Integrating Table Representations into Large Language Models for Improved Scholarly Document Comprehension

Buse Sibel Korkmaz
Imperial College London
London, UK
buse.korkmaz18@imperial.ac.uk

Antonio del Rio Chanona
Imperial College London
London, UK
a.del-rio-chanona@imperial.ac.uk

Abstract

We address the challenge of interpreting and reasoning over scientific tables with Large Language Models (LLMs), a crucial aspect of scholarly documents. Despite significant progress in natural language processing, the integration of tabular data into scientific LLMs remains limited. We propose an innovative approach leveraging intermediate task pre-training on table question-answering datasets, followed by model adaptation to comprehend tables in computer science literature. Our findings reveal that incorporating table understanding substantially improves the performance of LLMs on scientific literature understanding tasks, which we showcase in peer-review score prediction. This improvement underscores the importance of utilizing tabular data in the training of scientific language models. The code and models are publicly available.[1]

1 Introduction

Large Language Models (LLMs) have experienced significant advancements in recent years and have been adapted to numerous natural language understanding and generation tasks (Zhao et al., 2023). Particularly in the scientific community, they have received increasing attention with their applications to scientific literature understanding tasks such as citation prediction (Cohan et al., 2020), paper classification (Zhang et al., 2023d), scientific literature search (Faggioli et al., 2023; Lála et al., 2023) and paper recommendation (Kanakia et al., 2019) to accelerate scientific discovery. In addition, domain-specialized research assistant language models have been developed (Beltagy et al., 2019; Luo et al., 2022; Taylor et al., 2022; Azerbayev et al., 2024).

Although these specialized models on scientific texts demonstrate success in the scientific literature understanding benchmarks such as MAPLE (Zhang et al., 2023d) and SciRepEval (Singh et al., 2023), the benchmarks and the corpora used in the training of these scientific language models predominantly focus on textual data. A critical component - and the focus of this study - often remains overlooked, which is *tables*. Tables encapsulate key findings, offering a condensed view of the research outcomes. In this work, contrary to the existing approaches, we hypothesize that understanding tables can significantly enhance the performance of LLMs on scientific literature tasks by providing a more holistic understanding of research papers.

We first tackle the challenge of interpreting tables and reasoning over them to answer questions requiring arithmetic operations and choosing information from specific cells through intermediate task pre-training. Then, we adapt our trained model to comprehend scientific tables in published computer science papers. This training process is designed to enable the models to reason with scientific table data. The scientific tables dataset we use is fundamentally different from the datasets used in intermediate task pre-training for table question-answering, incorporating more extensive summaries of scientific tables. Finally, we demonstrate that utilizing table representations extracted from fine-tuned LLMs with our approach improves the prediction of peer-review scores.

Overall, we develop a pipeline that allows LLMs to incorporate scientific knowledge from tables. The main contributions of this work are: (i) we apply an intermediate task pre-training approach that allows LLMs to understand tables, (ii) we do a detailed comparison of scientific table understanding by different models with different sizes, architectures, and under various settings, and (iii) we show how learning to represent scientific tables improves the understanding of scholarly documents, using the peer-review score prediction as a case study.

[1] https://github.com/buseskorkmaz/Integrating-Table-Representations-into-LLMs

Proceedings of the Fourth Workshop on Scholarly Document Processing (SDP 2024), pages 293–306

2 Related Work

2.1 Scientific language models

The majority of pre-training datasets for scientific LLMs consist primarily of textual data, with a notable absence of tables. Widely used datasets in pre-training such as MAPLE (Zhang et al., 2023d), SciFact (Wadden et al., 2020), SciERC (Luan et al., 2018), ACL-ARC (Bird et al., 2008), SciCite (Cohan et al., 2019), GENIA (Kim et al., 2003), and BC5CDR (Li et al., 2016) include only titles, abstracts, references, or citations. The S2ORC (Lo et al., 2020) dataset includes full texts with parsed tables, yet its potential for enhancing table understanding in LLMs remains largely under-explored.

Models such as SciBERT (Beltagy et al., 2019) have been trained exclusively on words and sentences from scientific texts. Similarly, SPECTER (Cohan et al., 2020) focuses on titles, abstracts, and citations, without incorporating table data into its training process. BioMedGPT (Zhang et al., 2023a) acknowledges the significance of tabular data understanding but leaves it as a future task. Even recently developed models such as SciMult (Zhang et al., 2023c) and SciNCL (Ostendorff et al., 2022), which includes the S2ORC (Lo et al., 2020) dataset in its training mix, fail to leverage table data effectively. SciMult is trained on datasets of MAPLE, SciFact, and SciRepEval (Singh et al., 2023), which do not include tabular data, and SciNCL, despite its access to a dataset with parsed scientific tables (S2ORC) does not utilize table data in the training.

2.2 Table understanding

Recent advancements in table understanding have seen significant contributions. Pasupat and Liang (2015) introduced a compositional semantic parsing approach, which established the WikiTQ dataset for benchmarking. TAPAS by Herzig et al. (2020), leveraged the BERT architecture (Devlin et al., 2019), and advanced table parsing by identifying operations through a classification layer for answer generation. Eisenschlos et al. (2020) focused on enhancing table entailment through pre-training on open-source tables, aligning closely with our approach in Section 3.2. Hegselmann et al. (2023) explored the application of LLMs for few-shot classification of tabular data. Li et al. (2023) recognized the value of information in tables and developed a scientific information extraction pipeline to improve data availability for tabular content within scientific papers.

