All Data on the Table: Novel Dataset and Benchmark for Cross-Modality Scientific Information Extraction

Anonymous ACL submission

Abstract

Extracting key information from scientific papers has the potential to help researchers work more efficiently and accelerate the pace of scientific progress. Over the last few years, research on Scientific Information Extraction (SciIE) witnessed the release of several new systems and benchmarks. However, existing paper-focused datasets mostly focus only on specific parts of a manuscript (e.g., abstracts) and are single-modality (i.e., text- or table-only), due to complex processing and expensive annotations. Moreover, core information can be present in either text or tables or across both. To close this gap in data availability and enable cross-modality IE, while alleviating labeling costs, we propose a semi-supervised pipeline for annotating entities in text, as well as entities and relations in tables, in an iterative procedure. Based on this pipeline, we release SciCM, a high-quality scientific cross-modality IE benchmark, with a large-scale corpus and a semi-supervised annotation pipeline. We further report the performance of state-of-the-art IE models on the proposed benchmark dataset, as a baseline[†]. Lastly, we explore the potential capability of large language models such as ChatGPT for the current task. Our new dataset, results, and analysis validate the effectiveness and efficiency of our semi-supervised pipeline, and we discuss its remaining limitations.

1 Introduction

As scientific communities grow and evolve, there has been an explosion in the number of scientific papers being published in recent years[1], which makes it increasingly difficult for researchers to discover useful insights and new techniques in the vast amount of information. One approach to help researchers keep abreast with the latest scientific advances and quickly identify new challenges and opportunities is to automatically extract and organize crucial scientific information from collections of research publications (Viswanathan et al., 2021). Scientific information extraction (SciIE) aims to extract such information from scientific literature corpora and has seen growing interest recently, with the rapid evolution of systems and benchmarks (Jain et al., 2020; Zhuang et al., 2022).

SciIE serves as an important pre-processing step for many downstream tasks, including scientific knowledge graph construction (Wang et al., 2021), academic question answering (Dasigi et al., 2021), and method recommendation (Luan, 2018). Additionally, the development of scientific large language models (LLMs), such as Galactica (Taylor et al., 2022) and Mozi (Lan et al., 2023), allows the exploration of several practical science scenarios (e.g., suggest citations, ask scientific questions, and write scientific code). However, language models can hallucinate without verification. Moreover, language models are frequency-biased and often overconfident. SciIE along with appropriate QA or retrieval systems - e.g., TIARA (Shu et al., 2022) - can help alleviate such problems and facilitate model performance on downstream tasks, as similarly demonstrated in leveraging Wikidata to improve LLM factuality (Xu et al., 2023).

Initial corpora and benchmarks extracted information from specific parts of a paper text, such as abstracts (Gábor et al., 2018; Luan et al., 2018) or selected paragraphs (Augenstein et al., 2017; D'Souza and Auer, 2020; Hou et al., 2021). However, scientific entities are spread through the whole paper body; thus neglecting any text fragments or tables will likely result in missing key information. This is especially true in tables that condense complex information and data on experimental results. Jain et al. (2020) first attempts to create a scientific IE benchmark at the document level and Hou et al. (2019) first annotates entities in tables as well as image captions. Unfortunately,

[†] Our code and data will be available after acceptance.

[1] https://arxiv.org/stats/monthly_submissions

Benchmarks	IE Task	Modality & Coverage	Domain	Size
SemEval-2017 Task 10 (Augenstein et al., 2017)	NER, RE	Text (several paragraphs)	CS, MS, Phy	500
SemEval-2018 Task 7 (Gábor et al., 2018)	RE	Text (abstract)	CS (NLP)	500
SciERC (Luan et al., 2018)	NER, RE	Text (abstract)	CS	500
NLP-TDMS (Hou et al., 2019)	NER, ResE	Text (abstract), Tables	CS (NLP)	332
SciREX (Jain et al., 2020)	NER, N-ary RE	Text (full)	CS (ML)	438
NLPContributions (D'Souza and Auer, 2020)	NER, RE	Text (several paragraphs)	CS (NLP)	50
TDMSci (Hou et al., 2021)	NER	Text (several sentences)	CS (NLP)	2,000
ORKG-TDM (Kabongo et al., 2021)	NER, ResE	Text (several sections), Tables	CS	[5,361]
TELIN (Yang et al., 2022)	NER, ResE	Tables	CS (ML)	731
GASP-NER (Otto et al., 2023)	NER	Text (full)	CS (ML)	100
SciCM (ours)	NER, ResE, RE	Text (full), Tables	CS, Stat, EESS, ...	70 + [12,817]

Table 1: An overview of existing scientific IE benchmarks. Domain acronyms: CS refers to Computer Science; MS refers to Material Science; Phy refers to Physics. Task acronyms: NER refers to Named Entity Recognition; RE refers to Relation Extraction; ResE refers to Result Extraction. "[ ]" indicates automated annotations.

these benchmarks mostly focus on a single modality of paper content and coarser annotations due to the large gap between modalities, plus high processing and annotation costs. Text-only modality covers information presented in an unstructured and narrative way, whereas table-only modality for structured and concise information representation. Nonetheless, it is increasingly clear that information obtained from a single modality will inevitably miss critical information in a given paper, e.g., it is hard to extract experiment results and settings only from text or to extract models and metrics from tables. Combining modalities is thus critical and enables further scenarios like hybrid QA (Wu et al., 2023). However, it is non-trivial to directly extract entities from tables due to the lack of context information and the label imbalance issue. Leveraging entities consistently appearing in both text and tables can intuitively facilitate this process. To address the above-mentioned gaps in data availability, a new cross-modality and document-level benchmark dataset for SciIE is needed.

Annotating such a benchmark remains challenging because 1) it requires domain expertise and considerable annotation effort to comprehensively label a sizeable benchmark, 2) it requires annotators to understand the domain and the whole paper to maintain annotation consistency across different modalities, and 3) annotations need to be fine-grained enough (and consistent across documents) to unlock relevant semantics. To overcome these annotation challenges, we develop a semi-supervised pipeline for annotating entities in text and both entities and relations in tables of academic papers, which involves a two-stage iterative procedure. Specifically, a replaceable extractor is first trained on a small amount of high-quality manually annotated papers and then utilized to label a large number of papers automatically. Experts are introduced next to correct any false labels. This process can be repeated iteratively multiple times, with the extractor becoming more accurate as it can use the newly labeled data to improve its performance. During training, we also adopt label mapping in text extraction via leveraging existing benchmarks (Luan et al., 2018; Jain et al., 2020) to enrich our annotations. We also prove that informing the table extractor with text extraction results leads to more accurate and context-rich table annotations.

Based on this pipeline, we release 1) **SciCM**, a high-quality expert annotated benchmark that supports multiple **Sci**entific **C**ross-**M**odality IE tasks; 2) a large-scale corpus containing automatically annotated papers with different domains; 3) a visualization tool that enables researchers to get a global view of key information in scientific papers; and 4) the pipeline itself, which can be utilized as provided or further extended. We conduct experiments on SciCM utilizing SoTA IE models and popular LLMs such as ChatGPT to showcase its characteristics and as a baseline. To further investigate the efficacy of our pipeline, we assess both the quality and the speed of annotation on SciCM. Additionally, a thorough error analysis is conducted.

We summarize our key contributions as follows: 1) we are the first to explore cross-modality IE to build a bridge between text and tables in scientific articles; 2) we develop a semi-supervised pipeline for annotating scientific terms along with their relations without requesting lots of human supervision at document-level; 3) we release a high-quality benchmark that enables diverse SciIE tasks and a large-scale corpus to facilitate research over the scientific literature; 4) extensive experiments validate the efficiency, effectiveness, and adaptability of our semi-supervised pipeline.

2 Related Work

An overview of existing SciIE benchmarks is shown in Table 1. The field of SciIE began with extracting information from only the text modality. Augenstein et al. (2017) proposed SemEval-2017 Task 10 to support the task of identifying entities and relations in a corpus of 500 paragraphs taken from open-access journals. Gábor et al. (2018) presented SemEval-2018 Task 7 on relation extraction from 500 abstracts of NLP papers. Luan et al. (2018) released SciERC containing 500 abstracts with more fine-grained types of entities (i.e., 6 types) and relations (i.e., 7 types). More recently, D'Souza and Auer (2020) and Hou et al. (2021) proposed benchmarks which are composed of several paragraphs from NLP papers, aiming to capture contributed scientific terms and extract `<Task, Dataset, Metric>` (TDM) triples, respectively. Since scientific terms can appear anywhere in the paper, Jain et al. (2020) proposed SciREX that tries to comprehensively annotate the full paper text. Otto et al. (2023) manually annotated GASP-NER for identifying named entities associated with the interplay between machine learning model entities and dataset entities. Inspired by them, we consider document-level extraction and define more complete entity types compared with prior works. We also leverage two high-quality benchmarks, i.e., SciERC and SciREX, to boost our text extractor for a subset of entities.

