Towards Fine-Grained Extraction of Scientific Claims from Heterogeneous Tables Using Large Language Models

Daniele Bertillo
Roma Tre University
Italy
daniele.bertillo@uniroma3.it

Laks V.S. Lakshmanan
Univ. of British Columbia
Canada
laks@cs.ubc.ca

Paolo Merialdo
Roma Tre University
Italy
paolo.merialdo@uniroma3.it

Divesh Srivastava
AT&T Chief Data Office
USA
divesh@research.att.com

ABSTRACT

The rapid growth of scientific literature presents both opportunities and challenges for extracting actionable insights to support evidence-based decision-making, in-depth analyses, and resolution of discrepancies among contradicting scientific results. While tables in research papers are critical sources of scientific claims, their structural diversity and dispersed contextual details hinder automated analysis. We introduce a novel approach to model and extract fine-grained claims, structured as ⟨subject, measure(s), outcome(s)⟩ triples, from such tables using large language models (LLMs). As part of the DESIREE project, which focuses on developing scalable methods for analyzing scientific literature to support robust, future research, our contributions include: (i) a claim model capturing detailed experimental contexts; (ii) a benchmark of 1,698 fine-grained claims from 80 papers in medicine and computer science; and (iii) experimental evaluation with three LLM-based approaches, using four LLMs. Our manually curated benchmark serves as a valuable resource for future research, and our results highlight the potential of large language models (LLMs) to support in-depth analysis of scientific findings.

VLDB Workshop Reference Format:
Daniele Bertillo, Laks V.S. Lakshmanan, Paolo Merialdo, and Divesh Srivastava. Towards Fine-Grained Extraction of Scientific Claims from Heterogeneous Tables Using Large Language Models. VLDB 2025 Workshop: Tabular Data Analysis (TaDA).

VLDB Workshop Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/dbertillo/desiree_tada25.

1 INTRODUCTION

The global scientific production, measured in number of publications, is growing at a very fast pace. The insights and research findings reported in scientific papers represent a unique opportunity as they can serve as a treasure trove of knowledge with the potential to support evidence-based decision-making processes, inspire new research directions, and drive innovation in various domains. To harness its full potential, it is important to conduct in-depth analyses of scientific literature. However, the sheer diversity of scientific results raises major challenges and warrants considerable effort for analyzing published literature.

Many attempts have been made to develop tools and techniques for automatically exploring, analyzing, and extracting insights from the scientific literature. For example, significant efforts focused on the automation of systematic reviews of the literature [29, 45], whose goal is to provide comprehensive summaries of research articles, addressing one or more research questions, in order to synthesize a large number of publications. More recently, some commercial solutions based on large language models (LLMs) are being proposed, e.g., ScholarAI [2], Scite [3], Consensus [1], as well as the "DeepResearch" features by ChatGPT and Gemini. These tools are designed to facilitate automated literature reviews across large corpora of articles. They provide effective summaries and demonstrate good reasoning abilities, but they typically tend to summarize and reason about findings reported in publications at a coarse level of granularity.

We observe that a fine-grained extraction of scientific results is crucial to enable precise and in-depth analyses, for example, to reveal the causes of apparent contradictions in the findings of different articles, or discover new research directions.

For instance, in medicine, a systematic review often includes a meta-analysis: a statistical technique that synthesizes quantitative data from multiple independent studies investigating similar hypotheses, aiming to produce a more robust and reliable conclusion. This process requires detailed data from numerous studies to enhance the precision and power of the overall estimate. To give an example, consider recent meta-analyses that study the correlation between coffee consumption and the incidence of various types of cancer: Kennedy *et al.* [28], which focuses on hepatocellular carcinoma, analyzed 26 papers; Yu *et al.* [25], on lung cancer, 26 papers; Wang *et al.* [48], on breast cancer, 45 papers. The authors of these studies had to extract detailed and structured information from each article, including information about the study design, population characteristics (e.g., age, sex, and health status), consumption details (type, dosage, duration), and the comparator or control condition. These data extraction activities are extremely time-consuming and require significant efforts from researchers [4].

Proceedings of the VLDB Endowment. ISSN 2150-8097.

It is worth observing that these studies, which are based on fine-grained analysis of large corpora of papers at scale, are crucial to accelerate the discovery of new insights and drive scientific research towards previously ignored or underexplored directions. For example, the study by Kennedy *et al.* suggested that each additional two cups of coffee per day was linked to a 35% lower risk of hepatocellular carcinoma. Building on such findings, other researchers pursued new research directions in liver cancer. Deng *et al.* [13] used genetic markers to confirm a causal relationship between coffee intake and reduced liver cancer risk in an East Asian population, ruling out confounding factors. Additionally, Fan *et al.* [15] proposed that caffeine in coffee may inhibit a pro-inflammatory complex involved in liver carcinogenesis, opening new avenues for research, such as identifying caffeine-affected biomarkers.

In a computer science scenario, consider a researcher examining recent advances in Entity Resolution, a well-known data management problem. When analyzing state-of-the-art methods, the researcher encounters discrepancies among results reported in different papers. For example, a paper by Peeters *et al.* [41] reports an F1-score of 82.11 for Ditto, a state-of-the-art solution, on the widely used ABT-Buy dataset, while the original Ditto paper [35] claims a higher F1-score of 89.33 on the same dataset. This discrepancy (82.11 vs. 89.33) stems from variations in Ditto's pretrained embeddings used in the two studies,[1] underscoring the need for careful fine grained data analysis of the experimental settings from the literature. Interestingly, a subsequent study focused precisely on the analysis of pretrained embeddings for Entity Resolution [50].

Our paper introduces a novel approach to model and extract scientific findings at a fine granularity, specifically by identifying their detailed contextual information as reported in scientific publications.

This work is part of a broader project, DESIREE (Data-driven Empirical Science: Improving Robustness, Explainability, and Exploration), that aims to develop scalable solutions for analyzing scientific literature, to enable robust, future research.

In DESIREE, we model scientific findings with fine-grained, structured data called *claims*. Intuitively, a claim represents both the measure and the outcome of a scientific finding, along with the associated details of the relevant experimental setting. Example claims are presented in Section 3.

Automatically identifying and representing scientific findings and their associated context in the form of claims is a challenging issue, which requires converting the information conveyed by papers into a form amenable for complex tasks, such as meta-analysis or comparison of literature results. As tables are the predominant format for presenting findings in a research paper [17, 24, 27], in our solution they serve as the primary source for extracting information to construct the claims found in research papers in a structured format. However, although tables can serve as crucial sources of information for building claims, many details necessary to complete the specification of a claim are often dispersed throughout the paper, appearing in the table caption and the main text.

Since Large Language Models (LLMs) have demonstrated remarkable abilities in natural language processing tasks, they are promising for the challenging task of extracting claims from scientific articles.

In this paper, we present the results of our experimental analysis on leveraging LLMs to extract fine-grained claims from scientific articles, and make the following contributions: (*i*) We introduce a the notion of *claims* to model scientific findings at a fine-granularity level. (*ii*) We introduce a *benchmark* comprising over 1,698 fine-grained claims extracted from 80 papers in the domains of medicine and computer science. (*iii*) We report the results of three LLM-based approaches for the efficient and effective extraction of claims, using a variety of LLMs.

