Integrating Materials and Manufacturing Innovation (2024) 13:669–687
https://doi.org/10.1007/s40192-024-00362-6

THEMATIC SECTION: HARNESSING THE POWER OF MATERIALS DATA

How Well Do Large Language Models Understand Tables in Materials Science?

Defne Circi[1] · Ghazal Khalighinejad[2] · Anlan Chen[1] · Bhuwan Dhingra[2] · L. Catherine Brinson[1]

Received: 16 March 2024 / Accepted: 10 May 2024 / Published online: 19 July 2024

Abstract

Advances in materials science require leveraging past findings and data from the vast published literature. While some materials data repositories are being built, they typically rely on newly created data in narrow domains because extracting detailed data and metadata from the enormous wealth of publications is immensely challenging. The advent of large language models (LLMs) presents a new opportunity to rapidly and accurately extract data and insights from the published literature and transform it into structured data formats for easy query and reuse. In this paper, we build on initial strategies for using LLMs for rapid and autonomous data extraction from materials science articles in a format curatable by materials databases. We presented the subdomain of polymer composites as our example use case and demonstrated the success and challenges of LLMs on extracting tabular data. We explored different table representations for use with LLMs, finding that a multimodal model with an image input yielded the most promising results. This model achieved an accuracy score of 0.910 for composition information extraction and an F_1 score of 0.863 for property name information extraction. With the most conservative evaluation for the property extraction requiring exact match in all the details, we obtained an F_1 score of 0.419. We observed that by allowing varying degrees of flexibility in the evaluation, the score can increase to 0.769. We envision that the results and analysis from this study will promote further research directions in developing information extraction strategies from materials information sources.

Keywords Large language models · Information extraction from tables · Polymer composites · Materials informatics

Introduction

In this paper, we examine the effect of using different input types for information extraction from tables in the polymer composite domain which will help scientists and engineers to easily find information without attempting to search through millions of relevant articles. It is important to connect data from different resources in materials science, as existing data directs future discoveries and research. Peer-reviewed research publications currently form the official source of reliable information on a large variety of materials research. However, due to their unstructured nature and highly unique writing and presentation styles, it is difficult to utilize the vast majority of materials data locked in these journal articles and reports [1]. Moreover, sifting through the articles and determining the structure, processing steps, and properties of each material sample is tedious, time-consuming, and error prone. Individuals cannot possibly read, understand and utilize the vast literature even in small subfields. Therefore, materials understanding and discoveries are handicapped. In this paper, we examine the effect of using different input types for information extraction from tables in the polymer composite domain which will help scientists or engineers to easily find information without attempting to search through millions of relevant articles.

✉ L. Catherine Brinson
cate.brinson@duke.edu

Defne Circi
defne.circi@duke.edu

Ghazal Khalighinejad
ghazal.khalighinejad@duke.edu

Anlan Chen
anlan.chen@duke.edu

Bhuwan Dhingra
bdhingra@cs.duke.edu

[1] Department of Mechanical Engineering and Materials Science, Duke University, Durham, NC 27708, USA

[2] Department of Computer Science, Duke University, Durham, NC 27708, USA

It is important to connect data from different resources in materials science, as existing data directs future discoveries and research.

To connect data from different resources, robust structured data platforms to store, visualize, and analyze materials data are critical for downstream tasks of material discovery, process optimization and virtual metrology/characterization [2], as recently demonstrated by Szymanski et al [3]. NanoMine, part of MaterialsMine, focuses on collecting experimental data from literature on the specific material system of polymer composites that meets these needs. To date, NanoMine data have been collected manually and stored in an accessible and queryable knowledge graph framework [4]. However, due to challenges mentioned above, it is impractical to manually curate the data from more than 1 million published papers even in this relatively small subfield.

Therefore, automation of the data curation process has gained increasing attention to enable rapid growth of a robust repository of prior published data [2, 5–11]. Leveraging natural language processing (NLP) and large language models (LLMs) can make vital material information such as material identification, composition, properties, or experimental details readily available in a machine-readable format [12–17]. Of the initial explorations of LLMs for information extraction from the scientific literature, most have focused on extraction from text only.