Moreover, improvements in table understanding have enhanced adjacent tasks such as table-based fact verification, as seen with the TabFact dataset (Chen et al., 2020), and extended to specialized fields such as finance, demonstrated by the TAT-QA benchmark (Zhu et al., 2021). Zhang et al. (2023b) developed a generalist table understanding model, TableLlama based on LLaMA-2 (7B) (Touvron et al., 2023) using fine-tuned 1.24M tables for 8 different table-based tasks such as table interpretation, augmentation and QA. We evaluate their model for the scientific table understanding task to investigate the capabilities of a generalist model in a scientific domain.

2.3 Peer-review prediction

The utilization of language models in predicting peer review outcomes, as highlighted by Rogers and Augenstein (2020), reflects their potential to understand the scientific literature. Accurately predicting the quality of scientific research through models could address the subjectivity, biases, and inefficiencies identified in the peer review process (Shah, 2022).

The PeerRead dataset (Kang et al., 2018) serves as a foundational dataset for peer-review prediction research, covering acceptance outcomes and review helpfulness. The availability of public peer-review datasets has accelerated the expansion of peer-review research, including studies on review content and decision outcomes (Gao et al., 2019), the introduction of innovative approaches to publication representation (Muangkammuen et al., 2023) and the development of predictive models for review scores.

In peer-review prediction, the accurate construction of scholarly document representations is important to learn the correct relationship between the documents and their peer reviews. The PeerRead dataset (Kang et al., 2018) includes comprehensive details of document bodies and associated peer reviews along with outcomes. Despite the dataset's richness, the predominant methodology focuses on utilizing only the textual components of documents for representation. For example, peer-review prediction models DeepSentiPeer (Ghosal et al., 2019) and PeerAssist (Bharti et al., 2021) rely on the Science Parse library by AllenAI for extracting information from scholarly documents in PeerRead. Unfortunately, this library does not parse tables. This is a limitation if we are to capture the full scope of a scholarly document for peer review pre-

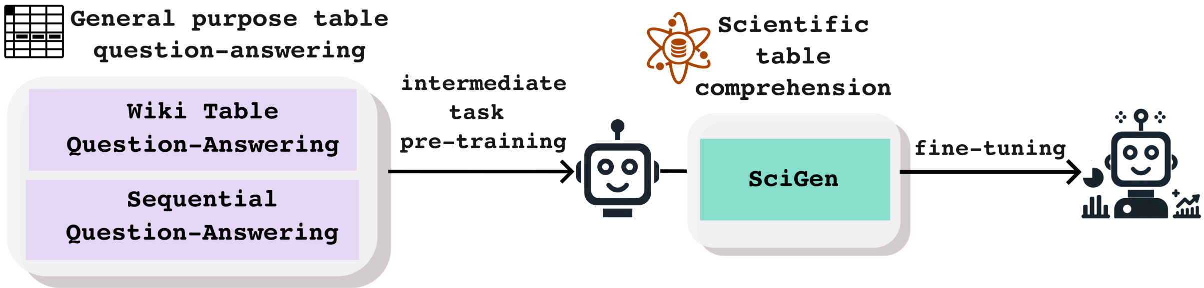

Figure 1: Overview of the training methodology for enhancing large language models with scientific table understanding. The training process begins with intermediate task pre-training using the WikiTQ and SQA datasets to build foundational table reasoning skills. This is followed by fine-tuning on the SciGen dataset to adapt the model specifically for scientific tables. The final model effectively integrates structured table data into text, improving performance on scientific literature tasks such as peer-review score prediction.

diction, as findings in result tables can substantially influence review outcomes.

3 Methodology

3.1 Datasets

WikiTableQuestions (WikiTQ) (Pasupat and Liang, 2015) WikiTQ is a benchmark dataset designed for evaluating the ability of models to perform question-answering (QA) over complex tables sourced from Wikipedia. This dataset challenges models to understand and interpret tabular data in context, requiring both a semantic understanding of questions and the ability to extract and reason relevant information from structured data. The inclusion of WikiTQ in our intermediate task pre-training regimen aims to enhance the model's proficiency in handling structured data and improve its capability to reason over tables, an essential skill for understanding scientific tables.

SQA (Iyyer et al., 2017) The SQA dataset extends the complexity of QA by introducing a sequential aspect, where answers to follow-up questions depend on the context established by previous interactions. However, our end use case is to describe scientific tables that do not have a conversational nature. Hence, we use a portion of the SQA dataset including the first questions in the sequence of questions over a given table. This dataset enables our model to further improve fundamental table understanding by adding diversity to the set of questions.

SciGen (Moosavi et al., 2021) SciGen stands out for its focus on generating coherent and contextually accurate textual descriptions from scientific tables, primarily containing numerical data. The ability of arithmetic reasoning to interpret tables in scientific papers and generate appropriate textual narratives presents the main challenge we aim to address. Thus, we subsequently fine-tune our model to adapt scientific tables on the SciGen dataset following pre-training on WikiTQ and SQA datasets.

	ALCHEMY		TANGRAMS		SCENE	
system	3utts	5utts	3utts	5utts	3utts	5utts
LONG+16	56.8	52.3	64.9	27.6	23.2	14.7
REINFORCE	58.3	44.6	**68.5**	**37.3**	47.8	33.9
BS-MML	58.7	47.3	62.6	32.2	53.5	32.5
RANDOMER	**66.9**	**52.9**	65.8	37.1	**64.8**	**46.2**

Parse columns, rows and values in the table

```
<R> <C> system <C> Alchemy 3utts <C> Alchemy 5utts
<C> Tangrams 3utts <C> Tangrams 5utts <C> Scene
3utts <C> Scene 5utts <R> <C> Long+16 <C> 56.8 <C>
52.3 <C> 64.9 <C> 27.6 <C> 23.2 <C> 14.7 <R> <C>
REINFORCE <C> 58.3 <C> 44.6 <C> [BOLD] 68.5 <C>
[BOLD] 37.3 <C> 47.8 <C> 33.9 <R> <C> BS-MML <C>
58.7 <C> 47.3 <C> 62.6 <C> 32.2 <C> 53.5 <C> 32.5
<R> <C> RandoMer <C> [BOLD] 66.9 <C> [BOLD] 52.9
<C> 65.8 <C> 37.1 <C> [BOLD] 64.8 <C> [BOLD] 46.2
```

Figure 2: An example of parsing tables for use with large language models. The table (Guu et al., 2017) structure is encoded using special tokens, with rows represented by <R>, columns by <C>, and associated captions by <CAP> as in (Moosavi et al., 2021).