Extracting information from the table modality has attracted much research attention in recent years since they contain relevant structural knowledge that aids extraction. To the best of our knowledge, there are currently only three datasets that explore cross-modality extraction in the scientific domain: NLP-TDMS (Hou et al., 2019), ORKG-TDM (Kabongo et al., 2021), and TELIN (Yang et al., 2022). These datasets associate score entities extracted from result tables with their corresponding TDM triples, but they only concentrate on simple result extraction and ignore many valuable types of entities that tables can provide such as `Model` and `Metric`. Additionally, there has been a lack of benchmarks supporting relation extraction in the table modality, which may lead to the absence of important relationships, as well as table-text relationships. According to these limitations, our goal is to develop a benchmark covering information extraction in both text and table modalities and across them, to close the gap in scientific data availability and facilitate more comprehensive and accurate information extraction.

3 Preliminaries

3.1 Data Collection

Extracting large amounts of tables, body text, and other metadata from scientific articles requires expert knowledge and comprehension of the article, which can be very time-consuming for expert annotation. To collect paper data, we propose a method to automatically download and pre-process LaTeX source files of scientific articles. Leveraging the structure of LaTeX source files, we can easily locate each component of papers such as titles, sections, and tables.

Specifically, we first search and crawl the arXiv LaTeX sources from its official website[2] using its official library[3]. We design a simple parser to extract textual content (including all sections and subsections) and tables from a scientific paper. The titles of sections and sub-sections are retained. Each extracted table contains its caption and all table cells. To ensure that the extracted data is in a readable format for annotations, we utilize two public libraries, e.g., latexml[4] and dashtable[5] to "clean" the extracted data. This process enables us to efficiently collect and pre-process large amounts of scientific papers for annotation and analysis.

3.2 Task Definition

We aim to perform NER on a corpus of scientific papers across both text and table modalities. Unlike previous benchmarks (Augenstein et al., 2017; Luan et al., 2018), we only perform RE on tables. Cross-sentence relations in text can be very challenging to be annotated (Jain et al., 2020) and error prone. Formally, we denote each scientific paper as D, containing a sequence of paragraphs $P = \{p_1, p_2, ..., p_{|P|}\}$ and a sequence of tables $T = \{t_1, t_2, ..., t_{|T|}\}$. Each paragraph p is composed of a sequence of sentences $\{s_1, s_2, ...s_n\}$ and each sentence is composed of a sequence of words $\{w_1, w_2, ..., w_i\}$. Each table t is flattened via inserting separators and concatenated with its caption as a sequence of cells $\{c_{1,1}, c_{1,2}, .., c_{y,z}\}$, where y and z are the numbers of table rows and columns. Each cell is also composed of a sequence of words $\{w_1, w_2, ..., w_j\}$. Formally, the three IE tasks we support are defined as follows.

[2] https://arxiv.org/
[3] https://github.com/lukasschwab/arxiv.py
[4] https://math.nist.gov/~BMiller/LaTeXML/
[5] https://dashtable.readthedocs.io/en/latest

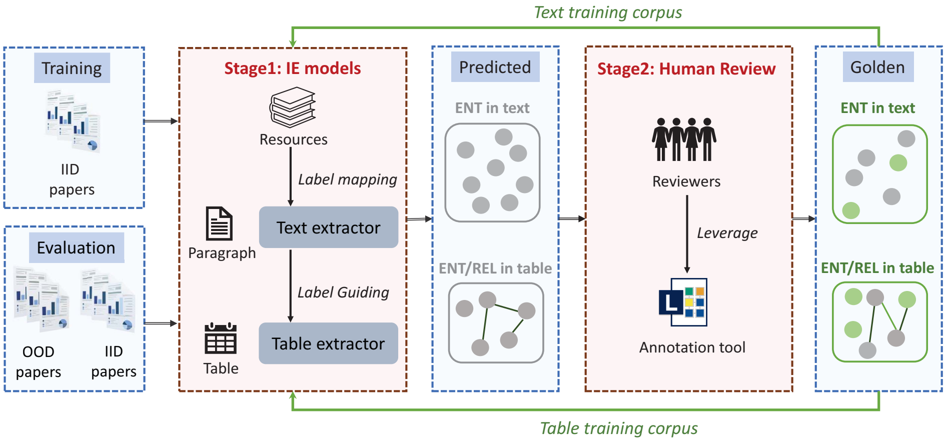

Figure 1: Overview of our semi-supervised pipeline. It consists of two stages: 1) training of text and table extractors through a limited number of manually annotated in-domain papers. Extractors are then employed to automatically annotate a larger set of both out-of-domain and in-domain papers; 2) expert reviewers rectify any false labels to obtain golden annotations, which are used to expand training and update extractors via iterative learning.

Text/Table NER The goal of text/table NER is to examine all possible spans of words, denoted as $\{w_l, ..., w_r\}$ within a sentence/cell, where l and r represent the left and right indices of the span, and recognize if the span describes an entity and classifies it with its type if any.

Table RE The goal of this task is to examine all unordered pairs of entities appearing within a given table, denoted as (e_1, e_2), and determine the existence of any relationship between them.

4 Semi-supervised Annotation Pipeline

4.1 Overview

The overview of our semi-supervised pipeline is depicted in Figure 1, which involves a two-stage iterative process. Specifically, two extractors are trained on a small amount of manually annotated or reviewed papers and leveraged to make automatic annotations on the unlabeled papers (Stage 1). Expert reviewers are introduced later to perform necessary corrections to generate high-quality annotations, which are utilized to update the extractors for iterative training (Stage 2). The pipeline can be repeated multiple times, with extractors becoming more accurate as they have the ability to use the newly labeled data to improve their extraction performance.

Our proposed pipeline exhibits exceptional performance in the annotation of cross-modality entities and relations in the scientific literature. Specifically, it demonstrates the following advantages: i) **Efficiency**. The curation of datasets composed of long scientific documents with both paragraphs and tables can be a costly and labor-intensive task. However, our pipeline only requires a smaller number of labeled papers for training than fully-supervised learning, making it a cost-effective solution in comparison to existing benchmarks. ii) **Effectiveness**. We employ techniques such as label mapping and label guiding for text and tables respectively, to ensure high-quality annotations. iii) **Adaptability**. We conduct both in-domain (IID) and out-of-domain (OOD) evaluations to demonstrate that the pipeline can adapt to new domains with ease, which is particularly useful in the scientific domain where new research is continually being published. The performance of our pipeline will be further investigated in Section 5.

4.2 Text Information Extraction

Entity types We find that existing benchmarks (Luan et al., 2018; Hou et al., 2019; Jain et al., 2020; Hou et al., 2021; Kabongo et al., 2021) normally consider fine-grained knowledge with multiple entity types. However, all these datasets simply regard the "models" proposed by papers as "methods" and do not treat `Model` as a separate type of entity. It is not reasonable, as "models" and "methods" are distinct entities with different characteristics, with "models" being more representative of the paper and more likely to appear in experimental tables. To address this issue, we present more appropriate entity types, including `Task`, `Dataset`, `Metric`, `Model`, and `Method` (see Appendix A.1).

Label mapping Due to the mappable label sets between SciIE benchmarks, we are able to reuse some high-quality labels collected from existing benchmarks to boost the performance of extraction. Inspired by (Yang et al., 2017; Francis et al., 2019), we perform a label mapping step prior to training, using two fully-supervised annotated datasets (Luan et al., 2018; Jain et al., 2020). To be specific, we collect `Task`, `Metric`, and `Method` entities from SciERC (Luan et al., 2018) and `Task`, `Dataset`, `Metric`, and `Method` entities from SciREX (Jain et al., 2020) to enrich our entity annotations.

Extractor We apply the publicly available SciIE model PL-Marker (Ye et al., 2022) to perform text NER. PL-Marker inserts levitated markers in text and incorporate two packing strategies to achieve state-of-the-art F_1 scores and a notable level of efficiency on SciERC (Luan et al., 2018).