2 RELATED WORK

Automated extraction of structured research findings from the rapidly expanding volume of scientific literature has garnered significant attention, driven by its potential to accelerate scientific discovery and facilitate knowledge synthesis.

Efforts to automate systematic reviews have aimed to synthesize large corpora of scientific articles [29, 45]. Commercial tools like ScholarAI [2], Scite [3], and Consensus [1] leverage LLMs to provide summaries and answer research questions. However, these tools focus on coarse-grained insights, lacking the detailed contextual information (e.g., study design, population characteristics) needed for precise meta-analyses or discrepancy resolution, which our approach targets.

Several efforts have focused on building knowledge graphs from scientific abstracts to represent structured knowledge [5, 9, 14, 16, 21, 23, 40, 46, 49]. These approaches rely on ontologies to formalize predefined concepts and relationships, limiting their ability to specific domains and high-level entities. By focusing solely on abstracts, they overlook detailed experimental findings that are reported in tables, which our method targets for comprehensive, fine-grained extraction.

Inspired by SciBERT [6], a pretrained language model based on BERT, trained on a science and technology corpus, many pretrained language models have been specialized in diverse scientific fields, ranging from computer science [22] to biomedicine [19, 30, 31]. While effective for entities explicitly mentioned in text, these models struggle with tabular data and require extensive fine-tuning on domain-specific labeled datasets, which is resource-intensive. Their lack of generalizability across scientific domains contrasts with our approach, which leverages LLMs for cross-domain adaptability without extensive fine-tuning.

Recent studies have explored LLMs for extracting structured experimental data. For example, Dagdelen *et al.*[11] employ LLMs to extract structured data from scientific publications in the field of solid-state materials, but the extraction is performed only from abstracts, thus missing important details of the experimental outcomes. ChatExtract [42] focuses on material science and aims at extracting Material–Value–Unit triplets using engineered prompts and follow-up questions to ensure accuracy. Similarly, Circi *et al.* [10] present a domain-specific solution for polymer composites, aiming to extract structured material properties from both text and tables. In the machine learning domain, Kardas *et al.* [26] developed a pipeline that identifies tables containing scientific results and classifies each cell into one of a set of predefined categories: dataset,

[1] Namely, [41] adopted BERT instead of RoBERTa, which was used in Ditto.

metric, paper model, cited model, and task. In the broader domain of computer science, Hou *et al.*[20] addressed the related problem of extracting from an experimental scientific paper: tasks, datasets, evaluation metrics, and the corresponding best numeric scores. While these approaches mark a step towards fine-grained data extraction from tables, they remain restricted to a predefined set of properties and measurement contexts within a narrow scientific area, limiting their broader applicability.

The challenge of extracting information from scientific tables has also been explored through table-based question answering [33, 43, 51] and scientific fact-checking [8, 39]. However, these systems typically retrieve answers to specific questions, rather than capturing all relevant claims expressed in tabular form.

Lu *et al.* [38] provide a comprehensive survey on the use of large language models for table processing across a broad range of tasks, including table question answering, fact verification, table-to-text generation, table detection, table extraction, column type annotation, and entity linking. Complementing this, another recent study [44] evaluates the capabilities of LLMs in understanding the structural properties of tables, focusing on tasks such as table partitioning, table size detection, merged cell detection, cell lookup and reverse lookup, and column and row retrieval. While these works offer valuable insights into the structural and functional aspects of table processing, they primarily address generic table understanding and task-specific interactions. In contrast, our work aims at extracting all scientific claims embedded in experimental tables—a task that requires not only structural parsing but also fine-grained semantic interpretation of the tabular content in the context of scientific experimentation.

3 MODELING CLAIMS

In this section, we present our model for abstracting fine-grained claims from scientific papers, capturing detailed experimental findings and contextual information. A scientific publication reports scientific "findings" or "results" discovered by scientists from a study. To enable fine-grained extraction of these findings, we represent scientific claims as triples, as follows:

$$\langle subject, measure(s), outcome(s) \rangle.$$

The *subject* comprises a set of $|name, value|$ pairs, each pair termed *specification*, which captures detailed contextual information. The *measure(s)* represent a vector of metrics or measurement attributes used in the experiments, while the *outcome(s)* denote the corresponding measured values. The distinction between a vector of a single measure and a vector of multiple measures depends on whether each measure can be meaningfully understood in isolation. If a measure cannot be interpreted independently of other related measures, then they must be reported together in the same vector.

To give an example, consider the table shown in Figure 1, which reports the results of an experimental evaluation of different text-to-SQL models. Such a table includes 24 claims, where each claim reports the performance expressed by either the "Syntactical Accuracy" or the "Execution Accuracy" metric of a model (e.g., "Pointer-SQL+EG(3)"), on a split (either "Dev" or "Test") of the "WikiSQL" dataset. Three illustrative claims extracted from this table are:

- ⟨ { |`Model`, Pointer-SQL (2017)|, |`Dataset`, WikiSQL|, |`Split`, Dev| }, [`Syntactical Accuracy`], [61.8] ⟩

Model	Dev		Test	
	Acc_{syn}	Acc_{ex}	Acc_{syn}	Acc_{ex}
Pointer-SQL (2017)	61.8	72.5	62.3	71.9
Pointer-SQL + EG (3)	66.6	77.3	66.7	76.9
Pointer-SQL + EG (5)	**67.5**	**78.4**	**67.9**	**78.3**
Coarse2Fine (2018)	72.9	79.2	71.7	78.4
Coarse2Fine + EG (3)	75.6	83.4	74.8	83.0
Coarse2Fine + EG (5)	**76.0**	**84.0**	**75.4**	**83.8**

Table 1: Test and Dev accuracy (%) of the models on WikiSQL data, where Acc_{syn} refers to syntactical accuracy and Acc_{ex} refers to execution accuracy. "+ EG (k)" indicates that model outputs are generated using the execution-guided strategy with beam size k.

Figure 1: An example in computer science, from [47].

- ⟨ {|`Model`, Pointer-SQL + EG (3)|, |`Strategy`, Execution-guided|, |`Beam size`, 3|, |`Dataset`, WikiSQL|, |`Split`, Dev|}, [`Syntactical Accuracy`], [66.6] ⟩
- ⟨ {|`Model`, Pointer-SQL + EG (3)|, |`Strategy`, Execution-guided|, |`Beam size`, 3|, |`Dataset`, WikiSQL|, |`Split`, Test|}, [`Syntactical Accuracy`], [66.7] ⟩

This example highlights several challenges inherent in correctly extracting claims from tables. First, while some specifications (e.g., "`Model`") and their corresponding values are explicitly stated within the table, others (e.g., "`Dataset`") are located only in the table caption. For some specifications the name (and the semantics) need to be extracted from the text of the paper: in our example, this is the case for "`Split`" of the dataset (whose value is either "Dev" or "Test"), which is mentioned in the paper, in the paragraph describing the experimental setting. A further challenge stems from nested table structures: for instance, the table of our example is horizontally nested by "`Split`". Moreover, some cells contain multiple specifications, often encoded through ad hoc formatting. In our example, the "`Execution strategy`" and "`Beam size`" are embedded within the name of the model, with their semantics clarified in the table caption. Finally, metric names—such as 'Syntactical Accuracy' and 'Execution Accuracy'—may exhibit slight variations between the table and its caption, yet consistently convey the same essential information.

