scientific QA datasets, which involve limited reasoning types and fail to consider the relevance between tables and text, we propose SCITAT, the QA benchmark for scientific tables and text with diverse reasoning types. To incorporate diverse reasoning types, we analyze the questions posed by researchers and combine the types in prior works, summarizing 4 reasoning types with 13 subtypes. To ensure that the questions encompass both tables and text, we require the questions to include both elements whenever possible. For SCITAT, we introduce CAR, a baseline that combines reasoning methods to enhance the performance across various reasoning types, with handling both tables and text. Experimental results show that CAR outperforms other baselines by an average of 4.1%, demonstrating its effectiveness. Error analysis reveals the challenges in SCITAT, such as grounding relevant context, complex numerical reasoning, and the need for domain-specific knowledge.

Limitations

(*i*) SciTaT currently supports only the English language. Future versions will include additional languages. (*ii*) Currently, we focus on single-turn QA for scientific tables and texts in SciTaT. Multi-turn dialogues on scientific tables and text will be explored in future work.

Ethics Statement

All models used in this paper are publicly available, and our utilization of them strictly complies with their respective licenses and terms of use. We collect papers from ariv.org following its terms of use and regulations. During the dataset construction process, we ensure that the collected papers are publicly accessible and do not infringe on any copyrights. Additionally, we confirm that the compensation provided to annotators is significantly higher than the local minimum wage.

References

Microsoft Research AI4Science and Microsoft Azure Quantum. 2023. The impact of large language models on scientific discovery: a preliminary study using gpt-4. *Preprint*, arXiv:2311.07361.

Emily M Bender and Batya Friedman. 2018. Data statements for natural language processing: Toward mitigating system bias and enabling better science. *Transactions of the Association for Computational Linguistics*, 6:587–604.

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W. Cohen. 2023. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *Transactions on Machine Learning Research*.

Wenhu Chen, Hanwen Zha, Zhiyu Chen, Wenhan Xiong, Hong Wang, and William Yang Wang. 2020. HybridQA: A dataset of multi-hop question answering over tabular and textual data. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1026–1036, Online. Association for Computational Linguistics.

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan Routledge, and William Yang Wang. 2021. FinQA: A dataset of numerical reasoning over financial data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3697–3711, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A. Smith, and Matt Gardner. 2021. A dataset of information-seeking questions and answers anchored in research papers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4599–4610, Online. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco Guzmán, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Rohan Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vítor Albiero, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Sorouralsadat Fatemi and Yuheng Hu. 2024. Enhancing financial question answering with a multi-agent reflection framework. In *Proceedings of the 5th ACM International Conference on AI in Finance*, pages 530–537.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Pal: program-aided language models. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Parker Glenn, Parag Dakle, Liang Wang, and Preethi Raghavan. 2024. BlendSQL: A scalable dialect for unifying hybrid question answering in relational algebra. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 453–466, Bangkok, Thailand. Association for Computational Linguistics.

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, et al. 2024. Qwen2. 5-coder technical report. *arXiv preprint arXiv:2409.12186*.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. 2019. PubMedQA: A dataset for biomedical research question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2567–2577, Hong Kong, China. Association for Computational Linguistics.

Yoonjoo Lee, Kyungjae Lee, Sunghyun Park, Dasol Hwang, Jaehyeon Kim, Hong-In Lee, and Moontae Lee. 2023. QASA: Advanced question answering on scientific articles. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 19036–19052. PMLR.

Zicheng Lin, Zhibin Gou, Tian Liang, Ruilin Luo, Haowei Liu, and Yujiu Yang. 2024. CriticBench: Benchmarking LLMs for critique-correct reasoning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 1552–1587, Bangkok, Thailand. Association for Computational Linguistics.

Xinyuan Lu, Liangming Pan, Qian Liu, Preslav Nakov, and Min-Yen Kan. 2023. SCITAB: A challenging benchmark for compositional reasoning and claim verification on scientific tables. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7787–7813, Singapore. Association for Computational Linguistics.

Nafise Sadat Moosavi, Andreas Rücklé, Dan Roth, and Iryna Gurevych. 2021. Scigen: a dataset for reasoning-aware text generation from scientific tables. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Anusri Pampari, Preethi Raghavan, Jennifer Liang, and Jian Peng. 2018. emrQA: A large corpus for question answering on electronic medical records. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2357–2368, Brussels, Belgium. Association for Computational Linguistics.

Liangming Pan, Michael Saxon, Wenda Xu, Deepak Nathani, Xinyi Wang, and William Yang Wang. 2024. Automatically correcting large language models: Surveying the landscape of diverse automated correction strategies. *Transactions of the Association for Computational Linguistics*, 12:484–506.

Dimitris Pappas, Ion Androutsopoulos, and Haris Papageorgiou. 2018. BioRead: A new dataset for biomedical reading comprehension. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Dimitris Pappas, Petros Stavropoulos, Ion Androutsopoulos, and Ryan McDonald. 2020. BioMRC: A dataset for biomedical machine reading comprehension. In *Proceedings of the 19th SIGBioMed Workshop on Biomedical Language Processing*, pages 140–149, Online. Association for Computational Linguistics.