4.3 Table Information Extraction

Entity and relation types In contrast to text NER, in table NER we extract two additional types of entities: `Score` and `Setting`. Scores rarely appear in the paper text but frequently appear in tables. We follow previous benchmarks (Hou et al., 2019; Kabongo et al., 2021; Yang et al., 2022) in extracting `Score` entities from tables to enable a more comprehensive understanding of the experimental results reported in scientific papers. `Setting` is another important entity type, as it refers to the context or environment in which the study was conducted. We use `Setting` entities to indicate different experimental settings, for example, BERT_{large} and BERT_{base}, one-hop and multi-hop.

Similar to (Hou et al., 2019; Jain et al., 2020; D'Souza and Auer, 2020; Hou et al., 2021; Kabongo et al., 2021), we do not pre-define specific relation types and instead aim to uncover indirect relationships between pairs of entities. For more details please see Appendix A.2.

Label Guiding Inconsistency in NER predictions between tables and text can result in entities with the same name being assigned different entity types across different modalities. To address this issue, we utilize a label guiding rule that leverages the annotation results of the previous stage, i.e., text NER. Specifically, when an entity appears in both text and tables, we maintain consistency by assigning the same entity type in the table as the entity type identified in the text. This approach ensures that the entity types are consistent across different modalities.

Table IE We treat table IE tasks, which include table NER and table RE, as classification tasks. The model structure of our table IE model is depicted in Figure 2, which comprises an encoding layer and a classification layer.

Prompting the table NER model is straightforward. We prompt the model with a statement that specifies the entity type of a given cell in a table, i.e., "The entity type of the cell $c_{i,j}$, in row i, column j is $[E]$.", where $[E] \in \Gamma$ and Γ is the set of predefined entity types. For each table cell, we generate m prompts and match these prompts with their entity types, where m is the number of entity types. Let $E = \{s_i, t_i, E_i^+, E_{i,1}^-, E_{i,m-1}^-\}_{i=1}^m$ be the training data that consists of m instances. Each instance consists of a statement s_i , a table t_i, a correct (positive) entity type E_i^+ , along with (m-1) wrong (negative) entity types $E_{i,j}^-$.

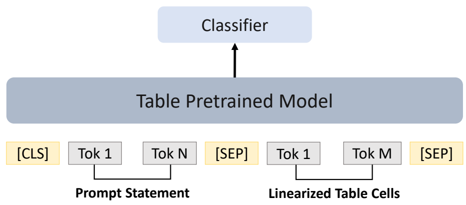

Figure 2: Table IE model architecture. A pre-trained model encodes the input table and its output is fed into a classifier to output the probabilities of entity types for each cell, or the relation probabilities of any two cells.

While TURL (Deng et al., 2020) can solve column relation extraction, it cannot extract relations in fine-grained granularity, such as at the cell level. To address this limitation, we design a solution for table RE, which is a binary classification task that determines whether two cells in a table have a relation or not. The template statement is "Whether the cell $c_{i,j}$, located in row i, column j, has a relation with the cell $c_{p,q}$, located in row p, column q, or not?", where the label is either 0 or 1. We follow the model architecture in Figure 2, where the input table is encoded by a pre-trained table model and a binary classifier is used to output the relation probabilities of any two cells in the same table.

4.4 Visualization

We utilize OneLabeler (Zhang et al., 2022) [6], a tool to aid in the annotation process, enabling reviewers to label data objects of various entities and relations. To incorporate semi-supervised pipeline in the labeling tool, we start with the labeling work-

[6] https://microsoft.github.io/OneLabeler-doc

flow template and add a labeling module, which we configure to be implemented with the built-in tagger of OneLabeler. As a result, we are able to load the automatically extracted entities and relations into the tool, which allows annotators to just remove incorrect detections and create any missing false negative spans/relations, saving time and effort. More details about annotation rules and notes can be found in Appendix A.3.

In Figure 3, we show a portion of the text of the BERT (Devlin et al., 2019) paper, a milestone in the field of NLP with high citations. We annotate various types of entities, such as `Model`, `Task`, and `Method` in the abstract and other sections of the paper using different colors to distinguish between them, which allows annotators to efficiently and accurately label different kinds of entities. Due to space limitations, we provide two additional visualization examples in Appendix B.

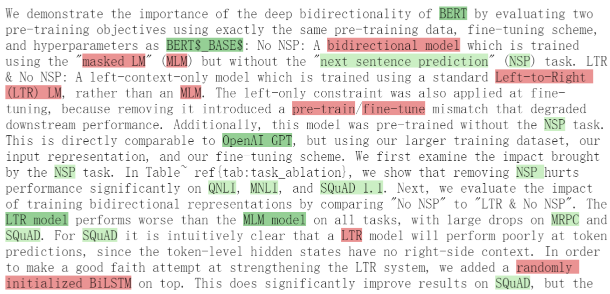
We demonstrate the importance of the deep bidirectionality of BERT by evaluating two pre-training objectives using exactly the same pre-training data, fine-tuning scheme, and hyperparameters as BERT$_BASE$: No NSP: A bidirectional model which is trained using the "masked LM" (MLM) but without the "next sentence prediction" (NSP) task. LTR & No NSP: A left-context-only model which is trained using a standard Left-to-Right (LTR) LM, rather than an MLM. The left-only constraint was also applied at fine-tuning, because removing it introduced a pre-train/fine-tune mismatch that degraded downstream performance. Additionally, this model was pre-trained without the NSP task. This is directly comparable to OpenAI GPT, but using our larger training dataset, our input representation, and our fine-tuning scheme. We first examine the impact brought by the NSP task. In Table~ ref{tab:task_ablation}, we show that removing NSP hurts performance significantly on QNLI, MNLI, and SQuAD 1.1. Next, we evaluate the impact of training bidirectional representations by comparing "No NSP" to "LTR & No NSP". The LTR model performs worse than the MLM model on all tasks, with large drops on MRPC and SQuAD. For SQuAD it is intuitively clear that a LTR model will perform poorly at token predictions, since the token-level hidden states have no right-side context. In order to make a good faith attempt at strengthening the LTR system, we added a randomly initialized BiLSTM on top. This does significantly improve results on SQuAD, but the

Figure 3: Visualization of annotated entities of BERT paper. Different types of entities are assigned distinct colors to facilitate their identification, with dark green representing the `Model`, light green representing the `Task`, and red representing the `Method`.

4.5 Released Benchmark

This section describes the corpora constructed to train and evaluate our semi-supervised pipeline for cross-modality SciIE, as shown in Table 2 and the final dataset (SciCM).

Over 90% of papers with arXiv IDs have source code available, allowing us to directly download and preprocess a large number of papers. For the first round of training, we use 10 manually annotated papers as seeds in the computer science (CS) domain with human-annotated entities and relations. We then add 30 new papers in the same domain and ask domain experts to review the model's predicted results on these papers. We use these human-reviewed 30 papers as added corpus to further train a new version of extractors. We then automatically annotate other 30 papers and review them for both in-domain (IID) and out-of-domain (OOD) evaluation, with 10 papers each from CS, statistics (STAT), and electrical engineering and systems science (EESS), respectively. Finally, we utilize the extractors to automatically annotate 12817 new papers in five different domains to facilitate research in scientific literature. This large-scale annotation of papers enables researchers to efficiently search for relevant information and extract insights from a large corpus of scientific literature.

	Size	Domains	Auto	Reviewed
Seeds	10	CS		✔
Added corpus	30	CS	✔	✔
Test set	30	CS/STAT/EESS	✔	✔
Corpus	12817	CS/STAT/EESS PHYSICS/MATH	✔	

Table 2: Domain distribution and labeling methods of different parts of SciCM.

5 Experiments

5.1 Dataset and Annotation Statistics

We evaluate our semi-supervised pipeline on the test set that includes both IID evaluation and OOD evaluation. Table 4 presents the data statistics of the added corpus and test set, both of which are first automatically annotated and then manually reviewed. It can be seen from the table that scientific tables contain a considerable number of entities and relations, demonstrating the necessity of information extraction from tables. In addition, the long length (i.e., averaging 7000+ words per paper) and cross-modality attributes of scientific papers make full annotation challenging, prompting us to focus on a more efficient annotation pipeline.

5.2 Evaluation Metrics

We follow the standard evaluation protocol (Zhong and Chen, 2021) and use precision, recall, and F_1 as evaluation metrics. For text NER and table NER, both entity boundary and type are required to be correctly predicted. For table RE, the boundaries of the subject entity and the object entity should be correctly identified.