In recent works, we have also examined the use of LLMs to extract information from the text portions of materials papers [18, 19]. In these work, it became apparent that information we can collect from text only is limited. In fact, in another preliminary analysis of materials science papers, Gupta et al. found that 85% of compositions and their associated properties are reported only in tables [20]. Thus, tables in the materials science domain contain rich information about the properties and composition of materials. Indeed, tables that contain composition and property information are available not only in the polymer composite field but in all materials subfields, and other fields including medicine, food and nutrition [20]. For this reason, information extraction from tables will be crucial in automated data curation as structured data is often presented in both tabular and other visual formats [21].

There have been a number of efforts to extract data such as compositions and properties of materials from tables. Zhang et al [22] parsed the tables and their captions in XML/HTML files to extract fatigue data using a table extractor tool which was initially developed to extract zeolite synthesis data [23]. Using the same tool to obtain raw XML tables and captions, Gupta et al. introduced the task of composition information extraction from tables and developed a graph neural network based pipeline to extract glass compositions [20]. Zaki et al. found that using advanced LLMs such as GPT-4 to extract composition performed worse than a graph neural network model [24] and suggested task specific prompting strategies and fine-tuning in domain-specific datasets. Oka et al. [25] also used XML versions of the articles to extract limited number of target polymer properties from the literature.

This prior work indicates that while tables can be an excellent form to present condensed information for human readers, automated extraction of information from them remains a challenging task. Even for trivial tasks such as detecting the table size, LLMs can fail although they have some structural understanding of tables [26]. Additionally, some tables in published articles and reports are not available in XML format and are locked in PDF documents, necessitating table extracting and parsing approaches. Finally, it is important to develop flexible approaches to extract a broad set of properties and conditions from the wide variety of tables appearing in materials papers efficiently and reliably. Toward this end, we complement the structural understanding capabilities of the off-the-shelf LLMs, and their understanding of basic materials vocabulary, by using unique prompting and input types and evaluation strategies to explore viability of accurate and efficient knowledge extraction from tables in materials science papers.

Our study focuses on extracting polymer composite sample information, where each sample is identified by its composition (matrix name, filler name, composition fraction, filler surface treatment) and is associated with property (output) details. Polymer nano- and microcomposites are a class of materials consisting of a polymeric matrix material in which one or more types of nanoparticle or microparticle fillers are embedded. These fillers often have surface chemical groups added to them in order to improve the dispersion and properties of the resulting composite [27, 28]. Although the details of composition and processing leading to given output properties are still poorly understood, these materials show immense promise for numerous environmental and industrial applications [29]. Successful data extraction of composition and properties information together could allow for rapid new understanding and discoveries of functional composite materials.

We constructed a dataset with detailed, annotated ground truth from 37 tables and employed LLMs, namely GPT-4 Turbo and GPT-4 Turbo with vision, for named entity recognition and relation extraction tasks in tables in the materials science subdomain of polymer composites. Our study confronted several challenges, detailed in Sect. 3.1, that underscore the complexity of this task. These challenges included (a) layout challenges, such as merging multiple rows, (b) entity classification challenges, like differentiating between filler names and particle surface treatments (PST), and (c) relationship classification challenges, specifically in associating properties with their names and metrological

parameters. To explore the effectiveness of these models in extracting information from tables, we investigated how different input formats, namely image, OCR (optical character recognition), and structured formats such as CSV, influence the extraction process. This aspect of our research aligns with the findings of Sui et al. [26], who highlighted the impact of input formats on LLMs' ability to process complex data representations. Our findings contribute to the broader understanding of LLMs' capabilities in information extraction within scientific contexts, demonstrating both their potential and the challenges.

Methods

Article and Dataset Preparation

The data for this study consist of tables with information about polymer nano- and microcomposite samples. The articles were selected from MaterialsMine [30]. MaterialsMine contains 240 manually curated articles on nanocomposites with a total of 2,512 samples. The detailed sample information which includes properties, processing details and characterization methods is available in MaterialsMine. In this study, we focused on the composition and properties of the polymer nano- and microcomposites as extracted from tables. Two graduate students annotated 37 tables that came from 18 articles [31–48] to provide the ground truth. They read the same instructions that were provided to the LLMs. Within selected tables, each table has an average of approximately 4.9 samples with a minimum of 2 and a maximum of 15 samples for a total of 182 samples. On average, there are 3.1 properties in each table.