Shraman Pramanick, Rama Chellappa, and Subhashini Venugopalan. 2024. Spiqa: A dataset for multimodal question answering on scientific papers. *NeurIPS*.

Qi Shi, Han Cui, Haofeng Wang, Qingfu Zhu, Wanxiang Che, and Ting Liu. 2024. Exploring hybrid question answering via program-based prompting. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11035–11046, Bangkok, Thailand. Association for Computational Linguistics.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*.

Liangtai Sun, Yang Han, Zihan Zhao, Da Ma, Zhennan Shen, Baocai Chen, Lu Chen, and Kai Yu. 2024. Scieval: A multi-level large language model evaluation benchmark for scientific research. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19053–19061.

Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. 2022. Galactica: A large language model for science. *arXiv preprint arXiv:2211.09085*.

Ye Tian, Baolin Peng, Linfeng Song, Lifeng Jin, Dian Yu, Lei Han, Haitao Mi, and Dong Yu. 2024. Toward self-improvement of LLMs via imagination, searching, and criticizing. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

George Tsatsaronis, Georgios Balikas, Prodromos Malakasiotis, Ioannis Partalas, Matthias Zschunke, Michael R Alvers, Dirk Weissenborn, Anastasia Krithara, Sergios Petridis, Dimitris Polychronopoulos, et al. 2015. An overview of the bioasq large-scale biomedical semantic indexing and question answering competition. *BMC bioinformatics*, 16:1–28.

Dingzirui Wang, Longxu Dou, and Wanxiang Che. 2022. A survey on table-and-text hybridqa: Concepts, methods, challenges and future directions. *arXiv preprint arXiv:2212.13465*.

Xiaoxuan Wang, Ziniu Hu, Pan Lu, Yanqiao Zhu, Jieyu Zhang, Satyen Subramaniam, Arjun Loomba, Shichang Zhang, Yizhou Sun, and Wei Wang. 2023. SCIBENCH: Evaluating college-level scientific problem-solving abilities of large language models. In *The 3rd Workshop on Mathematical Reasoning and AI at NeurIPS'23*.

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.

Dayong Wu, Jiaqi Li, Baoxin Wang, Honghong Zhao, Siyuan Xue, Yanjie Yang, Zhijun Chang, Rui Zhang, Li Qian, Bo Wang, Shijin Wang, Zhixiong Zhang, and Guoping Hu. 2024a. SparkRA: A retrieval-augmented knowledge service system based on spark large language model. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 382–389, Miami, Florida, USA. Association for Computational Linguistics.

Xianjie Wu, Jian Yang, Linzheng Chai, Ge Zhang, Jiaheng Liu, Xinrun Du, Di Liang, Daixin Shu, Xianfu Cheng, Tianzhen Sun, et al. 2024b. Tablebench: A comprehensive and complex benchmark for table question answering. *arXiv preprint arXiv:2408.09174*.

Yiyao Yu, Yuxiang Zhang, Dongdong Zhang, Xiao Liang, Hengyuan Zhang, Xingxing Zhang, Ziyi Yang, Mahmoud Khademi, Hany Awadalla, Junjie Wang, Yujiu Yang, and Furu Wei. 2025. Chain-of-reasoning: Towards unified mathematical reasoning in large language models via a multi-paradigm perspective. *Preprint*, arXiv:2501.11110.

Dan Zhang, Ziniu Hu, Sining Zhoubian, Zhengxiao Du, Kaiyu Yang, Zihan Wang, Yisong Yue, Yuxiao Dong, and Jie Tang. 2024. Sciinstruct: a self-reflective instruction annotated dataset for training scientific language models. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Zhehao Zhang, Xitao Li, Yan Gao, and Jian-Guang Lou. 2023. CRT-QA: A dataset of complex reasoning question answering over tabular data. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2131–2153, Singapore. Association for Computational Linguistics.

Yilun Zhao, Lyuhao Chen, Arman Cohan, and Chen Zhao. 2024a. TaPERA: Enhancing faithfulness and interpretability in long-form table QA by content planning and execution-based reasoning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12824–12840, Bangkok, Thailand. Association for Computational Linguistics.

Yilun Zhao, Hongjun Liu, Yitao Long, Rui Zhang, Chen Zhao, and Arman Cohan. 2024b. Financemath: Knowledge-intensive math reasoning in finance domains. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12841–12858, Bangkok, Thailand. Association for Computational Linguistics.

Yilun Zhao, Yitao Long, Hongjun Liu, Ryo Kamoi, Linyong Nan, Lyuhao Chen, Yixin Liu, Xiangru Tang, Rui Zhang, and Arman Cohan. 2024c. DocMath-eval: Evaluating math reasoning capabilities of LLMs in understanding long and specialized documents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16103–16120, Bangkok, Thailand. Association for Computational Linguistics.

Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. 2021. TAT-QA: A question answering benchmark on a hybrid of tabular and textual content in finance. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3277–3287, Online. Association for Computational Linguistics.