5.3 Implementation Details

Both our text and table IE models are implemented using Pytorch version 1.13 and the Huggingface's Transformers (Wolf et al., 2020) library, running on an A100 GPU. Specifically, we adopt in-domain *scibert-scivocab-uncased* (Beltagy et al., 2019) encoder with 110M parameters for PL-Marker

Test domains	Settings	Text NER			Table NER			Table RE		
		P	R	F_1	P	R	F_1	P	R	F_1
					Round 1 Evaluation					
Overall (IID & OOD)	Fine-tune	67.4	39.5	49.8	49.3	10.6	17.5	4.2	**47.2**	7.7
* CS (IID)	-	67.4	31.9	43.3	51.0	10.6	17.6	5.3	47.2	9.5
* STAT (OOD)	-	71.6	47.7	57.2	55.1	11.0	18.4	2.9	38.3	5.5
* EESS (OOD)	-	60.7	37.9	46.7	38.1	10.0	15.8	5.0	66.7	9.3
					Round 2 Evaluation					
Overall (IID & OOD)	Fine-tune	**79.0**	**75.8**	**77.4**	46.7	**83.0**	**59.8**	**83.1**	28.4	42.3
		+11.6%	*+36.3%*	*+27.6%*	*-2.6%*	*+72.4%*	*+42.3%*	*+78.9%*	*-18.8%*	*+34.6%*
* CS (IID)	-	79.4	65.5	71.8	50.3	83.6	62.8	83.8	21.8	34.6
* STAT (OOD)	-	83.9	87.4	85.6	41.8	76.2	53.9	81.5	33.8	47.7
* EESS (OOD)	-	71.0	72.8	71.9	39.7	80.8	53.1	85.6	35.9	50.6
					ChatGPT Evaluation					
Overall (IID & OOD)	1-shot ICL	24.0	45.2	31.4	53.1	16.8	25.5	73.5	32.8	45.3
Overall (IID & OOD)	2-shot ICL	30.4	38.8	34.1	**57.4**	24.7	34.5	76.2	34.7	**47.7**

Table 3: Evaluating SoTA SciIE models on the test set with different domains. We evaluate in two rounds: *Round 1* that is only trained on a small amount of paper seeds and *Round 2* that is boosted leveraging added corpus. We also report the performance of ChatGPT on the few-shot ICL setting, providing a different number of demonstrations.

Statistics (avg per paper)	**Added corpus**	**Test set**
Sentences	269.9	279.0
Words	6787.7	7872.1
Tables	5.7	3.9
Cells	390.3	190.3
Entities in text	133.2	152.9
Entities in tables	151.6	94.4
Relations in tables	81.3	59.5

Table 4: Statistics of the added corpus and test set.

and *tapas-base* encoder with 110M parameters for our table IE models. During training, we adopt AdamW (Loshchilov and Hutter, 2018) with a learning rate of 2e-5 (5e-5 for tables) and a batch size of 16 (32 for tables). Since scientific papers can be very long, we leverage cross-sentence information (Luan et al., 2019; Ye et al., 2022) to extend each sentence by its neighbor sentences and set the maximum length as 512, which is the input length limit for many transformer-based models.

5.4 Overall Performance

We report the performance of our pipeline in Table 3 with two rounds: *Round 1* has access to only a small set of seeds for training, and *Round 2* extends the training set by incorporating expert-reviewed papers. Our discussion focuses on the effectiveness and adaptability of our pipeline, taking into account the evaluation results. We also provide an analysis of ChatGPT's performance on the benchmark.

Effectiveness study. The experimental results clearly show that the performance of our pipeline in *Round 2* significantly outperforms that of *Round 1* in all three IE tasks, with improvements of 27.6, 42.3, and 34.6 F_1 points, respectively. It is attributed to the availability of more golden annotations for training in the added corpus. Specifically, text NER achieves better results in both rounds compared to the other tasks, as our label mapping step provides a useful and sufficient amount of NER labels, especially when training papers are few. In addition, we could observe that text IE yields better performance than table IE, possibly due to the fact that current table IE models can only rely on structural information and cannot utilize contextual semantics effectively. We discuss the label imbalance issue of table NER in Appendix D.

Adaptability study. In addition to performing IID evaluation, we also report OOD evaluation results based on papers from the STAT and EESS domains. The results indicate that our pipeline is capable of achieving comparable performance (i.e., $\pm$ 13.7, $\pm$ 9.7, $\pm$ 16.0 F_1 points in three tasks) when tested on other domains, demonstrating its ability to adapt to new domains and generalize to unseen data. It is particularly valuable in the scientific domain, where new research is continually being published. Surprisingly, STAT yields the best performance even compared with the CS domain, possibly due to the sparsity of entities and relations in statistics papers, as well as the low-quality labeling by reviewers.

ChatGPT Evaluation. LLMs pre-trained on massive corpora, such as ChatGPT, have demonstrated impressive few-shot learning ability on many NLP tasks. We investigate ChatGPT's capa-

Model	Text NER		Table NER		Table RE	
	M (F_1)	Speed (sent/s)	M (F_1)	Speed (table/s)	M (F_1)	Speed (table/s)
ChatGPT	34.1	18.6	49.8	0.2	**47.7**	0.2
Our pipeline	**77.4**	**49.9**	**59.8**	**33.0**	42.3	**4.6**

Table 5: We compare our pipeline and ChatGPT in both accuracy and annotation speed. The accuracy is measured as the F_1 on the test set. The speed is measured on a single A100 GPU with a batch size of 32.

bilities on our benchmark in terms of the few-shot In-context Learning (ICL) setting. To construct few-shot ICL prompts, we design the prompt template carefully and select demonstrations from the training set. For more details see Appendix E. We use the official API[7] to generate all outputs from ChatGPT. To prevent the influence of dialogue history, we generate the response separately for each testing sample. We compare ChatGPT with our pipeline for all sub-tasks in Table 3. Comparing 2-shot ICL with 1-shot ICL, it can be seen that providing more demonstrations generally leads to improvements (i.e., +2.7, +9.0, +2.4 F_1 points, respectively). We also observe that ChatGPT still struggles to achieve comparable performance with traditional fine-tuned IE models, indicating a need for further exploration in improving the performance of LLMs in cross-modality SciIE.

5.5 Annotation Speed

Table 5 presents the inference speed of our pipeline and ChatGPT. Our automatic labeling approach enables us to process 49.9 sentences per second in text NER and 33.0/4.6 tables per second in table IE. In text NER, our pipeline outperforms ChatGPT by achieving 77.4 F_1 points and a faster processing speed of 49.9 tables/s. Similarly, in table NER, our pipeline achieves a higher F_1 score (59.8) and a faster processing speed of 33.0 tables/s, while ChatGPT achieves a lower F_1 score of 49.8 and a slower processing speed of 0.2 tables/s. However, in table RE, ChatGPT achieves a better F_1 score of 47.7 compared to our table RE model, yet it still has a slower processing speed of 0.2 tables/s.

5.6 Error Analysis

To further explore the limitations of our pipeline, we conduct an error analysis during the auto-annotation process and categorize major errors in both text and table modalities as follows. We also analyze ChatGPT's errors in Appendix C.

[7] https://platform.openai.com/docs

Text IE Errors i) **Over Annotation.** It occurs when terms that are too general to offer useful scientific information are labeled as entities. For instance, "neural network" and "natural language processing" are examples of scientific terms that are mistakenly labeled as `Model` entities. ii) **Missing Abbreviation.** Abbreviations appear frequently in scientific papers to represent entities with long names. Abbreviations are usually missed during auto-annotation. For example, "coupled multi-layer attention" can be recognized successfully, while its abbreviation "CMLA" is missed. iii) **Nested Entity Annotation.** Some entities that contain nested entities, such as "Bi-LSTM-CRF + CNN-char", should be recognized as a complete entity. However, the extractor tends to extract "Bi-LSTM-CRF" and "CNN-char" separately, leading to incomplete annotation. iv) **Inconsistent Annotation.** Same entity is specified with different entity types even if it appears in the same paragraph. v) **Insensitive to Context.** Sometimes an entity will be recognized as different types due to the different contextual environments.

Table IE Errors i) **Entity Type Error.** It occurs when an entity in the table is labeled with the wrong entity type. ii) **Inconsistency in Table Structure.** The structure of a table always provides important context and cues for extraction. Sometimes, there may be obvious inconsistencies in the entity types within the same column or row. For instance, recognizing `Dataset`, `Model`, and `Score` entities in the same column or row. iii) **NER Misleads RE.** It occurs when the NER model incorrectly identifies an entity, leading to incorrect relationship extraction of the current entity. iv) **Missing Relations:** RE model fails to identify a relationship between cells.