3.2 Experimental Setup

Pre-trained LLMs We use FlanT5 (Chung et al., 2022) and LLaMA-2 (Touvron et al., 2023) as pre-trained language models. Our task requires learning representations from structured tables. To compare how different architectures adapt to tabular data representation in our problem, we choose T5 (Roberts et al., 2019) and FlanT5 (Chung et al., 2022) to represent encoder-decoder architecture, and LLaMA-2 as a representative of decoder-only architectures.

Data pre-processing Following Moosavi et al. (2021), we denote rows with <R>, columns <C> and

Test Dataset	Setting	Model	Parameters	METEOR	ROUGE-1	BertS
Test (C&L)	Zero	T5-base*	0.22B	0.04	n/a	0.76
		T5-large*	0.77B	0.06	n/a	0.76
		FlanT5-small	0.08B	0.06	0.09	0.79
		FlanT5-base	0.25B	0.04	0.06	0.74
		FlanT5-large	0.78B	0.10	0.12	0.79
		FlanT5-xl	3B	0.08	0.10	0.78
		LLaMA2-7B-chat-hf	7B	0.08	0.07	0.70
		TableLlama	7B	0.13	0.14	0.77
Test (Other)	Zero	T5-base*	0.22B	0.04	n/a	0.76
		T5-large*	0.77B	0.06	n/a	0.76
		FlanT5-small	0.08B	0.06	0.08	0.79
		FlanT5-base	0.25B	0.05	0.07	0.74
		FlanT5-large	0.78B	0.11	0.12	0.79
		FlanT5-xl	3B	0.08	0.09	0.78
		LLaMA2-7B-chat-hf	7B	0.08	0.07	0.70
		TableLlama	7B	0.13	0.13	0.77

Table 1: The evaluation of pre-trained models (zero-shot referring to not fine-tuned) on the test datasets. The scores of the models with * are taken from the SciGen (Moosavi et al., 2021), except ROUGE-1 since it is not reported.

associated caption from scientific tables as `<CAP>`. Figure 2 demonstrates an example of this parsing operation. For LLaMA-2, we also see the benefit of using a special token for instructions `[INST]`. We also share the results reported in (Moosavi et al., 2021) over the SciGen dataset for T5 models (Roberts et al., 2019) in our result tables denoted with an asterisk (*) to benchmark our approach.

Intermediate task pre-training Our main goal is interpreting scientific tables to incorporate the learned representations into scientific language models and achieve better results over scientific literature tasks through a more comprehensive understanding of scholarly articles. As an initial experiment, we analyze the capabilities of the chosen LLMs on the SciGen test dataset and report results in Table 1 as a baseline to improve upon during intermediate task pre-training and fine-tuning. This test dataset includes further two subsets focusing on publications from Computational and Linguistics (Test C&L in Table 2) fields and a wide range of subfields of computer science (Test Other). The qualitative examination of generated texts from pre-trained language models (red-coloured zero-shot example in Figure 3) concludes that the models are not capable of understanding table structure represented with tokens `<R>` and `<C>`.

To address this first challenge, we employ an intermediate task pre-training approach, similarly (Eisenschlos et al., 2020). We use WikiTQ and SQA datasets to pre-train language models before fine-tuning them on scientific articles in the SciGen dataset. This intermediate step helps the language models to (1) capture the semantic relationships in the tables via our special tokens to represent them, (2) reason over tables to be able to answer questions requires arithmetic operations such as finding the maximum, and minimum values or selecting an answer from a specific cell.

Fine-tuning on scientific tables After the models gain the capability of understanding tables, we move to the next step to obtain specialized language models for scientific tables. At this stage, we utilize the large training dataset under SciGen. We use the provided "text" for each table as a reference and we expect the fine-tuned language model to produce similar text for a given table for the prompt of "Explain the given table". Further implementation details are given in Appendix A. Figure 1 depicts the end-to-end training methodology explained in this section.

Evaluation metrics Following the evaluations in previous work on SciGen (Moosavi et al., 2021), we use a subset of their metrics in our evaluation such as METEOR (Denkowski and Lavie, 2014), and BertScore (BertS) (Zhang et al., 2019). Considering our generations for scientific tables are expected to be similar to the reference text, we