A Comparison with Previous Datasets

A.1 Comparison with Previous Scientific QA Datasets

Table 1 presents the comparison of SCITAT with previous scientific QA datasets. We first introduce the existing datasets. BioRead (Pappas et al., 2018) is a cloze-style QA dataset on the biomedical papers, which only conteains the reasoning type of Look Up and focuses only on the text. QASA (Lee et al., 2023) is QA datasets on papers in AI and ML fields, but only concentrates on the text in papers, and lack the reasoning type of Tabulation. SciGen (Moosavi et al., 2021) aims to generate descriptions according to the tables in the papers in the field of Computer Science. SciTab (Lu et al., 2023) aims to judge the claims according to the scientific tables in the field of Computer Science, which only contains the reasoning types of Lok Up and Numerical Reasoning. SPIQA (Pramanick et al., 2024) is mulmimodal QA dataset on the scientific papers, which only focus on the split text and tables, ignoring the relevance between tables and text, and lacking the reasoning type of Data Analysis and Tabulation. It can be seen that SCITAT contains more diverse reasoning types and consider the relevance between tables and text.

A.2 Comparison with Previous QA Datasets for Tables and Text

Table 7 present the comparison of SCITAT with previous QA datasets over tabular and textual data and scientific QA datasets. It can be seen that SCITAT contains more diverse and closer to real-life user questions. Moreover, SCITAT requires the model to possess domain-specific knowledge in the scientific field. It must not only understand the dense terminology commonly found in papers but also apply this knowledge to solve questions, which is not required by other datasets.

B Prompt

In this section, we show the prompts we use to synthesize data and conduct experiments.

B.1 Prompt for Generating Data

Table 8 provides the prompt for generating questions, rationales, and answers when constructing SCITAT.

B.2 Prompt for Experiments

Table 9 shows the prompt to build the baselines in our experiments, and Table 10 shows the prompt used by CAR. The prompts of Three-Agent we use are referred to the original paper (Fatemi and Hu, 2024).

C Manual Annotation Procedure

C.1 Annotator Training Process

We recruit students from Computer Science or Artificial Intelligence programs who are willing to participate in the annotation task, offering a compensation of $1 per instance. Initially, we provide a detailed explanation of the task, including its definition, the specific responsibilities of the annotators, and how to use the annotation interface. We thoroughly explain the requirements for the questions, rationales, and answers, as well as how to select the source of the answers, as stated in §2.3. Additionally, we provide three examples and explain possible scenarios that might arise. Finally, we clarify the annotation deadline and inform them that the data will undergo additional checks. To promptly detect any errors or biases in the annotations, we sent the data in batches. After the two-round validation on the already annotated data, we communicate with the annotators to address any issues and proceed to send the next batch of data.

C.2 Statistics of the Manual Annotation Procedure

On average, annotating a single data point required 10 minutes per annotator. The annotation process for the 953 instances was completed in approximately two months. The first round of annotations was conducted by 10 annotators, with two additional annotators performing two-round validation. Overall, 43.9% of the initially generated data is filtered out, and the remaining 43.5% of initial answers are both modified and filtered. This process involved correcting incorrect answers and streamlining responses to ensure accuracy and reduce LLM-induced biases.

C.3 Annotating Interface

The annotation process is conducted using a custom tool developed by us. Figure 7, Figure 8, Figure 9, and Figure 10 show the overall user interface for the manual annotation.

Dataset	Domain	Reasoning Type			
		Look Up	Numerical Reasoning	Data Analysis	Tabulation
HybridQA (Chen et al., 2020)	Wiki	✓	✗	✗	✗
TAT-QA (Zhu et al., 2021)	Finance	✓	✓	✗	✗
FinQA (Chen et al., 2021)	Finance	✗	✓	✗	✗
DocMath-Eval (Zhao et al., 2024c)	Finance	✓	✓	✗	✗
FinanceMATH (Zhao et al., 2024b)	Finance	✓	✓	✗	✗
SciTaT	Science	✓	✓	✓	✓

Table 7: Comparison of SciTaT to recent QA datasets over tabular and textual data. Wiki denotes Wikipedia.

The prompt for Generating Questions

{Table}
{Paragraph}
You are a highly intelligent and obedient academic field question generation system.
Generate a question referring to the table and paragraph above which meets the requirements in the question description "{Type}".
The generated question must meet:
1. The question should be with fewer statements and more reasoning and calculation.
2. The question must be answerable based on the paragraph alone, and not answerable only based on the table.
3. The question must meet the question description.
4. Do not generate multiple questions or sub-questions at once.
Examples:
{Examples}

The prompt for Generating Rationales and Answers

{Table}
{Paragraph}
Based on the information in the Table and Paragraph, please answer the question "{Question}".
Represent your answer with: "Reason: <Your Reason> Answer: <Your Answer>"
If there are multiple questions, you need to answer them one by one, and the answers are separated by "

".
Examples:
{Examples}

Table 8: The prompts for generating the questions, rationales, and answers of SciTaT.

D Case Study for Error Analysis

In this section, we show examples of different error types, as shown in Figure 11, Figure 12, Figure 13, Figure 14, and Figure 15.

E Additional Experiments

In this section, we present additional experiments.

E.1 Results of Few-shot Prompts

In this subsection, we present the performance of CaR with the few-shot prompts in the short-context setting, as shown in Table 11. Specifically, we annotate 4 demonstrations, each corresponding to one of the four reasoning types in SciTaT. It can be observed that the performance of SciTaT with the few-shot prompts outperforms that under the zero-shot prompts.