6 Conclusion and Future Work

In this paper, we present SciCM, a novel dataset for training and evaluating cross-modality SciIE. Along with it, we propose a semi-supervised pipeline that automatically annotates entities and relations with high effectiveness and efficiency. Our pipeline performs well across SciIE tasks, with good inference speed for iterative runs. Moreover, we release a visualization tool that helps users to annotate scientific items with a global view. In future work, we plan to extend our corpus by incorporating images from scientific papers. We hope our release can facilitate downstream tasks in the scientific domain.

Limitations

In this paper, our focus was on proposing a benchmark that includes paper text and tables and a semi-supervised pipeline for automatic annotation. However, we did not consider images in the papers, which also contain a wealth of information. Beyond tables, images often illustrate the work process, pipeline, or framework presented in the scientific paper. In our future work, we plan to introduce and process images from scientific papers into our benchmark to further assist researchers in comprehending papers. This will enable us to provide a more holistic approach to understanding scientific papers and ensure that the benchmark covers all important modalities of information in scientific papers.

Ethics Statement

We collect paper from the free distribution service arXiv[8]. The random crawled papers collected may not have been peer-reviewed. Currently, extraction is based on the assumption of the correctness of the public papers.

References

Simran Arora, Avanika Narayan, Mayee F Chen, Laurel J Orr, Neel Guha, Kush Bhatia, Ines Chami, Frederic Sala, and Christopher Ré. 2022. Ask me anything: A simple strategy for prompting language models. *arXiv preprint arXiv:2210.02441*.

Isabelle Augenstein, Mrinal Das, Sebastian Riedel, Lakshmi Vikraman, and Andrew McCallum. 2017. Semeval 2017 task 10: Scienceie-extracting keyphrases and relations from scientific publications. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 546–555.

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.

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.

[8] https://arxiv.org/

Xiang Deng, Huan Sun, Alyssa Lees, You Wu, and Cong Yu. 2020. Turl: Table understanding through representation learning. *SIGMOD Rec.*, 51:33–40.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *ArXiv*, abs/1810.04805.

Shizhe Diao, Pengcheng Wang, Yong Lin, and Tong Zhang. 2023. Active prompting with chain-of-thought for large language models. *arXiv preprint arXiv:2302.12246*.

Jennifer D'Souza and Sören Auer. 2020. Nlpcontributions: An annotation scheme for machine reading of scholarly contributions in natural language processing literature. In *Proceedings of the 1st Workshop on Extraction and Evaluation of Knowledge Entities from Scientific Documents co-located with the ACM/IEEE Joint Conference on Digital Libraries in 2020 (JCDL 2020)*. Aachen: RWTH.

Sumam Francis, Jordy Van Landeghem, and Marie-Francine Moens. 2019. Transfer learning for named entity recognition in financial and biomedical documents. *Information*, 10(8):248.

Kata Gábor, Davide Buscaldi, Anne-Kathrin Schumann, Behrang QasemiZadeh, Haïfa Zargayouna, and Thierry Charnois. 2018. SemEval-2018 task 7: Semantic relation extraction and classification in scientific papers. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 679–688.

Ridong Han, Tao Peng, Chaohao Yang, Benyou Wang, Lu Liu, and Xiang Wan. 2023. Is information extraction solved by chatgpt? an analysis of performance, evaluation criteria, robustness and errors. *arXiv preprint arXiv:2305.14450*.

Yufang Hou, Charles Jochim, Martin Gleize, Francesca Bonin, and Debasis Ganguly. 2019. Identification of tasks, datasets, evaluation metrics, and numeric scores for scientific leaderboards construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5203–5213.

Yufang Hou, Charles Jochim, Martin Gleize, Francesca Bonin, and Debasis Ganguly. 2021. Tdmsci: A specialized corpus for scientific literature entity tagging of tasks datasets and metrics. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 707–714.

Sarthak Jain, Madeleine van Zuylen, Hannaneh Hajishirzi, and Iz Beltagy. 2020. Scirex: A challenge dataset for document-level information extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7506–7516.

Salomon Kabongo, Jennifer D'Souza, and Sören Auer. 2021. Automated mining of leaderboards for empirical ai research. In *International Conference on Asian Digital Libraries*, pages 453–470. Springer.

Tian Lan, Tianyi Che, Zewen Chi, Xuhao Hu, and Xianling Mao. 2023. Mozi: A scientific large-scale language model.

Bo Li, Gexiang Fang, Yang Yang, Quansen Wang, Wei Ye, Wen Zhao, and Shikun Zhang. 2023. Evaluating chatgpt's information extraction capabilities: An assessment of performance, explainability, calibration, and faithfulness. *arXiv preprint arXiv:2304.11633*.

Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Yi Luan. 2018. Information extraction from scientific literature for method recommendation. *arXiv preprint arXiv:1901.00401*.

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.

Yi Luan, Dave Wadden, Luheng He, Amy Shah, Mari Ostendorf, and Hannaneh Hajishirzi. 2019. A general framework for information extraction using dynamic span graphs. 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 3036–3046.

Wolfgang Otto, Matthäus Zloch, Lu Gan, Saurav Karmakar, and Stefan Dietze. 2023. Gsap-ner: A novel task, corpus, and baseline for scholarly entity extraction focused on machine learning models and datasets. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8166–8176.

Yiheng Shu, Zhiwei Yu, Yuhan Li, Börje F. Karlsson, Tingting Ma, Yuzhong Qu, and Chin-Yew Lin. 2022. TIARA: Multi-grained retrieval for robust question answering over large knowledge base. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Abu Dhabi, United Arab Emirates. 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*.

Vijay Viswanathan, Graham Neubig, and Pengfei Liu. 2021. Citationie: Leveraging the citation graph for scientific information extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 719–731.

Qingyun Wang, Manling Li, Xuan Wang, Nikolaus Parulian, Guangxing Han, Jiawei Ma, Jingxuan Tu, Ying Lin, Ranran Haoran Zhang, Weili Liu, et al. 2021. Covid-19 literature knowledge graph construction and drug repurposing report generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Demonstrations*, pages 66–77.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Jian Wu, Yicheng Xu, Yan Gao, Jian-Guang Lou, Börje F. Karlsson, and Manabu Okumura. 2023. TACR: A Table-alignment-based Cell-selection and Reasoning Model for Hybrid Question-Answering. In *Findings of the Association for Computational Linguistics: ACL 2023*, Toronto, Canada. Association for Computational Linguistics.

Silei Xu, Shicheng Liu, Theo Culhane, Elizaveta Pertseva, Meng-Hsi Wu, Sina J. Semnani, and Monica S. Lam. 2023. Fine-tuned llms know more, hallucinate less with few-shot sequence-to-sequence semantic parsing over wikidata. In *Proceedings of EMNLP2023*.

Sean Yang, Chris Tensmeyer, and Curtis Wigington. 2022. Telin: Table entity linker for extracting leaderboards from machine learning publications. In *Proceedings of the first Workshop on Information Extraction from Scientific Publications*, pages 20–25.

Zhilin Yang, Ruslan Salakhutdinov, and William W Cohen. 2017. Transfer learning for sequence tagging with hierarchical recurrent networks. In *ICLR (Poster)*.

Deming Ye, Yankai Lin, Peng Li, and Maosong Sun. 2022. Packed levitated marker for entity and relation extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4904–4917.

Yu Zhang, Yun Wang, Haidong Zhang, Bin Zhu, Si Chen, and Dongmei Zhang. 2022. Onelabeler: A flexible system for building data labeling tools. *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*.

Zexuan Zhong and Danqi Chen. 2021. A frustratingly easy approach for entity and relation extraction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 50–61.

Yuchen Zhuang, Yinghao Li, Jerry Junyang Cheung, Yue Yu, Yingjun Mou, Xiang Chen, Le Song, and Chao Zhang. 2022. Resel: N-ary relation extraction from scientific text and tables by learning to retrieve and select. *arXiv preprint arXiv:2210.14427*.

A Annotation Guideline

A.1 Entity Category

- **Task:** The specific task or problem that the paper aims to address. E.g., information extraction, machine reading comprehension, image segmentation, etc.
- **Model:** A formal representation or abstraction of the proposed system, which can be applied to solve the specific **Task**. E.g., BERT, ResNet, etc.
- **Method:** The approach, technique, and tool that is used to construct the **Model** to solve the **Task**. E.g., self-attention, data augmentation, Adam, batch normalization, etc.
- **Dataset:** A collection of data that is used for training, validating, and testing the proposed **Model**. E.g., GLUE, COCO, CoNLL-2003, etc.
- **Metric:** A quantitative measure or evaluation criterion that is used to assess the performance or quality of a **Model**. E.g., accuracy, F_1 score, etc.
- **Setting:** It often appears in tables and refers to the context or environment in which the study was conducted. **Setting** is an important aspect of a research paper, as it provides the necessary information to reproduce or replicate the experimental results. E.g., one-hop, multi-hop, dev set, test set, etc.
- **Score:** It refers to a numerical value that is used to evaluate the performance of a **Model** on a specific **Task**, using a particular **Dataset** and **Metric**, under a specific **Setting**.