Setting	Model	METEOR	ROUGE-1	BertS
		Test (C&L)		
SciGen-Large	T5-base*	0.13(+0.11)	n/a	0.79(+0.06)
	T5-large*	0.16(+0.12)	n/a	0.81(+0.07)
	FlanT5-small	0.04(-0.02)	0.05(-0.04)	0.82(+0.03)
	FlanT5-base	0.03(-0.01)	0.07(+0.01)	0.82(+0.08)
	FlanT5-large	0.08(-0.02)	0.14(+0.02)	0.79
	FlanT5-xl	0.14(+0.06)	0.23(+0.13)	0.85(+0.07)
	LLaMA2-7B-chat-hf	0.15(+0.07)	0.17(+0.10)	0.78(+0.08)
WikiTQ	FlanT5-xl	0.08	0.12(+0.02)	0.81(+0.03)
WikiTQ + SQA	FlanT5-xl	0.08	0.10	0.79(+0.01)
WikiTQ + SciGen	FlanT5-xl	0.14(+0.06)	0.23(+0.13)	0.85(+0.07)
WikiTQ + SQA + SciGen	FlanT5-xl	0.15(+0.07)	0.24(+0.14)	0.85(+0.07)
		Test (Other)		
SciGen-Large	T5-base*	0.13(+0.10)	n/a	0.79(+0.05)
	T5-large*	0.16(+0.11)	n/a	0.81(+0.06)
	FlanT5-small	0.03(-0.03)	0.04(-0.02)	0.82(+0.03)
	FlanT5-base	0.03(-0.02)	0.07	0.82(+0.08)
	FlanT5-large	0.07(-0.04)	0.12	0.77(-0.02)
	FlanT5-xl	0.13(+0.05)	0.23(+0.14)	0.85(+0.07)
	LLaMA2-7B-chat-hf	0.15(+0.07)	0.17(+0.10)	0.78(+0.08)
WikiTQ	FlanT5-xl	0.07(-0.01)	0.10(+0.01)	0.81(+0.03)
WikiTQ + SQA	FlanT5-xl	0.08	0.09	0.79(+0.01)
WikiTQ + SciGen	FlanT5-xl	0.13(+0.05)	0.23(+0.14)	0.85(+0.07)
WikiTQ + SQA + SciGen	FlanT5-xl	0.14(+0.06)	0.23(+0.14)	0.85(+0.07)

Table 2: The change in the scores compared to before applying the corresponding settings for each model is given in the parenthesis. We obtain the best results after applying intermediate task pre-training on WikiTQ and SQA to improve the reasoning capability of the model and subsequent fine-tuning on SciGen to adapt scientific table understanding.

also add the ROUGE (Lin, 2004) score into our evaluation metrics set.

4 Understanding Scientific Tables

4.1 Zero-shot Evaluation

We share the evaluation of models in zero-shot task (without fine-tuning or intermediate task pre-training) results in Table 1. These results serve as a baseline for the comparison in Table 2. We can see that while LLaMA-2 is the largest model (with 7B parameters) in the table, the ROUGE and BERTScore of this model are lower than the FlanT5-large (0.78B) and FlanT5-xl (3B). Considering the increasing computational demand in training associated with larger model sizes, we chose to use FlanT5-xl in our detailed experiments under different settings. It is worth noting that, the fine-tuning of SciGen dataset results in Table 2 demonstrates substantial improvement for LLaMA-2. This indicates that LLaMA-2 could also be adapted to scientific tables, although not with the same capacity as encoder-decoder architecture-based models.

The zero-shot evaluation results for TableLlama Zhang et al. (2023b) further validate the importance of adaptation to the scientific domain since being trained on a large training dataset and a variety of tasks, its performance on SciGen is still close to other language models which are not trained on table specific tasks.

4.2 Quantitative results

In Table 2, we report the scores obtained in each setting and model. When we compare the scores by model size, we can see that the largest model of FlanT5 family (Chung et al., 2022), FlanT5-xl (3B parameters) achieves the best scores on both test datasets introduced in (Moosavi et al., 2021). The scores of METEOR, ROUGE-1 and BertS also

Model	FNR	Precision	f1-score	Accuracy	FPR
SVM	08.00% (±0.48%)	94.77% (±0.25%)	93.34% (±0.18%)	93.44% (±0.16%)	05.07%
XGBoost	10.12% (±0.55%)	90.74% (±0.39%)	90.30% (±0.36%)	90.35% (±0.35%)	09.17%
DNN	24.40% (±3.53%)	90.40% (±1.73%)	82.23% (±1.45%)	83.72% (±0.89%)	08.13%
RF	13.07% (±0.35%)	92.73% (±0.32%)	89.73% (±0.22%)	90.05% (±0.21%)	06.82%
K-NN	7.45% (±0.48%)	93.36% (±0.70%)	91.38% (±0.55%)	91.28% (±0.57%)	10.00%
ELM	16.50% (±0.29%)	92.25% (±0.67%)	87.66% (±0.30%)	88.24% (±0.36%)	07.00%
Malytics	**05.53%** (±0.46%)	**95.88%** (±0.40%)	**95.17%** (±0.20%)	**95.20%** (±0.20%)	**04.06%**
[15]	11.60% (±2.76%)	88.16% (±1.8%)	– –	88.26% (±1.73%)	–

Reference:
Table IV provides more inside into Malytics. Table IV shows the average FNR, precision, f1-score, accuracy and FPR of Malytics with our proposed feature representation. The main important index of the test is FNR as a measure that shows how well Malytics detected new families.

Zero:
R> C> [EMPTY] C> (0. 18%) C> (0. 16%) C> (0. 15%) C> [EMPTY] C> (0. 55%) C> (0. 39%) C> (0. 36%) C> (0. 35%) C> [EMPTY] C> C> [EMPTY] C> (0. 35%) C> [EMPTY] C> C> DNN C> 24. 40% C> 90. 40% C> 82. 23% C> 83. 72% C> 08. 13% R> C> [EMPTY] C> (3. 53%) C> (1. 73%) C> (0. 89%) C> [EMPTY] C> C> RF C> 13. 07% C> 92. 73% C> 89. 73% C> 90. 05% C> 06. ... C> [EMPTY].

SciGen:
We also compare the performance of the proposed method with the state-of-the-art methods. We observe that the proposed method outperforms the state-of-the-art methods in all the metrics. The performance of the proposed method is comparable to the state-of-the-art methods.

WikiTQ+SciGen:
We can see that the K-NN model outperforms the other models in all the metrics. The performance of the K-NN model is comparable to the performance of the SVM model. The performance of the XGBoost model is comparable to the performance of the DNN model. The performance of the RF model is comparable to the performance of the RF model. The performance of the K-NN model is comparable to the performance of the RF model.