E.2 Statistics of The Number of Output Tokens

In this subsection, we show the comparison of the number of tokens output by different methods and the number of tokens of gold answers. (*i*) It can be found that the number of tokens output by PoT is consistently lower than that of other methods, whether it is a short-form answer or a free-form answer, which explains to a certain extent the reason why PoT has low performance, especially on the free-form answers. (*ii*) On the contrary, the number of tokens output by the Direct QA is generally high, which also reveals the reason why its EM is 0 on the short-form answers. (*iii*) And CaR is the closest in quantity to the number of tokens of gold answer, which shows that CaR can adapt to obtain answers of various reasoning types.

The prompt for DirectQA
Based on the information in the Table and Paragraph, you should answer the question. If there are multiple questions, you need to answer them one by one, and the answers are separated by " ". Table (including its label, caption, and content): {Table} Paragraph: {Paragraph} Please answer the question "{Question}".

The prompt for CoT
Based on the information in the Table and Paragraph, you should answer the question. Represent your answer with: "Reason: <Your Reason> Answer: <Your Answer>". If there are multiple questions, you need to answer them one by one, and the answers are separated by " ". Table (including its label, caption and content): {Table} Paragraph: {Paragraph} Please answer the question "{Question}".

The prompt for PoT
Table (including its label, caption and content): {Table} Paragraph: {Paragraph} Read the above Table and Paragraph, and then write code to answer the question "{Question}". Please **directly use** the information such as numbers in tables and paragraphs, do not define tables and then process them. You must return the answer 'ans = ' at the end of the code instead of 'print'. Attention that if there are multiple questions, you need to answer them one by one, and the answers are separated by " ".

Table 9: The prompts for baselines.

E.3 Influence of Order on CaR

To study the influence of order on the performance of CaR, we conduct experiments by reversing the two modules, with results presented in Table 13. Specifically, we first apply the Reasoner module and then feed its output into the Calculator, which verifies and corrects any numerical errors to produce the final result. The significant performance drop confirms the validity of CaR. The results suggest that depending on the program output for the final answer limits performance on SciTaT since the program struggles with free-form responses, as discussed in §4.2. The lower performance of PoT, as shown in Table 5, also supports our view.

F Discussion of Details

F.1 Why Papers from Three Subfields are Selected?

(*i*) The three subfields selected in SciTAT follow **mainstream science datasets**, including QASPER (Dasigi et al., 2021), QASA (Lee et al., 2023), SciGen (Moosavi et al., 2021), and SciTab (Lu et al., 2023). (*ii*) Our primary focus is on addressing diverse scientific questions and reasoning types, which are independent of the specific domains.

F.2 Why the Baselines are Chosen?

The selected baselines in the main experiments follow those used in previous datasets (SciTab (Lu et al., 2023), SciEval (Sun et al., 2024), SPIQA (Pramanick et al., 2024), SciBench (Wang et al., 2023)). To further validate the necessity

The prompt for Calculator

Table (including its label, caption, and content):
{Table}
Paragraph:
{Paragraph}
Read the above Table and Paragraph, and then write code to answer the question "{Question}".
Please **directly use** the information such as numbers in tables and paragraphs, do not define tables and then process them.
You must return the answer ‘ans = ‘ at the end of the code instead of ‘print‘.
You cannot return just one or a few numbers or words, you must return a complete sentence.

The prompt for Reasoner

Based on the Table and Paragraph with the Tips, you should answer the question.
Please determine whether the tips are correct, use the tips reasonably in Reason, and organize the Answer into an appropriate form.
Represent your answer with: "Reason: <Your Reason> Answer: <Your Answer>".
Attention that if there are multiple questions, you need to answer them one by one, and the answers are separated by
"

".

Table (including its label, caption, and content):
{Table}
Paragraph:
{Paragraph}
Tips:
{Tips}
Please answer the question "{Question}".

Table 10: The prompts for CAR.

Model	Scale	Method	EM	F1	BERTScore
Llama3.1	8B	CAR (zero-shot)	24.2	44.3	73.2
		CAR (few-shot)	**28.7**	**48.0**	**76.2**
	70B	CAR (zero-shot)	40.7	46.2	74.4
		CAR (few-shot)	**43.4**	**46.8**	**74.6**

Table 11: The results of CAR with few-shot prompts compared with zero-shot prompts.

of SCITAT and the effectiveness of CAR, we add Three-Agent (Fatemi and Hu, 2024), which is the state-of-the-art method for reasoning on both tables and text. The previous methods in the scientific question answering (SQA) task primarily focus on model training (Dasigi et al., 2021; Lee et al., 2023; Zhang et al., 2024), which makes direct comparisons with CAR infeasible. Therefore, we do not compare CAR with the baselines in the SQA task.

Now on Page **1**, total **222** pages

Select the Specific Page

Aready annotated 0 / 222 instances

Paper Information

URL: http://arxiv.org/pdf/2006.08332v1

Reasoning Type: Domain knowledge Caculation

Question: Based on the information provided in the paragraph and the table, if the Statistical Machine Translation system's dataset consists of 24k parallel sentences and achieves a BLEU score of 41, estimate the BLEU score improvement necessary for the Proposed neural machine translation system to achieve an average BLEU score improvement of 10% across similar datasets.