The notion of *claims* offers a structured, fine-grained representation of findings and information presented in scientific studies, allowing for data-oriented approaches to analyze the outcomes on a large scale across a vast collection of papers.

Table 3: Results of multivariate Cox proportional hazards regression analysis performed to assess the impact of multiple factors on overall survival (OS) from first phase I treatment

Variable	Contrast	Hazard ratio (95% CI)	*P* value
ECOG performance status	> 0 vs. 0	1.47 (0.90, 2.41)	0.12
Liver metastases	Yes vs. No	1.72 (1.01, 2.91)	0.045
No. of metastatic sites	>2 vs. <= 2	1.33 (0.9, 2.25)	0.28
Prior radiation therapy	Yes vs. No	1.58 (0.95, 2.61)	0.077
Prior FOLFIRINOX	Yes vs. No	1.73 (1.01, 2.98)	0.046
Prior gemcitabine plus nab-paclitaxel treatment	Yes vs. No	1.08 (0.58, 2.01)	0.80

Figure 2: An example in the medical domain, from [18].

Figure 2 shows another example, from the medical domain, that is useful to illustrate a vector of measurements. The table reports the results of a statistical analysis ("multivariate Cox proportional hazards regression") evaluating factors affecting overall survival (OS) in cancer patients starting phase I treatment. For each variable (e.g., ECOG performance status, liver metastases), it lists the contrast (e.g., >0 vs. 0), hazard ratio (HR) with 95% confidence intervals (CI), and p-value. The HR quantifies the relative risk of the event (death) occurring in one group compared to another, adjusted for other variables (for example, Liver metastases (Yes vs. No): HR = 1.72 (p = 0.045) indicates a 72% increased risk of mortality for patients with liver metastases). Confidence Interval (CI) and p-value provide additional insights about the hazard ratio (HR): the 95% CI provides a range within which the true hazard ratio is likely to lie with 95% confidence (a narrow CI, suggests a more precise estimate); the p-value measures the statistical significance of the HR, testing the null hypothesis that the variable has no effect on survival (i.e., HR = 1.0). We observe that 95% CI and p-value refer to the HR, and hence they have to be considered together as a composite measure.

Accordingly, two illustrative claims extracted from this table are as follows:

- ⟨ { `|Variable`, ECOG performance status`|`, `|Contrast`, >0 vs. 0`|`, `|Statistical method`, Multivariate Cox`|`, `|Impact on`, Overall survival`|`, `|Starting phase`, I treatment`|`}, `[HR, 95% CI, P-value]`, [1.47, 0.90–2.41, 0.12] ⟩
- ⟨ { `|Variable`, Liver metastases`|`, `|Contrast`, Yes vs. No`|`, `|Statistical method`, Multivariate Cox`|`, `|Impact on`, Overall survival`|`, `|Starting phase`, I treatment`|`}, `[HR, 95% CI, P-value]`, [1.72, 1.01–2.91, 0.045] ⟩

4 THE DESIREE BENCHMARK

We introduce a human-annotated benchmark consisting of 80 tables reporting scientific findings across two domains–computer science and medicine–each covering two specific topics: Text-to-SQL and Entity Resolution for computer science, and pancreatic cancer and HIV for medicine.

Numerous tools and techniques exist for extracting text and tables from research articles [12], including deep learning-based approaches for layout analysis [7], OCR [53], and table parsing [34]. Our contribution in this work is orthogonal to this body of work. Specifically, we focus on the downstream task of utilizing this data rather than on refining the parsing process. Given this, we chose to work with open access publications directly available in a textual format, namely HTML on arxiv [2] and in XML on PubMed, [3] similarly to [26].

We randomly selected 80 publications, using targeted search keywords related to *text-to-SQL*, *entity resolution*, *pancreatic cancer* and *HIV* to cover the *computer science* and *medicine* domains. For each publication, we randomly selected one table reporting experimental results. We limited our selection to a single table per publication to maximize diversity, as authors often use a consistent style across multiple tables within the same article. For each table, we extracted the following elements:

- the HTML (or XML, in case of PubMed) source of the table;
- the associated caption;
- any footnotes referenced in the table;
- the first paragraph referencing the table in the text of the article.

Table 1 shows structural features of the tables. A header or index (i.e,. the first column if it is a dimension of the table) is considered nested if at least one its columns (or rows) is nested. Hence, a table is flagged as nested if it presents at least one nested header or nested index, otherwise it is flagged relational. A table is reported as cross-table if presents features on both header and index, and for which the metric or measure used in the experiment is mentioned in the caption or the paragraph. Finally, we report the average number of rows and columns.

Two observations are worthy of brief discussion regarding the dataset. Cross-tables and relational tables seems to be more common in the computer science's topics we have taken into consideration, while in medicine it appears to present a generally larger number of nested tables (specifically nested indexes) compared to computer science. Also, the average number of rows is higher in medicine than computer science.

For each table, we manually constructed a ground truth set of the claims it reports. As we are interested in a comprehensive fine-grained description of the claims, we have taken into account also the experimental details, specifications or measures, that may have been described in the caption of the table or in the referencing paragraph and not necessarily captured in the table explicitly.

Table 2 shows the statistics of the content of the benchmark dataset, both for source data and ground truths. For each topic, we show the total count of characters for each of the elements in the data source (table, caption, footnotes and paragraph), along with the number of claims and specifications in the ground truths.

4.1 Metrics

We define two sets of evaluation metrics, each capturing different aspects of extraction accuracy:

- **Claims Precision, Recall, F1-score:**These metrics assess whether the extraction process successfully identifies and captures the correct set of claims. Precision is computed as the proportion of matched claims among all extracted claims, while recall measures the proportion of ground truth claims that were successfully matched to an extracted claim. F1-score is computed as the harmonic mean of precision and recall.
- **Specifications Precision, Recall, F1-score:** These metrics focus on the set of specifications within the matched claims; namely, the subject elements describing the experimental context (e.g., datasets, condition, treatment, population). The goal is to evaluate whether the model accurately captured the contextual elements of a result, independent of the associated metrics or outcomes.

This two-tiered evaluation—first at the level of claims, then at the level of specifications—allows us to assess not only whether the model identifies the right results but also whether it understands and captures the underlying experimental setup.

[2] https://arxiv.org/ https://ar5iv.labs.arxiv.org/
[3] https://pmc.ncbi.nlm.nih.gov/tools/openftlist/

Table 1: Structural characteristics of tables by domain and topic. For each topic, we report the number and percentage of tables exhibiting specific structural features. Each table is categorized as either *relational* or *nested*, and these two categories are mutually exclusive and exhaustive (i.e., their counts sum to 100%). Other features—*cross-table structure*, *nested index*, and *nested header*—are not mutually exclusive and can co-occur within the same table. Average row and column counts are also reported for each topic.