WikiTQ+SQA+SciGen:
We can see that the proposed model outperforms all the baselines in terms of FNR, precision, f1-score, accuracy, and FPR. The performance of the proposed model is comparable to the baselines, which shows that the proposed model is able to learn a good representation of the data.

Figure 3: Generations from different models for the sample table. The original caption of the table is “The Mean and Std of Malytics and the baselines for Dex Share Dataset”. We find the green-coloured generation of our proposed approach (WikiTQ+SQA+SciGen) is a more descriptive and helpful summary for literature understanding.

show an increasing trend by model size.

Considering the FlanT5-xl is the most prominent model in our benchmarking set, we conduct the intermediate task pre-training step using different datasets with this model. During the intermediate task pre-training, we use WikiTQ and SQA datasets introduced in Section 3.1. The largest increase in the scores happens when we use the SciGen dataset. It is an expected result since this dataset focuses on scientific tables and has a structure similar to the test datasets. We also see the benefit of WikiTQ in intermediate task pre-training with the increase in scores. Interestingly, when we move to the intermediate task pre-training on SQA after WikiTQ, the scores do not improve. We hypothesize that the difference between the structure of scientific tables and table QA tasks becomes more pronounced after two subsequent pre-training on table QA datasets without fine-tuning on scientific tables. As a final step of our training, we fine-tune the model, which is trained on both WikiTQ and SQA previously, on scientific tables which achieves the highest scores in our comparison of different settings. Consequently, our experiments demonstrate the essential advantage of leveraging intermediate task pre-training on table QA datasets, substantially improving LLMs' understanding and analysis of scientific tables.

4.3 Qualitative results

We share examples generated under different settings in Figure 3 for a sample table in the Test (Other) dataset, taken from (Yousefi-Azar et al., 2018). The table structure is encoded in the model input by using the tokens mentioned in Section 3.2. The dataset includes a reference text that assists us in quantifying the quality of our generations. The colourful texts in Figure 3 are the generations of the models. The red-coloured text is generated by FlanT5-xl without applying any intermediate task pre-training or fine-tuning. The generated text is non-sensible and indicates the model needs to adapt our table structure to understand the given information and produce a coherent text.

The generation of SciGen further demonstrates our motivation for intermediate task pre-training. Even though the generation is relatively high quality compared to the zero setting and factually correct for the given table sample, it is too generic and it is hard to extract tangible information using this generation. Thus, we find this kind of generation is not helpful for scientific literature understanding tasks. Comparing the SciGen generation, WikiTQ+SciGen output seems to contain more concrete information, however, some of the generated information is not factually correct when checking the table. Finally, the green-coloured generation is produced by the model pre-trained on WikiTQ and SQA, and fine-tuned on SciGen. We see the improvement in the generation quality as the output

Setting	Data	MSE	F1-score
Zero	Title + Abstract + Introduction + Table captions	6.54	0.14
	Title + Abstract + Introduction + Table representations	5.60	0.17
SciGen	Title + Abstract + Introduction + Table captions	3.02	0.24
	Title + Abstract + Introduction + Table representations	2.63	0.30
WikiTQ	Title + Abstract + Introduction + Table captions	5.49	0.23
	Title + Abstract + Introduction + Table representations	5.21	0.23
WikiTQ+SciGen	Title + Abstract + Introduction + Table captions	3.11	0.16
	Title + Abstract + Introduction + Table representations	6.05	0.24
WikiTQ+SQA+SciGen	Title + Abstract + Introduction + Table captions	2.61	0.28
	Title + Abstract + Introduction + Table representations	**2.30**	**0.38**

Table 3: Peer-review score prediction results using FlanT5-xl under different training settings. The model is evaluated on a subset of the PeerRead dataset, with embeddings generated from the title, abstract, introduction, and table captions or representations. The best results are obtained when the model is pre-trained on the WikiTQ and SQA datasets, followed by fine-tuning on the SciGen dataset (WikiTQ+SQA+SciGen setting). This demonstrates the promising potential of improved table understanding for scholarly document-based tasks.

is more concise, factually correct, and closer to the given reference. This conclusion aligns with our quantitative analysis findings in Section 4.2.

5 Peer Review Score Prediction

5.1 Experiments

To demonstrate the potential benefit of learning representations from tabular data in scientific articles, we incorporate tables into the peer-review score prediction task. We use the intersection of PeerRead (Kang et al., 2018) and SciGen (Moosavi et al., 2021) datasets, 55 publications across ICLR 2017, ACL 2017, and CoNLL 2016 as source data. Utilizing the entire content of the publications for peer-review prediction is impractical due to the context window length limitation of language models. Thus, previous approaches develop peer review predictive models using metadata, abstract and introduction sections of the paper (Singh et al., 2023).

In this section, we conduct experiments with table captions or representations generated by the models in addition to the title, abstract, and introduction to evaluate how table comprehension influences the accuracy of peer review score predictions, aligning them more closely with human reviewers' evaluations. We employ FlanT5-xl, identified in Section 4 as the most effective model, to create summaries of the tables. The summaries' embeddings serve as input for the prediction model, and we use XGBoost (Chen and Guestrin, 2016) for regression and classification. We then predict recommendation scores using embeddings from the FlanT5-xl model, which is fine-tuned under different settings. We evaluate our predictions using Mean Squared Error (MSE) and F1-score, defined as follows:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \qquad (1)$$

where n is the number of samples, y_i is the true peer-review score, and $\hat{y}_i$ is the predicted value.

$$\text{F1-score} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \qquad (2)$$

where $precision = \frac{\text{TP}}{\text{TP+FP}}$ and $recall = \frac{\text{TP}}{\text{TP+FN}}$, with TP being true positives, FP being false positives, and FN being false negatives. We share our findings in Table 3.