Predicted Answer: Reason: The Statistical Machine Translation (SMT) system has a BLEU score of 41. A 10% improvement over this score would mean a BLEU score improvement of 10% of 41. Therefore, the necessary BLEU score improvement for the Proposed neural machine translation system over the SMT system would be 0.10 * 41.Answer: 4.1 (an improvement to a BLEU score of 45.1 is needed for a 10% improvement).

Paragraph

\section {Sanskrit Hindi Machine Translation Systems} \label{Sanskrit Hindi Machine Translation Systems} The comparisons of the existing and proposed system are shown in Table. \ref{tab:comp} \begin{itemize} \subsection{Sanskrit-Hindi Anusaarka-2009} \item Approach:It is a \textbf{rule based} MT system \cite{bharati2009anusaaraka}. \item The insights for the system are taken from Panini's Ashtadhyayi. \item Developed By:Chinmaya International Foundation (CIF), Indian Institute of Information Technology, Hyderabad (IIIT-H) and University of Hyderabad at Department of Sanskrit Studies. \item Tool:\textbf{Samsadhani} It is a Language accessor cum machine translation system. Input can be of an of the following encodings Unicode-Devanagari, WX-alphabetic, Itrans 5.3, Velthuis (VH), Harward Kyoto (KH), Sanskrit Library Project (SLP). Output can be displayed in either Devanagari script or in Roman Diacritical Notation. The system has a Sanskrit language analyser which does the analysis

Figure 7: The user interface, showing the paper information and the paragraph.

Tables

Table 1:

Label: tab:comp

Caption: Comparison of Sanskrit Hindi Translation systems

\textbf{Sanskrit-Hindi (Rule Based)}	\textbf{Sanskrit-Hindi (Statistical)}	\textbf{Sanskrit Hindi (Proposed)}
The system fails when extended to large domains. It is developed for domains like kids stories, building interactive media and e-learning substance for kids.	Sanskrit-Hindi text corpora has been collected or developed manually from the literature, health, news and tourism domains.	The proposed system is based on neural machine translation technique that covers all domains in general based on the dataset used.
Hand crafted rules based on Panini's Ashtadhyayi.	Based on Bayes theorem. Similar words assigned random numbers.	Word embedding used, similar words have close numbers.
Separate modules like tokenizer, sandhi splitter, morph analyser, parser, word sense disambiguation, part of speech tagger, chunker, Hindi lexical transfer and a Hindi language generator used to get translation.	SMT system have three separate main components-The translation model, reordering model, and the language model.SMT would evaluate fluency of a sentence in a target language a few words at a time using N-gram language model.	Proposed system learns complex relationship between languages as one single model. Proposed model considers the entire sentence.

Figure 8: The user interface, showing the tables.

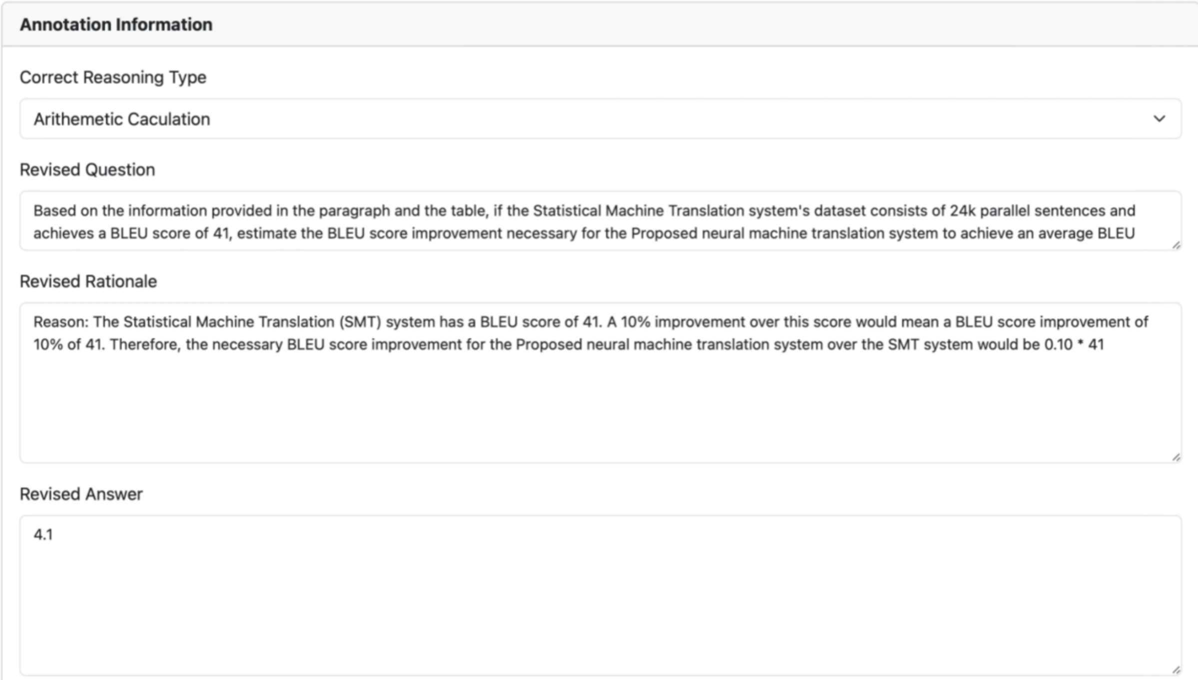


Figure 9: The user interface, showing the annotation information.