A.2 Relation Category

Due to the relative scarcity of explicit relationships in body text, we only focus on annotating relations presented in tables. Similar to previous SciIE benchmarks (Hou et al., 2019; Jain et al., 2020; D'Souza and Auer, 2020; Hou et al., 2021; Kabongo et al., 2021), we link different types of entities that are related, without requiring a specific type. This methodology allows for a more flexible representation of the relationships between entities, enabling an easier and deeper understanding of the underlying patterns and connections within the table content.

A.3 Rules and Notes

Considering that annotators may have varying understandings of the annotation details, we have defined a set of rules and notes to standardize the annotation process. The rules defined are the following:

1. Differences between **Model** and **Method**: A **Model** entity refers to the name of a model that can be applied to a specific task independently, while a **Method** entity cannot be used directly to solve a problem or task but can help models improve their performance.
2. The proposed framework, which stacks several models should be classified into **Model**. For example, "YOLOv5" should be annotated as **Model**.
3. We should annotate a combination of a **Model** and a **Method** as a **Method** rather than as a **Model**. For example, the "RNN-based encoder" belongs to a **Method**.
4. Terms such as "networks", "neural", and "model", which do not convey specific meaning and often appear at the beginning or end of entity names, should be excluded. For example, "FasterRCNN network" $\rightarrow$ "FasterRCNN", "neural machine translation" $\rightarrow$ "machine translation", and "question answering model" $\rightarrow$ "question answering"
5. Avoid annotating broad and unclear noun phrases as entities. Scientific terms should be specific and well-defined concepts to ensure clarity and precision. For example, phrases such as "neural network" and "encoder-decoder architecture" should not be considered as **Model** entities.
6. Adjectives that are not directly related to the domain of the research, such as "state-of-the-art", should be avoided. Instead, more specific adjectives that accurately describe the **Model** or **Method** should be kept. For instance, "Bidirectional LSTM".
7. Do not include any determinators (e.g., "the", "a"), or adjective pronouns (e.g., "this", "its", "these", "such") to the entity span.
8. Two entities with the same entity type should not have a relationship.

B Visualization Examples

Method		F1	Hits@1
IR-based methods			
EmbedKGQA*	`[saxena20improving]`	-	66.6
GRAFT-Net	`[sun18open]`	62.8	67.8
PullNet	`[sun19pullnet]`	-	68.1
TransferNet♡	`[shi21transfer]`	-	71.4
Relation Learning♣♡	`[yan21large]`	64.5	72.9
NSM♡*	`[he21improving]`	67.4	74.3
Subgraph Retrieval*	`[zhang22subgraph]`	74.5	83.2

Figure 4: Visualization of annotated entities in tables.

Figure 4 displays a visualization of Table 2 from the TIARA paper, which can be found on arXiv with id "2210.12925". We annotate various types of entities in tables using different colors to make them easily identifiable, which allows reviewers to efficiently and accurately label different kinds of entities and discover relations. Specifically, green represents the `Model`, dark orange represents the `Score`, light orange represents the `Setting`, blue represents `Metric`, and red represents the `Method`.

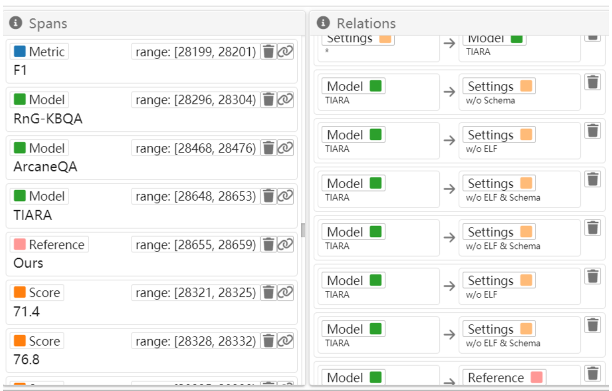

Figure 5: The interface for visualizing NER and RE annotations consists of two rectangles. The left rectangle, labeled "Spans", shows the entity name, entity type, and entity span. The right rectangle, labeled "Relations", shows the related entities within their types.

The interface for annotating entities and relations is depicted in Figure 5. Each entity is labeled with an associated entity type and its corresponding start and end positions within the document. Each relation is labeled with two entities. Entity annotation is done by directly selecting the boundary of the entity, while relation annotation is performed by clicking on the second button in the upper-right corner of the entity box. All entities and relations can be deleted by clicking the delete button.

Text NER	
Error types	**Cases**
Missing entities	ChatGPT struggles with extracting entities with long names. For example, "Question-Answer driven Semantic Role Labeling" fails to be extracted as a **Task** entity, and "dual discourse-level target-guided strategy" fails to be accurately identified as a **Method** entity.
Undefined types	In text NER, even with a predefined set of entity types, it is common for some miscellaneous types to be introduced, such as ["Gábor et al.", **Author**], ["Danqi Chen", **Person**], etc.
Table NER	
Error types	**Cases**
Unannotated entities	"Our model" in the table is not an entity but has been erroneously predicted as a **Model** entity. Similarly, "Train data" in the table is not an entity but has been incorrectly predicted as a **Dataset** entity.
Incorrect types	"K-means" should be labeled as a **Method** entity, but it has been incorrectly predicted as a **Model** entity. Similarly, both "dev set" and "test set" should be labeled as **Setting** entities, but they have been incorrectly predicted as **Dataset** entities.
Table RE	
Error types	**Cases**
Missing relations	If a cell contains a **Model** entity, it should be annotated in relation to other cells in the same row that contain **Score** entities. However, this relationship is often missing in ChatGPT.

Table 6: Cases of different types of extraction errors that can be output by ChatGPT.

C Error Analysis of ChatGPT

In this section, we follow (Han et al., 2023) to categorize and analyze ChatGPT's errors on our three sub-tasks. Through manual checking, we find that the errors mainly include:

- **Missing entities/relations:** Missing one or more annotated target entities or relations.
- **Unannotated entities/relations:** Output the entities or relations that are not annotated in the test set.
- **Incorrect types:** The entity boundaries are correct, while the corresponding type comes from the set of pre-defined types, but does not match the annotated type.
- **Undefined types:** Output the types beyond the pre-defined types when the corresponding entity boundaries are correct.

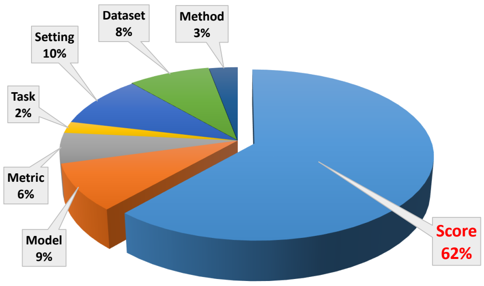

Figure 6: The entity types distribution of tables in the added corpus.

Settings	Table NER		
	P	R	F_1
Table NER w Scores	46.7	83.0	59.8
Table NER w/o Scores	**51.4**	**87.9**	**64.9**

Table 7: The performance of our table NER model with and without the **Score** entities.

- **Others:** Other errors apart from the above errors, such as incorrect output format, output unexpected information, etc.

Table 6 presents several cases of extraction errors made by ChatGPT.

D Discussion of Table NER

As shown in Figure 6, we observe a long-tail problem of entity type distribution of tables, where **Score** entity occupies over 60% of the total entity number. In this case, table NER models are more likely to predict the frequent **Score** entity, which can influence the model's generalization ability. Additionally, some numerical values in tables are simply parameters, data sizes, and other experimental information, but are prone to be predicted as **Score** entities.

In text NER and table RE tasks, label imbalance is unlikely to occur since **Score** entities are relatively rare in text NER, and there is no specific type of relation in table RE. To address this label imbalance issue, we try to re-train a table NER model and test ChatGPT without the **Score** entities.

D.1 Model Re-train

We remove the **Score** entities, and the experiment results shown in Table 7 demonstrate that our table NER without **Score** entities outperforms the previous table NER model, with improvements of 4.7, 4.9, and 5.1 points on precision, recall, and F_1, respectively.