5.2 Results

The MSE and F1-scores in Table 3 show improvements in all settings when the table representations are used in peer-review prediction, except WikiTQ+SciGen. This finding validates our hypothesis that scientific language models could benefit from learning tabular data to fully interpret scientific literature. We only see a drop in the MSE score of WikiTQ+SciGen generations. We suspect the model in this setting hallucinates more as in the given sample Figure 3 and it misleads the XGBoost algorithm in peer review score prediction. Lastly, we obtain the lowest MSE and highest F1-score using the embeddings from the WikiTQ+SQA+SciGen setting. This conclusion reinforces the findings in Section 4 and shows the effectiveness of our proposed approach.

6 Conclusions

Scientific language model development and document comprehension have accelerated progress in recent years parallel to advancements in large language models. However, their ability to effectively understand and reason over tabular data in scientific literature has remained under-explored. In this work, we addressed this issue by proposing an approach that combines intermediate task pre-training on table question-answering datasets with model adaptation to comprehend tables in computer science literature.

Our experiments demonstrated that by understanding tables better, LLMs can achieve higher performance in scientific literature understanding tasks. We validated this claim through a case study on peer-review score prediction, where our best-performing model, pre-trained on WikiTQ and SQA datasets and fine-tuned on the SciGen dataset, outperformed other settings in terms of mean squared error and F1-score. These results emphasize the importance of integrating tabular data into the training process of scientific language models.

Moreover, our qualitative analysis showed that the proposed approach generates more informative and contextually relevant summaries of scientific tables compared to generalist table models and models without intermediate task pre-training or fine-tuning. This finding suggests that our method can enhance the comprehension of scientific literature by providing more accurate and descriptive table representations. Future research directions could include extending our approach to other scientific domains, exploring the integration of table representations with other elements of scientific papers (e.g., figures and equations), and developing more sophisticated table encoding techniques. Additionally, incorporating larger and more diverse datasets for pre-training and fine-tuning could further improve the performance of LLMs on scientific literature tasks.

References

Zhangir Azerbayev, Hailey Schoelkopf, Keiran Paster, Marco Dos Santos, Stephen Marcus McAleer, Albert Q. Jiang, Jia Deng, Stella Biderman, and Sean Welleck. 2024. Llemma: An open language model for mathematics. In *The Twelfth International Conference on Learning Representations*.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. 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 3615–3620.

Prabhat Kumar Bharti, Shashi Ranjan, Tirthankar Ghosal, Mayank Agrawal, and Asif Ekbal. 2021. Peerassist: leveraging on paper-review interactions to predict peer review decisions. In *Towards Open and Trustworthy Digital Societies: 23rd International Conference on Asia-Pacific Digital Libraries, ICADL 2021, Virtual Event, December 1–3, 2021, Proceedings 23*, pages 421–435. Springer.

Steven Bird, Robert Dale, Bonnie Dorr, Bryan Gibson, Mark Joseph, Min-Yen Kan, Dongwon Lee, Brett Powley, Dragomir Radev, and Yee Fan Tan. 2008. The ACL Anthology reference corpus: A reference dataset for bibliographic research in computational linguistics. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794.

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. 2020. TabFact: A large-scale dataset for table-based fact verification. In *International Conference on Learning Representations*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Arman Cohan, Waleed Ammar, Madeleine van Zuylen, and Field Cady. 2019. Structural scaffolds for citation intent classification in scientific publications. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3586–3596, Minneapolis, Minnesota. Association for Computational Linguistics.

Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel Weld. 2020. SPECTER: Document-level representation learning using citation-informed transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2270–2282, Online. Association for Computational Linguistics.

Michael Denkowski and Alon Lavie. 2014. Meteor Universal: Language specific translation evaluation for any target language. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 376–380, Baltimore, Maryland, USA. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Julian Eisenschlos, Syrine Krichene, and Thomas Müller. 2020. Understanding tables with intermediate pre-training. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 281–296, Online. Association for Computational Linguistics.

Guglielmo Faggioli, Thibault Formal, Simon Lupart, Stefano Marchesin, Stephane Clinchant, Nicola Ferro, and Benjamin Piwowarski. 2023. Towards query performance prediction for neural information retrieval: Challenges and opportunities. In *Proceedings of the 2023 ACM SIGIR International Conference on Theory of Information Retrieval*, ICTIR '23, page 51–63, New York, NY, USA. Association for Computing Machinery.

Yang Gao, Steffen Eger, Ilia Kuznetsov, Iryna Gurevych, and Yusuke Miyao. 2019. Does my rebuttal matter? insights from a major NLP conference. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1274–1290, Minneapolis, Minnesota. Association for Computational Linguistics.

Tirthankar Ghosal, Rajeev Verma, Asif Ekbal, and Pushpak Bhattacharyya. 2019. DeepSentiPeer: Harnessing sentiment in review texts to recommend peer review decisions. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1120–1130, Florence, Italy. Association for Computational Linguistics.

Kelvin Guu, Panupong Pasupat, Evan Liu, and Percy Liang. 2017. From language to programs: Bridging reinforcement learning and maximum marginal likelihood. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1051–1062, Vancouver, Canada. Association for Computational Linguistics.

Stefan Hegselmann, Alejandro Buendia, Hunter Lang, Monica Agrawal, Xiaoyi Jiang, and David Sontag. 2023. Tabllm: Few-shot classification of tabular data with large language models. In *International Conference on Artificial Intelligence and Statistics*, pages 5549–5581. PMLR.

Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Eisenschlos. 2020. TaPas: Weakly supervised table parsing via pre-training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4320–4333, Online. Association for Computational Linguistics.

Mohit Iyyer, Wen-tau Yih, and Ming-Wei Chang. 2017. Search-based neural structured learning for sequential question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1821–1831, Vancouver, Canada. Association for Computational Linguistics.