Domain	Topic	#relational tables	#nested tables	#cross tables	#nested indexes	#nested headers	avg #rows	avg #cols
Computer Science	Text-to-SQL	11 (55%)	9 (45%)	13 (65%)	4 (20%)	7 (35%)	6.2	5.15
	Entity resolution	13 (65%)	7 (35%)	10 (50%)	3 (15%)	5 (25%)	5.0	5.55
Medicine	Pancreatic cancer	3 (15%)	17 (85%)	0	9 (45%)	11 (55%)	9.9	5.85
	HIV	11 (55%)	9 (45%)	4 (20%)	7 (35%)	3 (15%)	8.0	4.85

Table 2: Dataset statistics across domains and topics.

Domain	Topic	#chars				#claims	#specifications
		tables	captions	footnotes	paragraphs		
Computer Science	Text-to-SQL	30,852	3,227	0	14,333	471	2,086
	Entity resolution	23,957	2,585	0	19,810	406	1,544
Medicine	Pancreatic cancer	60,802	1,817	2,792	9,734	539	2,687
	HIV	24,828	1,899	5,274	6,512	282	1,392

5 EVALUATING CLAIM EXTRACTION BY LLMS

In this section, we present the experimental evaluation of three large language model (LLM) strategies designed to extract fine-grained claims, structured as ⟨subject, measure(s), outcome(s)⟩ triples, from the tables and their associated elements of our benchmark. The strategies we consider are **0-shot direct extraction**, **1-shot direct extraction**, and **bootstrap**, a few-shot extraction performed by a small LLM bootstrapped by examples generated by a larger LLM. These strategies are assessed using four distinct large language models (LLMs) for their effectiveness in extracting claims and their ability to capture detailed specifications. Our evaluation leverages an LLM-as-a-judge framework, where a large LLM evaluates the semantic and factual accuracy of extracted triples against the manually curated ground-truth. We report precision, recall, F1-score on both claim extraction, focusing on the completeness and correctness of subjects, measures, and outcomes, and specification extraction, emphasizing the richness of contextual details.

5.1 LLM Claim Extraction Pipelines

We explored three pipelines to assess the ability of LLMs to perform the task effectively. These pipelines serve as exploratory approaches aimed at identifying the limitations of LLMs, guiding the design of more effective and efficient solutions.

0-shot direct extraction: The LLM is prompted to extract all claims from the whole content available. The prompt describes the notion of claim and the JSON format we need in output, then reports the target table (in HTML or XML format), its caption and footnotes, and the first paragraph that mentions the table in the paper.

1-shot direct extraction: The prompt used in this pipeline extends that of the previous one by including an example consisting of a table and its associated set of claims. While the example is aligned with the scientific domain of the paper containing the table (medicine or computer science), it remains generic—that is, it does not reflect the specific topic of the paper. Moreover, the table has a relational structure, with no nested elements or cells containing multiple data points, as in preliminary experiments, we observed that examples with more complex table structures yielded poorer results, as they did not align well with the actual table and consequently confused the LLM.

Bootstrap Extraction: While more powerful LLMs generally yield better results, they come with higher computational and monetary costs. Aiming to mitigate these costs and help smaller LLMs perform more effectively, for each table, this pipeline first uses a large LLM with the same prompt as the previous pipeline, but applied to only a small portion of the table (e.g., a few rows). The claims extracted from this subset are then used as *in-context examples* to guide a smaller LLM in processing the whole corpus of the table. The rationale behind this pipeline is based on the observation that claim patterns are often consistent across the rows of a table. Therefore, providing a well-structured in-context example, generated by a large (and possibly more precise) LLM from the first few rows, can effectively guide the smaller model and potentially improve its performance when extracting claims from the entire table.

This approach significantly reduces the number of tokens processed by the large LLM (in particular those produced in output)[4]

[4]It is worth noting that, in commercial LLMs, output tokens are typically more expensive than input tokens—for instance, with OpenAI models, the cost of output tokens is 4× higher.

while providing the smaller, more cost-effective model with tailored examples specific to the input table.

All outputs produced by the pipelines are post-processed to retain only the lines that match a regular expression validating the expected claim format.

5.2 Large Language Models

In addition to evaluating the performance of different extraction strategies, we also employed a diverse set of large language models varying in size, including both proprietary and open-weight models. In particular, we selected the following models for our experiments:

- **GPT-4o**: the OpenAI model, accessed via the Microsoft Azure OpenAI Service.
- **Claude 3-5 Sonnet**: the Anthropic model, accessed through Amazon Bedrock.
- **LLaMA3-70B**: A state-of-the-art open-weight model developed by Meta. Accessed through Amazon Bedrock.
- **LLaMA3-8B**. A lightweight counterpart to the 70B model, also accessed via Amazon Bedrock.

In the bootstrap pipeline, for each topic, we used the best performing LLM to generate examples in the first phase, and then employed the smallest LLM (LLaMA3-8B) to perform the extraction on the full table.

5.3 Evaluation Procedure: LLM-as-a-Judge

Evaluating the correctness of extracted claims poses unique challenges due to the heterogeneity of the source data and the generative nature of large language models. Models may paraphrase, restructure, or hallucinate content, making it difficult to directly match extracted claims to those in the ground truth using exact string comparison. For example, from table in Figure 1 while the ground truth reports *Syntactical accuracy* as measure, some models extracted Acc_{syn}. Even using other techniques is rendered difficult by the diversity of actual information discussed in each paper for each topic. In addition, the large number of claims in the benchmark makes a human-based comparison process very time consuming.

To address these challenges, we adopt an LLM-as-a-Judge [52] approach. Specifically, we use an LLM (Claude 3–5 Sonnet) to evaluate whether an extracted claim matches a claim in the ground truth. Each extracted claim is compared to all the ground truth claims. Once a match is found, both the extracted and the ground truth claims are removed from further comparison. Note that we do not provide to the LLM-as-a-judge any constraints on the elements of the claims that must align to determine a match, thus even partial matches may be identified by the LLM. This matching procedure allows us to compute precision, recall, and F1-score scores for each experiment based on the number of matches found between the extracted and ground truth claims. For each pair of matching extracted and ground truth claims, the same LLM-as-a-judge approach is used to evaluate precision, recall and F1-score scores for specifications.

It is worth observing that this approach penalizes LLMs that extract the same claim multiple times with different representations, reducing precision, and LLMs that extract independent measures in one claim, reducing recall. As we will discuss later, this issue arises particularly in the medical domain, where the LLM, instead of generating a single claim with a vector of measurements, produces multiple claims, one for each individual element of the vector.