Settings	Table NER		
	P	R	F_1
1-shot ICL w Scores	53.1	16.8	25.5
1-shot ICL w/o Scores	52.3	41.5	46.3
2-shot ICL w Scores	**57.4**	24.7	34.5
2-shot ICL w/o Scores	53.2	**49.2**	**51.1**

Table 8: The performance of ChatGPT in the few-shot ICL setting with and without the **Score** entities.

D.2 ChatGPT without Scores

We conduct experiments on ChatGPT without the **Score** entities, and as shown in Table 8, the experimental results indicate that removing the **Score** entities leads to a significant improvement in recall for ChatGPT on both 1-shot and 2-shot ICL settings. However, it is important to note that removing the **Score** entities also results in a reduction in precision. Overall, our findings suggest that removing the **Score** entities could be a promising approach to improving the performance of table NER.

E Exemplar of the Prompt

When evaluating large language models, prompting is a brittle process wherein small modifications to the prompt can cause large variations in the model predictions, and therefore significant effort should be dedicated to designing a painstakingly crafted perfect prompt for the given task (Arora et al., 2022; Diao et al., 2023). In this study, we follow (Li et al., 2023) and (Han et al., 2023), who have conducted extensive evaluations of the power of ChatGPT in multiple information extraction tasks, to design prompt templates that are well-suited to our tasks. We investigate the performance of few-shot In-context Learning (ICL) on our benchmark. To eliminate the randomness, we manually select two demonstrations for each task, ensuring that all entity types are covered.

We give our designed input examples for three different kinds of scientific information extraction tasks to help readers understand our implementation, as shown in Table 9, 10, and 11, respectively.

Prompts of text NER (2-shot In-context Learning)

Scientific named entity extraction is a task in natural language processing that aims to identify specific entities with semantic meaning from paper text and classify them into predefined types.

Paper text is typically segmented into sequences of tokens and each entity is labeled with a tag indicating its type. Entity types include:

Task: The specific task or problem that the paper aims to address. E.g., information extraction, machine reading comprehension, image segmentation, etc.
Model: A formal representation or abstraction of the proposed system, which can be applied to solve the specific **Task**. E.g., BERT, ResNet, etc.
Method: The approach, technique, and tool that is used to construct the **Model** to solve the **Task**. E.g., self-attention, data augmentation, Adam, batch normalization, etc.
Dataset: A collection of data that is used for training, validating, and testing the proposed **Model**. E.g., GLUE, COCO, CoNLL-2003, etc.
Metric: A quantitative measure or evaluation criterion that is used to assess the performance or quality of a **Model**. E.g., accuracy, F_1 score, etc.

Here are two demonstrations:

Given type set: [Task, Model, Method, Dataset, Metric].
Sentences: Our task is to classify images in the CIFAR-10 dataset into their respective classes, which include animals, vehicles, and household items. This task has practical applications in areas such as autonomous driving, object recognition, and image search. In this paper, we propose a novel approach for image classification using a deep learning model based on the EfficientNet architecture and transfer learning techniques. Our proposed model, named EfficientNet-Transfer, is a modified version of the EfficientNet-B0 architecture that has been pre-trained on the ImageNet dataset and fine-tuned on our target dataset. We use the CIFAR-10 dataset, which contains 60,000 32x32 pixel color images in 10 classes, as our target dataset. We evaluate our model using classification accuracy, precision, recall, and F1-score.
Question: Please extract the named entity from the given sentences. Based on the given label set, provide the extraction results in the format: [[Entity Name, Entity Type]] without any additional things including your explanations or notes. If there is no entity in the given sentence, please return a null list like [].
Entities: [['CIFAR-10', 'Dataset'], ['autonomous driving', 'Task'], ['object recognition', 'Task'], ['image search', 'Task'], ['image classification', 'Task'], ['EfficientNet', 'Method'], ['transfer learning', 'Method'], ['EfficientNet-Transfer', 'Model'], ['EfficientNet-B0', 'Model'], ['ImageNet', 'Dataset'], ['accuracy', 'Metric'], ['precision', 'Metric'], ['recall', 'Metric'], ['F1-score', 'Metric']]

Given type set: [Task, Model, Method, Dataset, Metric].
Sentences: We perform sentiment analysis on movie reviews using deep learning techniques. We propose AttRNN, a novel model architecture based on a recurrent neural network (RNN) with attention mechanisms. Our model integrates both word-level and sentence-level attention mechanisms to improve its discriminative power and capture more relevant features from the input text. We evaluate our proposed model on the Movie Review Sentiment Analysis dataset, which consists of 50,000 movie reviews labeled as positive or negative. We use the accuracy metric to measure the performance of our model, which is defined as the percentage of correctly classified movie reviews in the test set. We also report the F1-score, which takes into account both precision and recall of the positive and negative classes.
Question: Please extract the named entity from the given sentences. Based on the given label set, provide the extraction results in the format: [[Entity Name, Entity Type]] without any additional things including your explanations or notes. If there is no entity in the given sentence, please return a null list like [].
Entities: [['sentiment analysis', 'Task'], ['AttRNN', 'Model'], ['recurrent neural network', 'Method'], ['RNN', 'Method'], ['word-level and sentence-level attention mechanisms', 'Method'], ['attention mechanisms', 'Method'], ['Movie Review', 'Dataset'], ['accuracy', 'Metric'], ['F1-score', 'Metric']]

Given type set: [Task, Model, Method, Dataset, Metric].
Sentences: [S]
Question: Please extract the named entity from the given sentences. Based on the given label set, provide the extraction results in the format: [[Entity Name, Entity Type]] without any additional things including your explanations or notes. If there is no entity in the given sentence, please return a null list like [].
Entities:

Table 9: The prompt template of text NER leveraging 2-shot In-context Learning. [S] denotes the sentences we want to extract.

Prompts of table NER (2-shot In-context Learning) without Score entities
Considering 6 entity types including ‘Task’, ‘Model’, ’Method’, ‘Dataset’, ‘Metric’, ‘Setting’.

Here are two demonstrations:

Given type set: [‘Task’, ‘Model’, ‘Method’, ‘Dataset’, ‘Metric’, ‘Setting’].
Table: [[‘System’, ‘MNLI-(m/mm)’, ‘QQP’, ‘QNLI‘, ‘SST-2’, ‘CoLA’, ‘STS-B’, ‘MRPC’, ‘RTE’, ‘Average’], [‘’, ‘392k’, ‘363k’, ‘108k’, ‘67k’, ‘8.5k’, ‘5.7k’, ‘3.5k’, ‘2.5k’, ‘-’], [‘Pre-OpenAI SOTA’, ‘80.6/80.1’, ‘66.1’, ‘82.3’, ‘93.2’, ‘35.0’, ‘81.0’, ‘86.0’, ‘61.7’, ‘74.0’], [‘BiLSTM+ELMo+Attn’, ‘76.4/76.1’, ‘64.8’, ‘79.8’, ‘90.4’, ‘36.0’, ‘73.3’, ‘84.9’, ‘56.8’, ‘71.0’], [‘OpenAI GPT’, ‘82.1/81.4’, ‘70.3’, ‘87.4’, ‘91.3’, ‘45.4’, ‘80.0’, ‘82.3’, ‘56.0’, ‘75.1’], [‘bertbase’, ‘84.6/83.4’, ‘71.2’, ‘90.5’, ‘93.5’, ‘52.1’, ‘85.8’, ‘88.9’, ‘66.4’, ‘79.6’], [‘bertlarge’, ‘86.7/85.9’, ‘72.1’, ‘92.7’, ‘94.9’, ‘60.5’, ‘86.5’, ‘89.3’, ‘70.1’, ‘82.1’]]
Question: Please extract the named entity from the given table and output a JSON object that contains the following: {‘Task’: [list of entities], ‘Dataset’: [list of entities], ‘Model’: [list of entities], ‘Method’: [list of entities], ‘Metric’: [list of entities], ‘Setting’: [list of entities]}. If no entities are presented in any categories keep it None.
Entities: {‘Task’: [‘QQP’, ‘MRPC’], ‘Dataset’: [‘MNLI-(m/mm)’, ‘QQP’, ‘QNLI’, ‘SST-2’, ‘CoLA’, ‘STS-B’, ‘MRPC ’, ‘RTE’, ‘Average’, ‘GLUE Test’, ‘WNLI set’, ‘STS-B’], ‘Model’: [‘Pre-OpenAI’, ‘BiLSTM+ELMo+Attn’, ‘OpenAI GPT’, ‘bertbase’, ‘bertlarge’, ‘BERT’, ‘OpenAI GPT’, ‘BERT’], ‘Method’: [], ‘Metric’: [‘F1 scores’, ‘Spearman correlations’, ‘accuracy scores’], ‘Setting’: []}