Choosing Inputs of Table Data

The next three subsections describe the approaches that were used for obtaining inputs of table data. All three methods leverage GPT-4, with one using GPT-4-Vision, and two approaches using digitization of the table, one in unstructured format using OCR, and the other using a structured tabular format. An example of different input types—image, OCR, structured format—and the ground truth for one of the samples of the same table can be seen in Fig. 1. In Sect. 3, the results obtained using these three input types are compared to understand the accuracy of data extraction from tables for polymer nano- and microcomposites.

GPT-4-Vision on Table Image

Initially, we manually captured screenshots of the articles, ensuring that these images include both the tables and their corresponding captions. An example can be seen in Fig. 1, Part a. To extract and interpret the data from these table images, we utilized GPT-4 Turbo with vision capabilities.

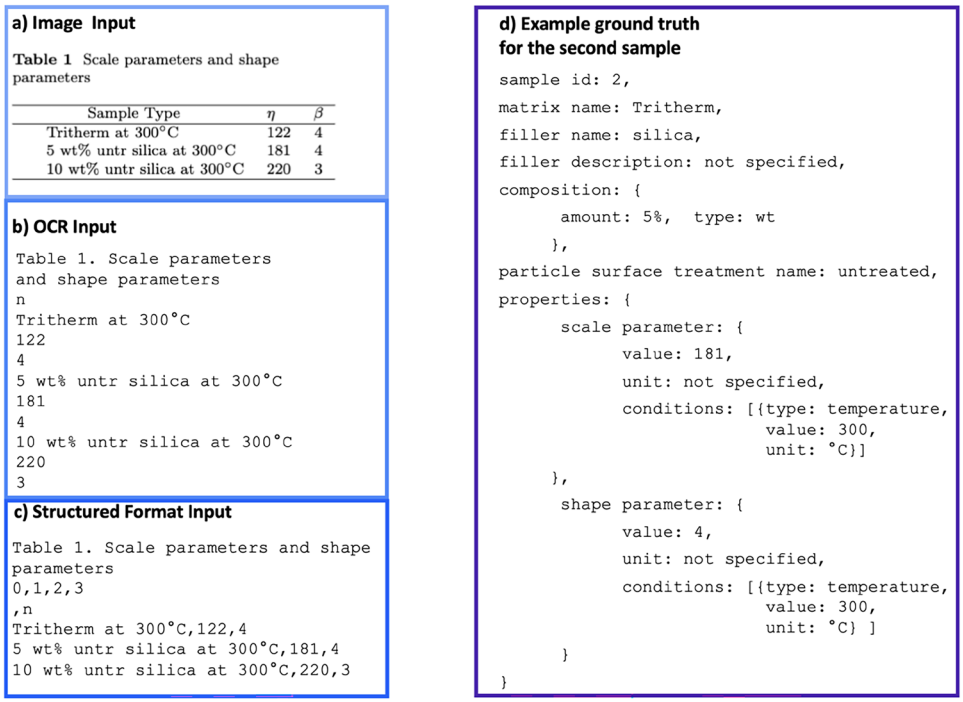

Fig. 1 Example of the three different input types: **a** GPT-4-Vision on sample table image (simulated table inspired by [36]) **b** GPT-4 on unstructured OCR given the table image in part a **c** GPT-4 on structured extracted table from the table image in part a **d** Example ground truth sample in JSON format

GPT-4 on Unstructured OCR Extraction from Table Image

For digitizing table content using OCR, we chose OCRSpace [49]. We provided image screenshots that include the captions to this platform, which enables the inclusion of table captions in the digitization process. However, it is important to note that this method does not preserve the original table structure. An example can be found in Fig. 1, Part b. Despite this limitation, OCRSpace's free API makes it a highly accessible and cost-effective solution for converting large volumes of data, with a rate limit of 500 requests within one day.

GPT-4 on Structured Table Output from PDF

We utilized the ExtractTable tool [50] to extract tabular data from images and convert it into a structured, standardized format. This process cost $0.04 per PDF page. This tool generates CSV files, efficiently structuring the table fields. However, it initially does not include table captions. Although the tool does not include table captions, it does maintain the tabular format which makes information extraction efficient. We generated two input files in structured format. The first one does not include the captions and the second one includes table captions that are manually added for fair comparison with the other input types which include table captions. Example can be found in Fig. 1, Part c.