To assess the reliability of the evaluation process, we performed an additional manual check on a sample of the results of the LLM as-a-judge evaluation. Across all the experiments, we randomly selected 100 samples of pairs of claims labeled as matched, 100 samples labeled as non-matching, 100 pairs of specifications labeled as matching, 100 labeled as non-matching. Upon manual inspection, we found that all pairs labeled as matched were correct. One claim was incorrectly labeled as non-matching. The extracted claim included all but one specification, which was present in a paragraph but not extracted, but the model retrieved two measures from distinct table columns while in the ground truth there was only one of those measures. As a consequence the LLM labeled it as not match.Based on this manual inspection, the estimated accuracy of our LLM-based evaluation approach is 99.5% for claim matching and 100% for specification equivalence in the sampled cases.

5.4 Claim Extraction: Results

For each topic, we conduct a total of nine experiments. Eight of these use the direct extraction pipelines (zero-shot and one-shot) and the four LLMs. The ninth experiment uses the bootstrap extraction pipeline, where the best-performing LLM from the direct extraction experiments is selected to generate the in-context examples, and Llama3-8B is used to extract claims from the full table.

Table 3 shows the results for each topic and each configuration. [5] We observe that the 1-shot setting consistently outperforms the 0-shot setting across all domains and models, with a margin of over 3.75%, and often with a much wider margin. This suggests that providing a single example significantly improves the ability of LLMs to extract claims effectively.

Performance across domains reveals a consistent trend: models tend to achieve higher precision and recall in computer science domains compared to those in the medical domain. One explanation for this discrepancy might have to do with the nature of scientific reporting. In computer science publications, performance metrics are often simpler and typically can be understood independently from each other. In contrast, medical publications report more complex metrics. These often include the results of statistical analyses, such as confidence intervals, hazard ratios, and significance indicators, which often require multiple measures to be included in the same vector, as we illustrated in the example shown in Figure 2. Analyzing the results, we observed that in many cases, measures that need to be reported in the same vector are extracted as separate claims. These erroneous extractions lead to imprecise claims and, as discussed in Section 5.3, significantly penalizes precision.

Precision and recall show some notable trends. GPT-4o consistently achieves the highest precision and F1-score across most topics and settings, particularly in the 1-shot configuration. On the other hand, Llama3-70B tends to perform better in terms of recall, particularly in the 1-shot setting, indicating its strength in retrieving a broader set of relevant claims. The only exception for this is

[5] We tested the direct extraction (1-shot) pipeline on this table with all models and obtained 100% average claims precision and 49% average claims recall (Llama3-70B extracted measure-outcome vectors correctly, while the other models extracted Precision, Recall and F1-score in the same claim for each row).

Table 3: Claim Extraction Performance of different LLMs across various extraction pipelines on each of the four topics. In bold are highlighted the results of configurations that performed best on precision, recall, and F1-score for each topic. All the experiments were invoked with temperature set to 0.1.

Topic	Pipeline	LLM	Claims		
			Precision	Recall	F1-score
Text-to-SQL	Direct Extraction (0-shot)	GPT-4o	0.81	0.80	0.81
		Claude3-5 sonnet	0.78	0.60	0.68
		Llama3-70B	0.77	0.67	0.71
		Llama3-8B	0.31	0.38	0.34
	Direct Extraction (1-shot)	GPT-4o	**0.88**	0.82	**0.85**
		Claude3-5 sonnet	0.88	0.79	0.83
		Llama3-70B	0.79	**0.83**	0.83
		Llama3-8B	0.28	0.43	0.34
	Bootstrap (1-shot)	GPT-4o + Llama3-8B	0.67	0.71	0.69
Entity resolution	Direct Extraction (0-shot)	GPT-4o	0.84	0.75	0.79
		Claude3-5 sonnet	0.80	0.71	0.75
		Llama3-70B	0.80	0.71	0.75
		Llama3-8B	0.34	0.36	0.35
	Direct Extraction (1-shot)	GPT-4o	**0.94**	0.97	**0.95**
		Claude3-5 sonnet	0.91	0.76	0.83
		Llama3-70B	0.91	**0.98**	0.94
		Llama3-8B	0.28	0.75	0.41
	Bootstrap (1-shot)	GPT-4o + Llama3-8B	0.43	0.72	0.54
Pancreatic cancer	Direct Extraction (0-shot)	GPT-4o	0.78	0.49	0.60
		Claude3-5 sonnet	0.69	0.47	0.56
		Llama3-70B	0.64	0.52	0.57
		Llama3-8B	0.49	0.35	0.41
	Direct Extraction (1-shot)	GPT-4o	**0.93**	0.49	0.64
		Claude3-5	0.88	0.51	0.64
		Llama3-70B	0.83	**0.53**	**0.65**
		Llama3-8B	0.48	0.26	0.34
	Bootstrap (1-shot)	Llama3-70b + Llama3-8B	0.66	0.40	0.50
HIV	Direct Extraction (0-shot)	GPT-4o	0.62	0.55	0.58
		Claude3-5 sonnet	0.54	0.60	0.57
		Llama3-70B	0.46	0.49	0.47
		Llama3-8B	0.37	0.33	0.35
	Direct Extraction (1-shot)	GPT-4o	**0.67**	**0.59**	**0.63**
		Claude3-5 sonnet	0.60	0.51	0.55
		Llama3-70B	0.67	0.58	0.62
		Llama3-8B	0.38	0.44	0.40
	Bootstrap (1-shot)	GPT-4o + Llama3-8B	0.39	0.54	0.45

the *HIV* topic, where GPT-4o and Llama3 70b present comparable performances (0.59 vs 0.58).

A final observation concerns the Bootstrap pipeline. While the use of an in-context example generated by a larger language model offers considerable improvement over the zero-shot setting of the corresponding smaller LLM (LLama3-8B), the overall performance remains lower than that of the 1-shot direct extraction pipeline (using, e.g., GPT-4o). This suggests that at least in its current configuration, the LLama3-8B model may still face challenges in accurately extracting claims, even when provided with relevant examples. These results highlight the importance of model capabilities and the challenges of balancing cost of model invocation and the quality of the extracted results.

5.5 Specification Extraction: Results

Merely observing the performance of the various pipelines w.r.t. claim extraction leaves open the question how each of them performs w.r.t. specification extraction. The reason is that each claim

Table 4: *Specifications* extraction performance metrics (Precision, Recall, F1), with average specifications per matched extracted claims and respective claimed ground truth claim, and data origin percentage. All results are from Direct Extraction (1-shot) pipelines with the best performing LLM for each topic.

Topic	Specifications			Avg #specs in matched		Origin	
	Precision	Recall	F1	Extracted claim	GT claim	%table	%unstructured
Text-to-SQL	0.55	0.67	0.60	4.7	4.3	66%	34%
Entity resolution	0.86	0.92	0.89	3.5	3.5	27%	73%
Pancreatic cancer	0.53	0.68	0.59	4.9	4.7	71%	29%
HIV	0.50	0.83	0.63	4.9	4.3	45%	55%

extraction metric (precision, recall, F1) encapsulates the performance of a pipeline on the extraction of subject (set of specifications) and outcome (measures and values). To evaluate the quality of the extracted contextual information, we introduce a second evaluation step based on the precision and recall of specification elements (i.e., subjects). This evaluation was limited to specifications from extracted claims that matched a claim in the ground truth. Initially, we applied a simple heuristic to identify direct matches between elements of the extracted claims and ground truth claims. For unmatched elements, we used a large language model to assess semantic equivalence, accounting for variations in wording and effectively mapping specifications. To evaluate whether two elements are equivalent the large language model is prompted with both elements and the context of the table (hence, the table itself, its caption and footnotes and paragraph). Importantly, recall was computed only on ground truth claims that had a corresponding match among the extracted claims rather than the whole set of ground truth claims. This restriction allows us to isolate and accurately evaluate the results of the extracted specification elements only where the claim was correctly identified. Including unmatched claims in the recall calculation would conflate errors in claim extraction with errors in specification extraction, making it difficult to distinguish whether a low recall is due to missed claims or incomplete specification extraction. By focusing only on matched claims, we can better assess the granularity of specification extraction independently of claim matching performance.