Anshul Kanakia, Zhihong Shen, Darrin Eide, and Kuansan Wang. 2019. A scalable hybrid research paper recommender system for Microsoft Academic. In *The world wide web conference*, pages 2893–2899.

Dongyeop Kang, Waleed Ammar, Bhavana Dalvi, Madeleine van Zuylen, Sebastian Kohlmeier, Eduard Hovy, and Roy Schwartz. 2018. A dataset of peer reviews (PeerRead): Collection, insights and NLP applications. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1647–1661, New Orleans, Louisiana. Association for Computational Linguistics.

J-D Kim, Tomoko Ohta, Yuka Tateisi, and Jun'ichi Tsujii. 2003. Genia corpus—a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(suppl_1):i180–i182.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Jakub Lála, Odhran O'Donoghue, Aleksandar Shtedritski, Sam Cox, Samuel G Rodriques, and Andrew D White. 2023. PaperQA: Retrieval-augmented generative agent for scientific research. *arXiv preprint arXiv:2312.07559*.

Jiao Li, Yueping Sun, Robin J Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J Mattingly, Thomas C Wiegers, and Zhiyong Lu. 2016. Biocreative v cdr task corpus: a resource for chemical disease relation extraction. *Database*, 2016.

Yuhan Li, Jian Wu, Zhiwei Yu, Börje F. Karlsson, Wei Shen, Manabu Okumura, and Chin-Yew Lin. 2023. All data on the table: Novel dataset and benchmark for cross-modality scientific information extraction.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel Weld. 2020. S2ORC: The semantic scholar open research corpus. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4969–4983, Online. Association for Computational Linguistics.

Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3219–3232, Brussels, Belgium. Association for Computational Linguistics.

Renqian Luo, Liai Sun, Yingce Xia, Tao Qin, Sheng Zhang, Hoifung Poon, and Tie-Yan Liu. 2022. BioGPT: generative pre-trained transformer for biomedical text generation and mining. *Briefings in Bioinformatics*, 23(6):bbac409.

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)*.

Panitan Muangkammuen, Fumiyo Fukumoto, Jiyi Li, and Yoshimi Suzuki. 2023. Intermediate-task transfer learning for peer review score prediction. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 40–47, Nusa Dua, Bali. Association for Computational Linguistics.

Malte Ostendorff, Nils Rethmeier, Isabelle Augenstein, Bela Gipp, and Georg Rehm. 2022. Neighborhood contrastive learning for scientific document representations with citation embeddings. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11670–11688, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Panupong Pasupat and Percy Liang. 2015. Compositional semantic parsing on semi-structured tables. 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)*, pages 1470–1480, Beijing, China. Association for Computational Linguistics.

Adam Roberts, Colin Raffel, Katherine Lee, Michael Matena, Noam Shazeer, Peter J. Liu, Sharan Narang, Wei Li, and Yanqi Zhou. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. Technical report, Google.

Anna Rogers and Isabelle Augenstein. 2020. What can we do to improve peer review in NLP? In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1256–1262, Online. Association for Computational Linguistics.

Nihar B Shah. 2022. Challenges, experiments, and computational solutions in peer review. *Communications of the ACM*, 65(6):76–87.

Shinichi Shirakawa, Yasushi Iwata, and Youhei Akimoto. 2018. Dynamic optimization of neural network structures using probabilistic modeling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Amanpreet Singh, Mike D'Arcy, Arman Cohan, Doug Downey, and Sergey Feldman. 2023. SciRepEval: A multi-format benchmark for scientific document representations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5548–5566, Singapore. Association for Computational Linguistics.

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*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or fiction: Verifying scientific claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7534–7550, Online. Association for Computational Linguistics.

Yu-Siang Wang, Yen-Ling Kuo, and Boris Katz. 2020. Investigating the decoders of maximum likelihood sequence models: A look-ahead approach. *arXiv preprint arXiv:2003.03716*.

Mahmood Yousefi-Azar, Leonard G. C. Hamey, Vijay Varadharajan, and Shiping Chen. 2018. Malytics: A malware detection scheme. *IEEE Access*, 6:49418–49431.

Kai Zhang, Jun Yu, Zhiling Yan, Yixin Liu, Eashan Adhikarla, Sunyang Fu, Xun Chen, Chen Chen, Yuyin Zhou, Xiang Li, et al. 2023a. Biomedgpt: A unified and generalist biomedical generative pre-trained transformer for vision, language, and multimodal tasks. *arXiv preprint arXiv:2305.17100*.

Tianshu Zhang, Xiang Yue, Yifei Li, and Huan Sun. 2023b. Tablellama: Towards open large generalist models for tables. *arXiv preprint arXiv:2311.09206*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Yu Zhang, Hao Cheng, Zhihong Shen, Xiaodong Liu, Ye-Yi Wang, and Jianfeng Gao. 2023c. Pre-training multi-task contrastive learning models for scientific literature understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12259–12275, Singapore. Association for Computational Linguistics.

Yu Zhang, Bowen Jin, Qi Zhu, Yu Meng, and Jiawei Han. 2023d. The effect of metadata on scientific literature tagging: A cross-field cross-model study. In *Proceedings of the ACM Web Conference 2023*, pages 1626–1637.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

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 Implementation Details

We use cross-entropy as a loss function, Adam (Kingma and Ba, 2015) as an optimizer with a fixed learning rate of 1e-6 in all iterations of intermediate task pre-training and fine-tuning. We experiment with larger learning rates but the best results are obtained with 1e-6. We train our models with an early-stopping approach with a maximum of 5 epochs using an A100 GPU for FlanT5 variants and 3 A100 GPUs for LLaMA-2-chat-hf. While tokenizing the tables, the maximum length is chosen as 512. The batch size for FlanT5-xl is 2 and LLaMA-2 is 1. Our longest training takes 30 hours for the full pipeline with the WikiTQ+SQA+SciGen setting for FlanT5-xl.