Prompt Design

Based on our knowledge of polymer composite materials, the key differentiating fields are matrix, filler, composition and PST. Therefore, we picked this minimal set to define the composition information of the samples. For each sample there are sets of material properties reported in the tables, such as storage modulus, dielectric breakdown strength, and glass transition temperature. For each property, we captured the name of the property, its value, unit and, if reported, conditions at which the property is measured, such as temperature or pressure. Each condition has its own value and unit. Having the property details broken out as useful chunks (value, unit and conditions) is important because the extracted information can be easily added to the knowledge graph, in this case to MaterialsMine. In the original MaterialsMine curation template, properties are curated with their units, values and, optionally, "other details." The "other details" field can take any set of words or sentences and can refer to many types of information about the material system. In our study, we leveraged the capabilities of LLMs to be precise in this "other details" field and instruct the GPT to extract conditions associated with the property measurement, broken down into type, value and unit (if for example a property is measured at a specific temperature (type) of 120 (value) degrees C (unit)). In this process, we enabled querying properties based on conditions associated with the properties which had not been possible before. The importance of accurately extracting contextual information, particularly conditions, is underscored by Hira et al [51]. They discovered that 9% of the materials science tables in their analysis of 100 tables included conditions.

We utilized the strength of few-shot prompting, which can perform well without any training data. The models extract the entities and find the relations simultaneously. The prompt included a template JSON file to be filled along with a description of the task. Based on the selected option as specified in Sect. 2.2, the type of input table to be incorporated in the prompt is determined. The prompt, which can be found in Appendix A, also includes two example samples to make the outputs more consistent.

Evaluation

Given the necessity of evaluating a large number of papers, having an automated pipeline for evaluation is crucial. For information extraction from text, in our previous work [18], we noticed that evaluating the task of sample extraction has several challenges as it requires determining the most accurate alignment between each predicted sample and its corresponding ground truth sample, simultaneously taking into account all fields that describe the samples. One approach to address this issue is by utilizing a maximum weight bipartite matching algorithm, as outlined in our work on extracting composition information from text in full-length articles [19]. In our table extraction process, we observed that the sequence of samples extracted by the model usually aligns with the sequence in human-annotated data. Consequently, for evaluation purposes, we assumed a direct match in the ordering of samples, implying that each sample's position in the model output corresponds to the same position in the human-annotated dataset.

We implemented an automated system for evaluating the accuracy of sample information extracted from tables. This evaluation focused on comparing the extracted data-obtained through the different input methods (Sect. 2.2)-image-based extraction, OCR, and structured data extraction-against the set of annotated ground truth tables. We have considered several factors affecting the evaluation:

- **Data format and preprocessing:** Both predicted and ground truth files were structured as JSON files. During preprocessing, any comments within the predicted files were ignored (the part that comes after "//" until the new line) to ensure that only valid JSON data was processed.
- **Handling missing samples:** To understand the models' performance, we analyzed the output both by including

and excluding missing samples. This dual approach helps identify the source of differences in accuracy assessments.

a. **Including missing samples:** This method considers every sample in the ground truth. There were instances where ground truth tables contained more samples than the predictions provided, which we labeled as "missing samples." Samples in the tables with no predictions or with predictions that have incorrect syntax in the LLMs predictions were also labeled as "missing samples." Having no prediction means that there was no corresponding JSON(s) in the output for the missing sample(s). This rare error occurred in cases of incorrect syntax (such as extra commas) of the input table or the model gave an output similar to *"The example JSON provided does not match the table data given below it. We would need a complete table that includes all the necessary details as per the JSON template provided."* instead of filling in the JSON with the provided information and leaving the rest as "not specified." Missing samples in the extracted data are assigned a score of zero, providing insight into the predictions' completeness.

b. **Excluding missing samples:** Here, we focused only on the samples extracted, disregarding any that are missing. We also excluded the tables with no predictions or those with predictions that have incorrect syntax. We called these tables "invalid tables." This analysis method focused on the quality of the data that was actually extracted, disregarding the impact of the samples that were not extracted.