Given type set: [‘Task’, ‘Model’, ‘Method’, ‘Dataset’, ‘Metric’, ‘Setting’].
Table: [[‘System’, ‘Dev’, ‘Dev’, ‘Test’, "Test’], [‘’, ‘EM’, ‘F1’, ‘EM’, ‘F1’], [‘Top Leaderboard Systems (Dec 10th, 2018)’, ‘Top Leaderboard Systems (Dec 10th, 2018)’, ‘Top Leaderboard Systems (Dec 10th, 2018)’, ‘Top Leaderboard Systems (Dec 10th, 2018)’, ‘Top Leaderboard Systems (Dec 10th, 2018)’], [‘Human’, ‘-’, ‘-’, ‘82.3’, ‘91.2’], [‘#1 Ensemble - nlnet’, ‘-’, ‘-’, ‘86.0’, ‘91.7’], [‘#2 Ensemble - QANet’, ‘-’, ‘-’, ‘84.5’, "90.5’], [‘Published’, ‘Published’, ‘Published’, ‘Published’, ‘Published’], [‘BiDAF+ELMo (Single)’, ‘-’, "85.6’, ‘-’, ‘85.8’], [‘R.M. Reader (Ensemble)’, ‘81.2’, ‘87.9’, ‘82.3’, ‘88.5’], [‘Ours’, ‘Ours’, ‘Ours’, ‘Ours’, ‘Ours’], [‘bertbase(Single)’, ‘80.8’, ‘88.5’, ‘-’, ‘-’], [‘bertlarge(Single)’, ‘84.1’, ‘90.9’, ‘-’, ‘-’], [‘bertlarge(Ensemble)’, ‘85.8’, ‘91.8’, ‘-’, ‘-’], [‘bertlarge(Sgl.+TriviaQA)’, ‘84.2’, ‘91.1’, ‘85.1’, ‘91.8’], [‘bertlarge(Ens.+TriviaQA)’, ‘86.2’, ‘92.2’, ‘87.4’, ‘93.2’]]
Question: Please extract the named entity from the given table and output a JSON object that contains the following: {‘Task’: [list of entities], ‘Dataset’: [list of entities], ‘Model’: [list of entities], ‘Method’: [list of entities], ‘Metric’: [list of entities], ‘Setting’: [list of entities]}. If no entities are presented in any categories keep it None.
Entities: {‘Task’: [], ‘Dataset’: [‘Dev’, ‘Test’, ‘SQuAD 1.1’], ‘Model’: [‘Human’, ‘#1 Ensemble-nlnet’, ‘#2 Ensemble - QANet’, ‘BiDAF+ELMo (Single)’, ‘R.M. Reader (Ensemble)’, ‘bertbase’, ‘bertlarge’, ‘bertlarge’, ‘bertlarge’, ‘bertlarge’, ‘BERT ensemble’], ‘Method’: [], ‘Metric’: [‘EM’, ‘F1’, ‘EM’, ‘F1’], ‘Setting’: [‘Top Leaderboard Systems (Dec 10th, 2018)’, ‘Published’, ‘Ours’, ‘Single’, ‘Single’, ‘Ensemble’, ‘Sgl.+TriviaQA’, ‘Ens.+TriviaQA’]}

Given type set: [‘Task’, ‘Model’, ‘Method, ‘Dataset’, ‘Metric’, ‘Setting’].
Table: [T]
Question: Please extract the named entity from the given table and output a JSON object that contains the following: {‘Task’: [list of entities], ‘Dataset’: [list of entities], ‘Model’: [list of entities], ‘Method’: [list of entities], ‘Metric’: [list of entities], ‘Setting’: [list of entities]}. If no entities are presented in any categories keep it None.
Entities:

Table 10: The prompt template of table NER leveraging 2-shot In-context Learning. [T] denotes the table we want to extract.

Prompts of table RE (2-shot In-context Learning)

Considering relations between cells in tables:

Here are two demonstrations:

Table: [['System', 'MNLI-(m/mm)', 'QQP', 'QNLI', 'SST-2', 'CoLA', 'STS-B', 'MRPC', 'RTE', 'Average'], ['', '392k', '363k', '108k', '67k', '8.5k', '5.7k', '3.5k', '2.5k', '-'], ['Pre-OpenAI SOTA', '80.6/80.1', '66.1', '82.3', '93.2', '35.0', '81.0', '86.0', '61.7', '74.0'], ['BiLSTM+ELMo+Attn', '76.4/76.1', '64.8', '79.8', '90.4', '36.0', '73.3', '84.9', '56.8', '71.0'], ['OpenAI GPT', '82.1/81.4', '70.3', '87.4', '91.3', '45.4', '80.0', '82.3', '56.0', '75.1'], ['bertbase', '84.6/83.4', '71.2', '90.5', '93.5', '52.1', '85.8', '88.9', '66.4', '79.6'], ['bertlarge', '86.7/85.9', '72.1', '92.7', '94.9', '60.5', '86.5', '89.3', '70.1', '82.1']]
Question: Please extract all relations from the given table and output a JSON object that contains the following: {[cell: cell]}. If no relations are presented keep it None.
Relations: {['Pre-OpenAI SOTA':'MNLI-(m/mm)', 'Pre-OpenAI SOTA':'QQP','Pre-OpenAI SOTA':'QNLI','Pre-OpenAI SOTA':'SST-2', 'Pre-OpenAI SOTA':'CoLA', 'Pre-OpenAI SOTA':'STS-B', 'Pre-OpenAI SOTA':'MRPC', 'Pre-OpenAI SOTA':'RTE', 'Pre-OpenAI SOTA':'Average', 'Pre-OpenAI SOTA':'80.6/80.1','Pre-OpenAI SOTA':'66.1','Pre-OpenAI SOTA':'82.3','Pre-OpenAI SOTA':'93.2','Pre-OpenAI SOTA': '35.0','Pre-OpenAI SOTA':'81.0','Pre-OpenAI SOTA':'86.0','Pre-OpenAI SOTA':'61.7', 'Pre-OpenAI SOTA':'74.0']}

Table: [['System', 'Dev', 'Dev', 'Test', 'Test'], ['', 'EM', 'F1', 'EM', 'F1'], ['Top Leaderboard Systems (Dec 10th, 2018)', 'Top Leaderboard Systems (Dec 10th, 2018)', 'Top Leaderboard Systems (Dec 10th, 2018)', 'Top Leaderboard Systems (Dec 10th, 2018)', 'Top Leaderboard Systems (Dec 10th, 2018)'], ['Human', '-', '-', '82.3', '91.2'], ['#1 Ensemble - nlnet', '-', '-', '86.0', '91.7'], ['#2 Ensemble - QANet', '-', '-', '84.5', "90.5'], ['Published', 'Published', 'Published', 'Published', 'Published'], ['BiDAF+ELMo (Single)', '-', "85.6', '-', '85.8'], ['R.M. Reader (Ensemble)', '81.2', '87.9', '82.3', '88.5'], ['Ours', 'Ours', 'Ours', 'Ours', 'Ours'], ['bertbase(Single)', '80.8', '88.5', '-', '-'], ['bertlarge(Single)', '84.1', '90.9', '-', '-'], ['bertlarge(Ensemble)', '85.8', '91.8', '-', '-'], ['bertlarge(Sgl.+TriviaQA)', '84.2', '91.1', '85.1', '91.8'], ['bertlarge(Ens.+TriviaQA)', '86.2', '92.2', '87.4', '93.2']]
Question: Please extract all relations from the given table and output a JSON object that contains the following: {[cell: cell]}. If no relations are presented keep it None.
Relations: {['#1 Ensemble - nlnet':'Dev', '#1 Ensemble - nlnet':'EM', '#1 Ensemble - nlnet': 'Top Leaderboard Systems (Dec 10th, 2018)', '#1 Ensemble - nlnet': '86.0', '#1 Ensemble - nlnet': 'Test','#1 Ensemble - nlnet':'F1', '#1 Ensemble - nlnet': '91.7']}

Table: [T]
Question: Please extract all relations from the given table and n output a JSON object that contains the following: {[cell: cell]}. If no relations are presented keep it None.
Entities:

Table 11: The prompt template of table RE leveraging 2-shot In-context Learning. [T] denotes the table we want to extract. Since the relations are very dense in tables, we here list part of the relations.