In Table 4, we present the evaluation results for specification extraction, focusing on the top-performing pipelines for each topic. For each topic, we also report the average number of specifications found in matched extracted claims and ground truth claims. Additionally, we include the percentage of specifications originating from the table content (i.e., the HTML structure of the table) versus the unstructured text, which includes captions, footnotes, and paragraphs.

The best performance is observed in the Entity Resolution topic, with both precision and recall exceeding 85%. As shown in Table 1 and Table 2, the tables in this topic are primarily relational or cross-tables, and are characterized by the fewest average number of rows and a relatively small number of columns but the longest paragraphs. Based on statistics, two out of three specifications come from unstructured text, confirming that models are able to extract relevant information from captions and paragraphs as well. Furthermore, footnotes—an important source of supplementary information—are much less commonly used in computer science tables compared to those in the medical domain.

In contrast, the pancreatic cancer topic exhibits the lowest performance. While its dataset contains some of the largest tables in terms of row and column count, a more notable factor is its high proportion of complex, nested tables—both in headers and indexes. Approximately 70% of the specifications are derived from tables, with only 30% coming from unstructured text. This suggests that the overall complexity and structure of the tables may significantly hinder the accurate extraction of specifications.

The HIV topic offers an almost reversed pattern: a majority of specifications originate from unstructured text. Dataset statistics reveal that this topic has the highest volume of characters in footnotes, which often provide rich experimental context—for instance, describing population characteristics used in the reported results. This reflects a broader trend in medical literature, where footnotes are more integral to interpreting tabular data (for example, by reporting variables used in computing hazard ratios, or the actual method used to compute a statistical measure). Results suggests that information scattered in text are correctly identified and successfully extracted as specifications, while a complex table layout -such in the case of pancreatic cancer tables in our dataset- reduce the capacity of the model to identify the specifications.

Finally, we offer a reflection on the precision results. Across topics, the precision for specifications extraction is around 50% (except for entity resolution). This relatively low precision indicates that approximately half of the extracted specifications in matched claims are either incorrect or not meaningful in the context of specification evaluation. For example, manual inspection reveals that models frequently extract metadata information like table references (e.g., `|table, 4|`, or `|source, table 4|`), metric labels (e.g., `|metric type, accuracy|`), or even specific results (e.g., `|p-value, 0.005|`).

5.6 Costs and Inference Time

For each LLM, in Table 5 we report a prospective of the costs for the 0-shot and 1-shot direct extraction on the tables of the whole dataset. In particular, we report the total monetary costs, amount of output tokens,[6] and inference time in minutes. It is worth noting that, since the models are hosted by a third-party service, the inference time might change due to update on the architecture to the third-party service side (for example, they might improve the inference time).

[6]The number of input tokens is constant.

Table 5: Average costs, inference time and output tokens amount, for one experiment, aggregated by LLM.

LLM	Costs	Inference Time	Output Tokens
GPT-4o	$0.130	5 minutes	23k
Llama3-8b	$0.0016	2.6 minutes	18k
Llama3-70b	$0.103	9.3 minutes	22k
claude3-5 sonnet	$0.216	10.3 minutes	16k

Table 6: Costs and number of output token comparison between Direct Extraction and Bootstrap pipelines across topics for the mentioned experiment. For *text-to-SQL, entity resolution* and *HIV* the first pipeline runs with GPT-4o, while the bootstrap runs with GPT-4o for the first step and Llama3-8b for the second step. For *pancreatic cancer*, GPT-4o is replaced with Llama3-70b.

Topic	1-shot		Bootstrap	
	Output tokens	Costs ($)	Output tokens	Costs ($)
Text-to-SQL	25k	$0.11	(4k + 19k)	$0.032
Entity resolution	21k	$0.092	(2k + 17k)	$0.008
HIV	23k	$0.101	(4k + 19k)	$0.032
Pancreatic cancer	23k	$0.105	(8k + 45k)	$0.0715

Some conclusions can be drawn considering the costs and the results. GPT-4o offers the best balance of performance, cost, and inference time, making it the most efficient and effective choice. Llama3-70b provides similar quality at a slightly lower cost but with longer inference time, but has the advantage of possibly being run locally, further reducing the costs. The quality of the results produced by Claude3-5 sonnet is similar, but it is more costly and slower. Llama3-8b is extremely fast and cheap but its quality performance is poor.

Table 6 shows the comparison of output token usage and costs of bootstrap extraction against the extractions made by the same models used in the direct extraction experiments. The input tokens for Bootstrap pipelines are roughly double the number of tokens used in Direct Extraction pipelines (as the context of the table needs to be prompted two times). The output tokens amount is reduced by at least 65% in the worst case (pancreatic cancer) and up to 90% in the best case (entity resolution).

6 CONCLUSIONS AND FUTURE WORK

Fine-grained claim extraction from scientific papers can enable in-depth analysis of the literature, uncover the causes of apparent contradictions across studies, and inspire new research directions. However, this task is particularly challenging, due to the syntactic and semantic heterogeneity in how findings are represented across papers. While large language models (LLMs) show promise in addressing these challenges, our findings suggest that more refined methodologies are necessary to effectively capture the full spectrum of relevant information.

To evaluate the generality of our results across domains and topics, we plan to improve the benchmark, expanding both the dataset and its corresponding ground truth including more papers and tables for additional domains and topics.

Our results show that smaller models such as LLaMA3-8B, despite lower baseline performance, benefit significantly from our bootstrap-based extraction pipelines. This opens up promising opportunities to develop more lightweight yet effective claim extraction systems. We also plan to explore fine-tuning strategies for both LLaMA3-8B and LLaMA3-70B to enhance their specialization in the claim extraction task.

One of the long-term goals of the DESIREE project is to enable fine-grained, data-driven analyses that accelerate scientific discovery. Achieving this vision requires overcoming significant challenges, particularly in aligning and querying claims across a highly heterogeneous body of literature. The variability in lexicons, representations, and value ranges across papers complicates the process of matching claims. For example, different studies may use varying terminology to describe the same concepts, or include a wide range of measures where only a subset is relevant to a query.