"sample id": 1,
"matrix name": "PP",
"filler name": "graphite",
"composition": {"amount": "2%", "type": "wt"},
"particle surface treatment name": "Vinylsilane"

Fig. 2 Composition information example; highlighted in bold are the four key values compared between ground truth and LLM prediction

Composition Information

Composition information is considered correct if the values in the matrix, filler, composition, and PST fields matched the ground truth sample. Here, the key values of the JSON files are fixed: matrix name, filler name, composition (amount and type) and PST. An example of this composition information can be seen in Fig. 2. Accuracy is used to evaluate the composition information. For each sample, we computed the accuracy by dividing the number of correct key-value pairs by the total number of key-value fields being checked. Then, we averaged these accuracies across all samples to find the accuracy of the table and report the average of all the tables.

The comparison functions are designed to be flexible in handling the following variations in the outputs as illustrated in Fig. 3:

- **Sub-string comparison:** In the case of PST, filler name and matrix name, we employed a sub-string comparison method, allowing either of the strings to be a subset of the other. For example, "vinylsilane treated" with "vinylsilane" and "epoxy resin" and "ether-bisphenol epoxy resin" are considered as matches.
- **Case-insensitive string comparisons:** For all non-numeric fields, the comparison was case-insensitive,

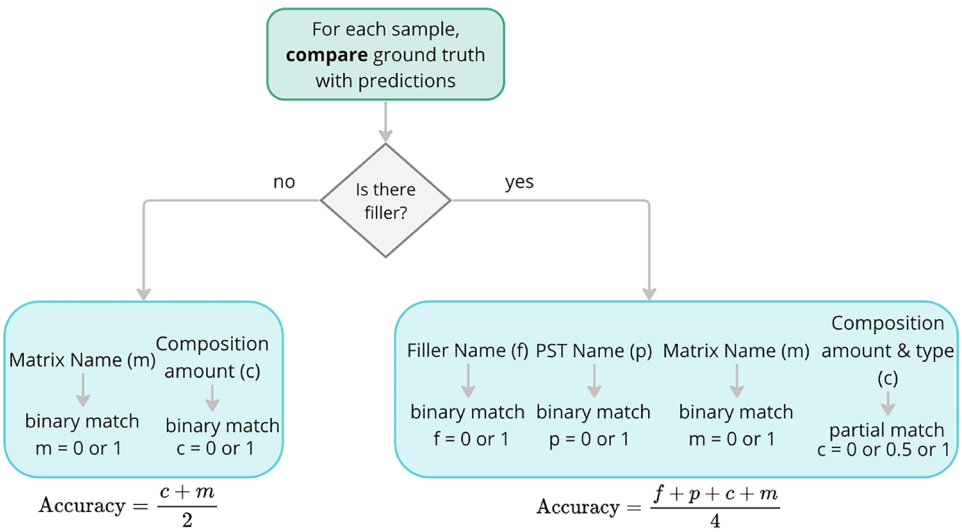

Fig. 3 Flowchart illustrating calculation of accuracy score for composition information considering matrix name (m), filler name (f), PST name (p) and composition (c). Note that some flexibility is allowed in matching m, f, and p in that sub-string matches are allowed

ensuring minor variations in text case did not skew the results.

- **Partial accuracy calculation:** We calculated partial accuracies for the composition field that includes "amount" and "type." This means that if some aspects of the field match, the comparison reflects this partial accuracy instead of treating it as a complete mismatch. To illustrate, consider an example composition entry represented in the following structure:

```
"composition": {
  "amount": "2%",
  "type": "wt"
}
```

In this example, the "amount" is specified as "2%," and the "type" is denoted as "wt," which stands for weight. Under our approach, if a data entry correctly matches the "amount" as "2%" but inaccurately identifies the "type," it is regarded as a partial match. Full correctness was assigned if both "amount" and "type" were correctly matched. Partial correctness was assigned if only one of the two components was correct. The composition was deemed incorrect if both components were inaccurate.

- **Handling numeric values and percentages:** For numeric values or percentages in the composition field, we first removed any whitespace and then converted these values into floats. We also chose to ignore the percentage symbol ("%") when comparing values.
- **Managing control samples (unfilled samples with composition value = 0.0):** If both the predicted and ground truth sample composition were 0, we did not consider filler name and PST in the accuracy calculation. See Fig. 3, the "no" branch.