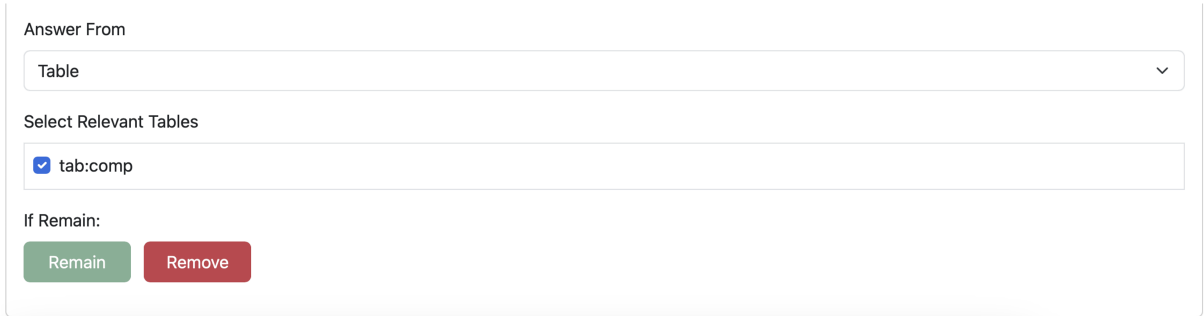


Figure 10: The user interface, showing the choice for the answer source, relevant tables, and if to remain.

Figure 5 essentially confirms the results obtained so far. However, we now observe, for higher α values, that the cluster values of textbook-based networks become seemingly indistinguishable from those observable for Wikipedia corpora-based networks – the same observation concerns the SZ-Eco-based networks. Textbook-based TCNs are again hardly distinguishable from TCNs derived from informal language newspaper articles about topics related to economics (SZ-Eco). In any case, Table 7 also shows that all value distributions along the x and y axis are now distinguishable with only three exceptions: the dynamics of clustering is obviously more corpus specific.

x-values:

	SZ-All	SZ-Eco	TB	WP-All	WP-Eco	WP-Top-1	WP-Top-3	Zeit-All
SZ-All	—	$1.332{,}3 \times 10^{-15}$	$6.661{,}3 \times 10^{-16}$	$3.774{,}8 \times 10^{-15}$	$3.774{,}8 \times 10^{-15}$	$3.254{,}7 \times 10^{-300}$	$2.164{,}6 \times 10^{-299}$	$1.443{,}3 \times 10^{-15}$
SZ-Eco	—	—	$2.133{,}0 \times 10^{-05}$	$1.554{,}3 \times 10^{-15}$	$1.554{,}3 \times 10^{-15}$	$1.332{,}3 \times 10^{-15}$	$1.332{,}3 \times 10^{-15}$	$1.110{,}2 \times 10^{-16}$
TB	—	—	—	$1.554{,}3 \times 10^{-15}$	$1.554{,}3 \times 10^{-15}$	$6.661{,}3 \times 10^{-16}$	$6.661{,}3 \times 10^{-16}$	$1.554{,}3 \times 10^{-15}$
WP-All	—	—	—	—	0.304,7	$9.832{,}6 \times 10^{-07}$	0.000,2	$2.109{,}4 \times 10^{-15}$
WP-Eco	—	—	—	—	—	$1.718{,}7 \times 10^{-07}$	$6.944{,}0 \times 10^{-05}$	$2.109{,}4 \times 10^{-15}$
WP-Top-1	—	—	—	—	—	—	0.477,5	$1.443{,}3 \times 10^{-15}$
WP-Top-3	—	—	—	—	—	—	—	$1.443{,}3 \times 10^{-15}$
Zeit-All	—	—	—	—	—	—	—	—

y-values:

	SZ-All	SZ-Eco	TB	WP-All	WP-Eco	WP-Top-1	WP-Top-3	Zeit-All
SZ-All	—	1.000,0	$6.661{,}3 \times 10^{-16}$	$3.774{,}8 \times 10^{-15}$	$3.774{,}8 \times 10^{-15}$	$1.086{,}4 \times 10^{-54}$	$2.445{,}3 \times 10^{-71}$	$1.443{,}3 \times 10^{-15}$
SZ-Eco	—	—	$1.221{,}2 \times 10^{-15}$	$1.554{,}3 \times 10^{-15}$	$1.554{,}3 \times 10^{-15}$	$1.332{,}3 \times 10^{-15}$	$1.332{,}3 \times 10^{-15}$	$1.110{,}2 \times 10^{-16}$
TB	—	—	—	$1.176{,}7 \times 10^{-09}$	$5.162{,}5 \times 10^{-14}$	$1.477{,}7 \times 10^{-11}$	$4.872{,}1 \times 10^{-10}$	$1.767{,}3 \times 10^{-10}$
WP-All	—	—	—	—	$6.728{,}1 \times 10^{-05}$	$3.734{,}6 \times 10^{-05}$	0.012,6	$7.908{,}8 \times 10^{-12}$
WP-Eco	—	—	—	—	—	$4.078{,}8 \times 10^{-11}$	$3.815{,}2 \times 10^{-06}$	$2.109{,}4 \times 10^{-15}$
WP-Top-1	—	—	—	—	—	—	0.000,9	$1.042{,}1 \times 10^{-12}$
WP-Top-3	—	—	—	—	—	—	—	$3.153{,}0 \times 10^{-14}$
Zeit-All	—	—	—	—	—	—	—	—

Table 7: P-values of the Kolmogorov-Smirnov goodness-of-fit test applied to the pairwise combinations of the x and y values of the distributions in Fig. 5.