While claims frequently appear in tables, they are also conveyed through other visualizations such as plots and charts. Recent advances in deep learning are enabling the extraction of structured data from these visual formats. Tools like DEPLOT [36] and MATCHA [37], both based on the PIX2STRUCT [32] framework, can effectively convert plots and charts into tabular representations: we plan to leverage these tools to extend the extraction of claims from papers including those presented in visual format. Finally, we plan to extend our evaluation by leveraging the capabilities of recent multimodal LLMs, which open new opportunities for advancing the claim extraction task.

ACKNOWLEDGMENTS

The second author's research was supported in part by a grant from the Natural Sciences and Engineering Research Council of Canada (Grant Number RGPIN-2020-05408).

REFERENCES

[1] 2025. Consensus. https://consensus.app/ Accessed: 2025-05-28.
[2] 2025. ScholarAI. https://scholarai.io/ Accessed: 2025-05-28.
[3] 2025. Scite. https://scite.ai Accessed: 2025-05-28.
[4] Eugene Ahn and Hyun Kang. 2018. Introduction to systematic review and meta-analysis. *Korean Journal of Anesthesiology* 71, 2 (Apr 2018), 103–112. https://doi.org/10.4097/kjae.2018.71.2.103
[5] Sören Auer, Allard Oelen, Muhammad Haris, Markus Stocker, Jennifer D'Souza, Kheir Eddine Farfar, Lars Vogt, Manuel Prinz, Vitalis Wiens, and Mohamad Yaser Jaradeh. 2020. Improving access to scientific literature with knowledge graphs. *Bibliothek Forschung und Praxis* 44, 3 (2020), 516–529.
[6] 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)*. 3615–3620.
[7] Galal M Binmakhashen and Sabri A Mahmoud. 2019. Document layout analysis: a comprehensive survey. *ACM Computing Surveys (CSUR)* 52, 6 (2019), 1–36.
[8] Katarina Boland, Pavlos Fafalios, Andon Tchechmedjiev, Stefan Dietze, and Konstantin Todorov. 2022. Beyond facts – a survey and conceptualisation of claims in online discourse analysis. *Semantic Web* 13 (2022), 793–827. https://doi.org/10.3233/SW-212838 5.

[9] Arthur Brack, Jennifer D'Souza, Anett Hoppe, Sören Auer, and Ralph Ewerth. 2020. Domain-independent extraction of scientific concepts from research articles. In *European Conference on Information Retrieval*. Springer, 251–266.

[10] Defne Circi, Ghazal Khalighinejad, Anlan Chen, Bhuwan Dhingra, and L Brinson. 2024. Extracting Materials Science Data from Scientific Tables. In *ACL 2024 Workshop Language+ Molecules*.

[11] John Dagdelen, Alexander Dunn, Sanghoon Lee, Nicholas Walker, Andrew S. Rosen, Gerbrand Ceder, Kristin A. Persson, and Anubhav Jain. 2024. Structured information extraction from scientific text with large language models. *Nature Communications* 15 (feb 2024), 1418. https://doi.org/10.1038/s41467-024-XXXX-X Published: 15 February 2024.

[12] Jianxin Deng, Gang Liu, Ling Wang, Jiawei Liang, and Bolin Dai. 2025. An efficient extraction method of journal-article table data for data-driven applications. *Information Processing & Management* 62, 3 (2025), 104006.

[13] Yunyang Deng, Junjie Huang, and Martin CS Wong. 2022. Associations between six dietary habits and risk of hepatocellular carcinoma: a Mendelian randomization study. *Hepatology communications* 6, 8 (2022), 2147–2154.

[14] Danilo Dessí, Francesco Osborne, Diego Reforgiato Recupero, Davide Buscaldi, and Enrico Motta. 2022. SCICERO: A deep learning and NLP approach for generating scientific knowledge graphs in the computer science domain. *Knowledge-Based Systems* 258 (2022), 109945.

[15] Frank S Fan. 2022. Coffee reduces the risk of hepatocellular carcinoma probably through inhibition of NLRP3 inflammasome activation by caffeine. *Frontiers in Oncology* 12 (2022), 1029491.

[16] Michael Färber and David Lamprecht. 2023. Linked Papers With Code: The Latest in Machine Learning as an RDF Knowledge Graph. *arXiv preprint arXiv:2310.20475* (2023).

[17] Liangcai Gao, Yilun Huang, Hervé Déjean, Jean-Luc Meunier, Qinqin Yan, Yu Fang, Florian Kleber, and Eva Lang. 2019. ICDAR 2019 competition on table detection and recognition (cTDaR). In *2019 International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 1510–1515.

[18] Jennifer B Goldstein, Chad Tang, Kenneth R Hess, David Hong, Vivek Subbiah, Filip Janku, Siqing Fu, Daniel D Karp, Aung Naing, Apostolia Maria Tsimberidou, et al. 2017. Outcomes of phase I clinical trials for patients with advanced pancreatic cancer: update of the MD Anderson Cancer Center experience. *Oncotarget* 8, 50 (2017), 87163.

[19] Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2021. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)* 3, 1 (2021), 1–23.

[20] 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*. Association for Computational Linguistics.

[21] Mohamad Yaser Jaradeh, Allard Oelen, Kheir Eddine Farfar, Manuel Prinz, Jennifer D'Souza, Gábor Kismihók, Markus Stocker, and Sören Auer. 2019. Open research knowledge graph: next generation infrastructure for semantic scholarly knowledge. In *Proceedings of the 10th International Conference on Knowledge Capture*. 243–246.

[22] Yuna Jeong and Eunhui Kim. 2022. SciDeBERTa: Learning DeBERTa for Science Technology Documents and Fine-Tuning Information Extraction Tasks. *IEEE Access* 10 (2022), 60805–60813.

[23] Tianwen Jiang, Tong Zhao, Bing Qin, Ting Liu, Nitesh V Chawla, and Meng Jiang. 2019. The role of" condition" a novel scientific knowledge graph representation and construction model. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1634–1642.

[24] Antonio Jimeno Yepes, Peter Zhong, and Douglas Burdick. 2021. ICDAR 2021 competition on scientific literature parsing. In *Document Analysis and Recognition–ICDAR 2021: 16th International Conference, Lausanne, Switzerland, September 5–10, 2021, Proceedings, Part IV 16*. Springer, 605–617.

[25] Shaoyue Jin and Youjin Je. 2024. Coffee consumption and risk of lung cancer: a meta-analysis of prospective cohort studies. *Nutrition and Cancer* 76, 7 (2024), 552–562.

[26] Marcin Kardas, Piotr Czapla, Pontus Stenetorp, Sebastian Ruder, Sebastian Riedel, Ross Taylor, and Robert Stojnic. 2020. AxCell: Automatic Extraction of Results from Machine Learning Papers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 8580–8594.

[27] Pratik Kayal, Mrinal Anand, Harsh Desai, and Mayank Singh. 2021. ICDAR 2021 competition on scientific table image recognition to LaTeX. In *Document Analysis and Recognition–ICDAR 2021: 16th International Conference, Lausanne, Switzerland, September 5–10, 2021, Proceedings, Part IV 16*. Springer, 754–766.