B Additional Examples

We share more examples of different samples from the test dataset in this section.

random	beam	ALCHEMY 3utts	ALCHEMY 5utts	TANGRAMS 3utts	TANGRAMS 5utts	SCENE 3utts	SCENE 5utts
		classic beam search					
None	32	30.3	23.2	0.0	0.0	33.4	20.1
None	128	59.0	46.4	60.9	28.6	24.5	13.9
		randomized beam search					
$\epsilon = 0.05$	32	58.7	45.5	61.1	32.5	33.4	23.0
$\epsilon = 0.15$	32	**61.3**	48.3	**65.2**	**34.3**	50.8	33.5
$\epsilon = 0.25$	32	60.5	**48.6**	60.0	27.3	**54.1**	**35.7**

Reference:
Table 3 shows that (cid:15)-greedy randomized beam search consistently outperforms classic beam search. Even when we increase the beam size of classic beam search to 128, it still does not surpass randomized beam search with a beam of 32, and further increases yield no additional improvement.

Zero:
R> C> None C> 32 C> 30. 3 C> 23. 2 C> 0. 0 C> 0. 0 C> 33. 4 C> 20. 1 R> C> None C> 128 C> 59. 0 C> 46. 4 C> 60. 9 C> 28. 6 C> 24. 5 C> 13. 9 R> C> [BOLD] randomized beam search C> [BOLD] randomized beam search C> [BOLD] randomized beam search C> [BOLD] randomized beam search C> [B C> ...

SciGen:
We also compare the performance of randomized beam search with classic beam search. We observe that randomized beam search outperforms classic beam search on all the datasets. This is because the randomized beam search is more flexible and can be used to search for more than one beam.

WikiTQ+SciGen:
We also compare the performance of our randomized beam search with classic beam search. We observe that the randomized beam search outperforms the classic beam search in all cases. This is because the randomized beam search is able to learn a more diverse set of features.

WikiTQ+SQA+SciGen:
We also compare the performance of randomized beam search with classic beam search. We observe that randomized beam search outperforms classic beam search on all datasets. This is expected since the randomized beam search is based on a random search of the beams.

Figure 4: The original caption of the table is "Randomized beam search. All listed models use gradient weight qMML and Tokens to represent execution history.". Taken from (Guu et al., 2017).

	Test error (%)	Time (hour)
AdaptiveNet	1.645 (0.072)	1.01
BO (budget=10)	1.780	9.59
BO (budget=20)	1.490	18.29

Reference:
Table 3 shows that the test errors of the stochastic networks obtained by the proposed method and the Bayesian optimization with different budgets, where budget indicates the number of hyper-parameters to be evaluated. we observe that the computational time of the Bayesian optimization proportionally increases for the number of budgets while our method is more computationally efficient. The proposed method can find a competitive stochastic network with reasonable computational time. The Bayesian optimization could find a better configuration in this case within several ten budgets.

Zero:
R> C> [EMPTY] C> Test error (%) C> Time (hour) R> C> AdaptiveNet C> 1. 645 (0. 072) C> 1. 01 R> C> BO (budget=10) C> 1. 780 C> 9. 59 R> C> BO (budget=20) C> 1. 490 C> 18. 29.

SciGen:
The proposed method is able to achieve the best performance in terms of test error and computational time. The results show that the proposed method is able to achieve the best performance in terms of test error and computational time.

WikiTQ+SciGen:
The proposed method is able to achieve better performance than the Bayesian optimization with different budgets.

WikiTQ+SQA+SciGen:
The proposed method is able to achieve a better performance than the Bayesian optimization with different budgets. The results show that the proposed method is able to adapt the stochastic network with a lower computational time.

Figure 5: The original caption of the table is "Test errors (%) and computational time of the proposed method (AdaptiveNet) and the Bayesian optimization (BO) with different budgets in the experiment of adaptation of stochastic network. The mean values over 30 trials are reported in the proposed method, and the value in parentheses denotes the standard deviation. For the Bayesian optimization, the result of a single run is reported.". Taken from (Shirakawa et al., 2018).

Search Strategy	BLEU
Greedy Search	86.24
2-LA	86.65
3-LA	86.71
4-LA	86.77
5-LA	**86.79**
Beam Search (B=10)	86.28

Reference:
Table 1: The performances of the IM2LATEX-100K Bi-LSTM model. We discover that the look-ahead improves the model from the greedy search method – noted that LA is more directly comparable to the greedy search because of their same beam size. We also show the scores of the beam search for the reference.

Zero:
R> C> Search Strategy C> BLEU R> C> Greedy Search C> 86. 24 R> C> 2-LA C> 86. 65 R> C> 3-LA C> 86. 71 R> C> 4-LA C> 86. 77 R> C> 5-LA C> [BOLD] 86. 79 R> C> Beam Search (B=10) C> 86. 28 CAP>.

WikiTQ+SciGen:
We also compare the performance of the beam search with the greedy search. The results show that the look-ahead improves the model from the greedy search.

SciGen:
We also compare the performance of the beam search with the greedy search. The results show that the look-ahead improves the model from the greedy search.

WikiTQ+SQA+SciGen:
We also compare the performance of the beam search with the greedy search. The results show that the look-ahead improves the model from the greedy search method. The 4-LA and 5-LA are more directly comparable to the greedy search because of their same beam size.

Figure 6: The original caption of the table is "The performances of the IM2LATEX-100K Bi-LSTM model. We discover that the look-ahead improves the model from the greedy search method—noted that LA is more directly comparable to the greedy search because of their same beam size. We also show the scores of the beam search for the reference". Taken from (Wang et al., 2020).