Question:
Which tables illustrate the indistinguish ability of textbook-based networks from Wikipedia corpora-based networks for higher $\\alpha$ values, and how does the paragraph support this observation?

Predicted Answer:
WP-All vs WP-Eco, WP-Top-1 vs WP-Top-3, and WP-Top-1 vs WP-Top-3.

Gold Answer:
Table tab:cws-x

The paragraph mentions that for higher $\\alpha$ values, that the cluster values of textbook-based networks become seemingly indistinguishable from those observable for Wikipedia corpora-based networks.

Figure 11: The case for the error type of "Miss".

We propose two calculations. First, we calculate the maximum number of bi-sentences that an individual can acquire by the age of 23. The results are given in Table 1. We first estimate the daily acquisition time, then we deduce the number of sentences acquired annually.

	Age	Sleep Hours	awake time -2H	occ. sent. per day	Cumul. per year (Million sent.) Analysis 1 bi-sent.	Analysis 2 L1	L2	bi-sent.
Infant	1	16	6	5,760	1.1	1.1		
Young child	2	16	6	5,760	2.1	3.2		
	3	14	8	7,680	3.5	6.0		
	4∼5	12	10	9,600	5.3	9.5		
Beg.prim.school	6	12	10	9,600	8.8	16.5		
	7∼9	11	11	10,560	10.7	20.3		
End prim. school	10	11	11	10,560	16.5	31.9		
	11	10	12	11,520	18.6	36.1		
	12	10	12	11,520	20.7	40.3		
	13	9	13	12,480	23.0	44.9		
End middle school	14∼17	8	14	13,440	25.4	49.8		
End high school	18	8	14	13,440	35.2	69.4		
	19∼20	8	14	13,440	37.7	70.6	1.2	1.2
End Bachelor	21∼22	8	14	13,440	42.6	73.1	3.7	3.7
End Master	23	8	14	13,440	47.5	75.5	6.1	6.1

Table 1: Cumulative number of sentences read or heard by a professional translator

Question:
How does the calculated cumulative number of bi-sentences by the end of a Bachelor's degree differ between Analysis 1 and Analysis 2 based on the interpretation of the data in Table \ref{tab1}?

Predicted Answer:
We cannot calculate the difference between Analysis 1 and Analysis 2 based on the data in Table \\ref{tab1}.

Gold Answer:
Analysis 1 is 42.6 million, and Analysis 2 is 3.7 million; the difference is 38.9 million bi-sentences.

Figure 12: The case for the error type of "Locate".

The results from the COMPAS dataset are detailed in Fig. 3, where minority population density, disparity, and loss measures are shown. In this scenario, the optimal control method shows comparable performance to RL-based algorithms. Specifically, TRPO and PPO reach a terminal state with a loss value of 0.574, which achieves the same level of performance compared to the proposed optimal control method. This similarity in performance is attributed to the lesser disparity in representation between different gender attributes within the COMPAS dataset.

Table 3: COMPAS: Terminal State with Initial $\lambda_0^1 = 0.6$ and $\lambda_0^2 = 0.4$

Environment M_1^*							
	Fair-agnostic	Fair-aware		Dynamic-aware			
	ERM	Minimax	DRO	PG	TRPO	PPO	Optim
Density-2 ↑	0.271	0.256	0.184	0.263	0.264	0.257	0.274
Disparity ↓	0.522	0.529	0.153	0.435	0.489	0.442	0.516
Loss ↓	0.770	0.803	1.391	0.848	0.808	0.858	0.764
Environment M_2^*							
	Fair-agnostic	Fair-aware		Dynamic-aware			
	ERM	Minimax	DRO	PG	TRPO	PPO	Optim
Density-2 ↑	0.316	0.298	0.075	0.317	0.317	0.317	0.317
Disparity ↓	0.684	0.702	0.925	0.863	0.683	0.683	0.683
Loss ↓	0.577	0.605	1.296	0.594	0.574	0.574	0.574

Question:
Which algorithm ranks the highest in terms of minimizing both disparity and loss across Environment $M^{\\ast}_2$?

Predicted Answer:
TRPO

Gold Answer:
TRPO, PPO, and Optim

Figure 13: The case for the error type of "Calculate".

In particular, given a graph, we first extract all cycles out of it. Then, all edges that are not inside the cycles are considered motifs. We consider combining cycles with more than two coincident nodes into a motif. Although this method cannot extract complex motifs like single-input and multi-input motifs, it can generate the most important motifs, such as ring structures in biochemical molecules and the feed-forward loop motif. By adopting this simple but general motif extraction method, we can explain a GNN model without any domain knowledge, making our explanation model more applicable. Need to be noted that, even though the motif extraction rule cannot extract single-input and multi-input motifs, these motifs can be implicitly identified by our attention layer. Experiments in the table 2 demonstrate it.