[28] Oliver John Kennedy, Paul Roderick, Ryan Buchanan, Jonathan Andrew Fallowfield, Peter Clive Hayes, and Julie Parkes. 2017. Coffee, including caffeinated and decaffeinated coffee, and the risk of hepatocellular carcinoma: a systematic review and dose–response meta-analysis. *BMJ open* 7, 5 (2017), e013739.

[29] Madian Khabsa, Ahmed Elmagarmid, Ihab Ilyas, Hossam Hammady, and Mourad Ouzzani. 2016. Learning to identify relevant studies for systematic reviews using random forest and external information. *Machine Learning* 102 (2016), 465–482.

[30] Eunhui Kim, Yuna Jeong, and Myung-Seok Choi. 2023. Medibiodeberta: Biomedical language model with continuous learning and intermediate fine-tuning. *IEEE Access* 11 (2023), 141036–141044.

[31] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics* 36, 4 (2020), 1234–1240.

[32] Kenton Lee, Mandar Joshi, Iulia Turc, Hexiang Hu, Fangyu Liu, Julian Eisenschlos, Urvashi Khandelwal, Peter Shaw, Ming-Wei Chang, and Kristina Toutanova. 2023. Pix2Struct: Screenshot Parsing as Pretraining for Visual Language Understanding. arXiv:2210.03347 [cs.CL]

[33] Sihang Li, Jin Huang, Jiaxi Zhuang, Yaorui Shi, Xiaochen Cai, Mingjun Xu, Xiang Wang, Linfeng Zhang, Guolin Ke, and Hengxing Cai. [n.d.]. SciLitLLM: How to Adapt LLMs for Scientific Literature Understanding. In *Neurips 2024 Workshop Foundation Models for Science: Progress, Opportunities, and Challenges*.

[34] Xiao-Hui Li, Fei Yin, He-Sen Dai, and Cheng-Lin Liu. 2022. Table structure recognition and form parsing by end-to-end object detection and relation parsing. *Pattern Recognition* 132 (2022), 108946.

[35] Yuliang Li, Jinfeng Li, Yoshihiko Suhara, AnHai Doan, and Wang-Chiew Tan. 2020. Deep entity matching with pre-trained language models. *PVLDB* 14, 1 (sep 2020), 50–60. https://doi.org/10.14778/3421424.3421431

[36] Fangyu Liu, Julian Martin Eisenschlos, Francesco Piccinno, Syrine Krichene, Chenxi Pang, Kenton Lee, Mandar Joshi, Wenhu Chen, Nigel Collier, and Yasemin Altun. 2022. DePlot: One-shot visual language reasoning by plot-to-table translation. *arXiv preprint arXiv:2212.10505* (2022).

[37] Fangyu Liu, Francesco Piccinno, Syrine Krichene, Chenxi Pang, Kenton Lee, Mandar Joshi, Yasemin Altun, Nigel Collier, and Julian Martin Eisenschlos. 2023. MatCha: Enhancing Visual Language Pretraining with Math Reasoning and Chart Derendering. arXiv:2212.09662 [cs.CL]

[38] Weizheng Lu, Jing Zhang, Ju Fan, Zihao Fu, Yueguo Chen, and Xiaoyong Du. 2025. Large language model for table processing: A survey. *Frontiers of Computer Science* 19, 2 (2025), 192350.

[39] Xinyuan Lu, Liangming Pan, Qian Liu, Preslav Nakov, and Min-Yen Kan. [n.d.]. SCITAB: A Challenging Benchmark for Compositional Reasoning and Claim Verification on Scientific Tables. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

[40] Shrey Mishra, Lucas Pluvinage, and Pierre Senellart. 2021. Towards Extraction of Theorems and Proofs in Scholarly Articles. In *Proc. DocEng*. Limerick, Ireland.

[41] Ralph Peeters and Christian Bizer. 2021. Dual-Objective Fine-Tuning of BERT for Entity Matching. *PVLDB* 14, 10 (jun 2021), 1913–1921. https://doi.org/10.14778/3467861.3467878

[42] Maciej P Polak and Dane Morgan. 2024. Extracting accurate materials data from research papers with conversational language models and prompt engineering. *Nature Communications* 15, 1 (2024), 1569.

[43] Shraman Pramanick, Rama Chellappa, and Subhashini Venugopalan. [n.d.]. SPIQA: A Dataset for Multimodal Question Answering on Scientific Papers. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

[44] Yuan Sui, Mengyu Zhou, Mingjie Zhou, Shi Han, and Dongmei Zhang. 2024. Table meets llm: Can large language models understand structured table data? a benchmark and empirical study. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*. 645–654.

[45] Raymon van Dinter, Bedir Tekinerdogan, and Cagatay Catal. 2021. Automation of systematic literature reviews: A systematic literature review. *Information and Software Technology* 136 (2021), 106589. https://doi.org/10.1016/j.infsof.2021.106589

[46] David Wadden, Kyle Lo, Bailey Kuehl, Arman Cohan, Iz Beltagy, Lucy Lu Wang, and Hannaneh Hajishirzi. 2022. SciFact-Open: Towards open-domain scientific claim verification. In *Findings of the Association for Computational Linguistics: EMNLP 2022*. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 4719–4734. https://doi.org/10.18653/v1/2022.findings-emnlp.347

[47] Chenglong Wang, Kedar Tatwawadi, Marc Brockschmidt, Po-Sen Huang, Yi Mao, Oleksandr Polozov, and Rishabh Singh. 2018. Robust Text-to-SQL Generation with Execution-Guided Decoding. arXiv:1807.03100 [cs.CL] https://arxiv.org/abs/1807.03100

[48] Shu Wang, Xiang Li, Yue Yang, Jingping Xie, Mingyue Liu, Ya Zhang, Yingshi Zhang, and Qingchun Zhao. 2021. Does coffee, tea and caffeine consumption reduce the risk of incident breast cancer? A systematic review and network meta-analysis. *Public Health Nutrition* 24, 18 (2021), 6377–6389.

[49] Jian Xu, Sunkyu Kim, Min Song, Minbyul Jeong, Donghyeon Kim, Jaewoo Kang, Justin F Rousseau, Xin Li, Weijia Xu, Vetle I Torvik, et al. 2020. Building a PubMed knowledge graph. *Scientific data* 7, 1 (2020), 205.

[50] Alexandros Zeakis, George Papadakis, Dimitrios Skoutas, and Manolis Koubarakis. 2023. Pre-trained embeddings for entity resolution: an experimental analysis. *Proceedings of the VLDB Endowment* 16, 9 (2023), 2225–2238.

[51] Bowen Zhao, Changkai Ji, Yuejie Zhang, Wen He, Yingwen Wang, Qing Wang, Rui Feng, and Xiaobo Zhang. 2023. Large Language Models are Complex Table

Parsers. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. 14786–14802.

[52] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P Xing, et al. 2023. Judging LLM-as-a-judge with MT-bench and Chatbot Arena. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*. 46595–46623.

[53] Wenzhen Zhu, Negin Sokhandan, Guang Yang, Sujitha Martin, and Suchitra Sathyanarayana. 2022. DocBed: A multi-stage OCR solution for documents with complex layouts. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 12643–12649.