TABLE 2
Results on quantitative studies for different explanation methods. Note that since the Sparsity cannot be fully controlled, we report Fidelity scores (The less the better) under similar Sparsity levels for five real-world datasets. For two synthetic datasets, BA-Shape and BA-2Motif, we report accuracy. S is the sparsity value. K is the maximum number of edges required by baseline models. Our MotifExplainer does not need this required hyper-parameter. The best performances on each dataset are shown in **bold**.

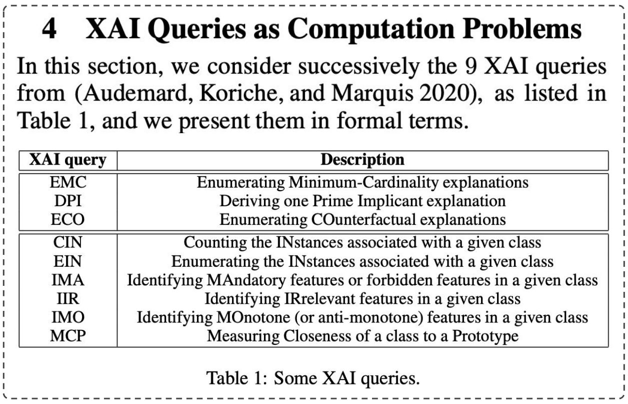
	MUTAG S=0.7	PTC S=0.7	NCI1 S=0.7 (Fidelity)	PROTEINS S=0.7	IMDB S=0.7	BA-2Motifs K=5	BA-Shape K=5 (Accuracy)
GNNExplainer	0.260	0.441	0.365	0.453	0.365	0.742	0.925
PGExplainer	0.241	0.388	0.402	0.521	0.225	0.926	0.963
SubgraphX	0.287	0.227	0.303	0.021	0.167	0.774	0.874
ReFine	0.221	0.349	0.409	0.435	0.127	0.932	0.954
MotifExplainer	**0.031**	**0.129**	**0.115**	**-0.030**	**0.101**	**1.0**	**1.0**

Question:
Using the explanation method described in the paragraph, calculate the number of motifs in a biochemical molecule represented by a graph where all nodes form a single ring and there are no other connecting edges outside this cycle.

Predicted Answer:
1

Gold Answer:
0

Figure 14: The case for the error type of "Knowledge".



4 XAI Queries as Computation Problems

In this section, we consider successively the 9 XAI queries from (Audemard, Koriche, and Marquis 2020), as listed in Table 1, and we present them in formal terms.

XAI query	Description
EMC	Enumerating Minimum-Cardinality explanations
DPI	Deriving one Prime Implicant explanation
ECO	Enumerating COunterfactual explanations
CIN	Counting the INstances associated with a given class
EIN	Enumerating the INstances associated with a given class
IMA	Identifying MAndatory features or forbidden features in a given class
IIR	Identifying IRrelevant features in a given class
IMO	Identifying MOnotone (or anti-monotone) features in a given class
MCP	Measuring Closeness of a class to a Prototype

Table 1: Some XAI queries.

Question:
If the task requires solving EMC and DPI queries successively, with EMC taking 10 minutes and DPI taking 5 minutes, what is the total time taken to perform 3 EMC queries and 2 DPI queries?

Predicted Answer:
3 EMC queries take 3 x 10 = 30 minutes, and 2 DPI queries take 2 x 5 = 10 minutes. Therefore, the total time taken to perform 3 EMC queries and 2 DPI queries is 30 + 10 = 40 minutes.

Gold Answer:
40 minutes

Figure 15: The case for the error type of "Redundancy".

Model	Scale	Method	Long-context Short-form Answers	Long-context Free-form Answers	Short-context Short-form Answers	Short-context Free-form Answers
-	-	Gold Answer	1.5	45.1	1.5	45.1
Llama3.1	8B	Direct QA	120.3	197.7	109.3	135.1
		CoT	35.8	75.4	23.3	74.4
		PoT	5.3	26.6	2.8	25.1
		CAR	24.3	54.8	21.4	53.8
	70B	Direct QA	118.9	220.1	118.9	220.1
		CoT	5.9	44.2	14.3	46.2
		PoT	3.6	30.0	3.3	36.4
		CAR	17.1	43.5	16.5	43.0
gpt-4o	-	Direct QA	151.2	213.2	105.7	141.6
		CoT	30.1	84.5	20.7	69.1
		PoT	3.0	24.4	4.6	52.4
		CAR	10.6	67.0	10.6	82.4

Table 12: Statistics of the number of tokens of gold answers and different results.

Scale	Method	EM	F1	BERTScore
8B	CaR	**24.2**	**44.3**	**73.2**
	Reversing	21.5	21.4	47.2
70B	CaR	**40.7**	**46.2**	**74.4**
	Reversing	37.1	35.9	65.1

Table 13: The results of CaR, compared with reversing the two modules (*Reversing*) in the short-context setting.