Properties

Unlike the composition part where a small number of known fields consistently define the composite composition, property fields are not predefined and there are hundreds of possible properties that could be measured and reported. Each table can contain information of multiple properties that are studied in the article, and the exact number of these properties is also unknown. An example of the property field extraction and its variability can be seen in Fig. 4. While we could have provided the models with a list of possible properties, we elected to allow the models to interpret properties freely as a human curator would do, using the embedded material property understanding in the LLM. We evaluated the performance of GPT-4 using the F_1 metric for the extraction of properties for each sample. We take the average of F_1 scores for each sample in a given table and then report the average F_1 considering all the tables.

```
"properties":
{
  "Young's modulus": {"value": 1300, "unit": "MPa"},
  "Elongation at break": {"value": 8, "unit": "%"},
  "Crystallization temperature": { "value": 402, "unit": "K",
    "conditions": [{"type": "cooling speed", "value": -10, "unit": "K/min"}]}
}
```

Fig. 4 Property information example JSON illustrating a few of the wide variety of property names, parameters, and conditions appearing in tables containing material property information

Precision, recall and F_1 are defined as:

$$Precision = \frac{TP}{TP + FP}, \tag{1}$$

$$Recall = \frac{TP}{TP + FN}, \tag{2}$$

$$F_1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}, \tag{3}$$

where true positive (TP) is defined as the number of properties in the model output that are matched with a property in the ground truth, false positive (FP) is defined as properties in the model output that are not matched with a property in the ground truth and false negative (FN) is defined as the number of properties in the ground truth that are not matched with a property in the model output.

To match properties in the ground truth for each sample with the properties in the model output, we performed this analysis in two stages, where the first stage identified the match for the property name and the second stage considered the property value, unit and other conditions associated with the property measurement (for example, temperature at which the property was measured).

Stage 1: Considering property names to find property matches

In this initial stage, we first sought a match between the property names in the ground truth and the model prediction. Given the wide variation available in property names, we did not require an exact match, but used the Levenshtein distance as described below. For instance, a property annotated as "AC %decrease" in the ground truth data is referred to as "percentage decrease" in the predicted data in Fig. 9. In this first stage, the F_1 scores were only calculated based on the property name and did not include value, unit or conditions of the properties.

To compare the similarity between predicted and ground truth data, we utilized the Levenshtein distance method [52]. For each property in both datasets, we first generated a property name string by extracting keys from the property entities; these keys represent the property names.

We then calculated the normalized Levenshtein distance between these strings. To identify the closest match, we compared each predicted property name with all names in the ground truth dataset, selecting the ground truth property that exhibited the smallest Levenshtein distance, as long as it was below a predefined threshold = 0.6. For example, normalized Levenshtein distance between "AC %decrease" and "percentage decrease" is 0.4375. For unique matching, we maintained an index set of already matched ground truth properties. When a predicted property is successfully matched, the index of its corresponding ground truth property is added to this set. In subsequent comparisons, we only considered those ground truth properties not already matched, as indicated by their absence from the index set.

Stage 2: Evaluating values, units and conditions of the properties

To take into account the details of the properties in F_1 score, we needed a comprehensive and nuanced approach to compare entities' values, units, and conditions of the properties to evaluate the performance. For each of the entities, we calculated a matching score. The final score for a property was an average of these individual scores. We employed a threshold to determine what is considered a match (true positive) for a property. It is important to note that F_1 scores obtained in stage 2 are affected by the performance of the match mechanism explained in stage 1 as we compared the values, units and conditions of the properties that are matched considering their names.

- **Values:** We used an equality check, where a score of 1 is assigned for an exact match and 0 for a mismatch between "value" of the property in the ground truth and the predicted results.
- **Units:** Similar to values, units (such as K, min, etc.) are compared using an equality check.
- **Conditions:** Conditions are comprised of multiple entities: "type," "value," and "unit." The similarity between conditions in the prediction and the ground truth was evaluated by comparing these entities. The conditions entity is a list because properties can be measured or reported under multiple additional conditions. For example, the same property could be measured at different temperature values and different humidity values. We iterated through each condition in the predictions and identify the condition in the ground truth that had the highest match score without being previously matched. For each pair of conditions–one from the prediction and one from the ground truth–a match score was calculated based on the three entities: type, value, and unit. If an entity exactly matched, it scores 1, if not, it scores 0. However, we could use other methods such as similarity metric as we did to match the properties or sub-string comparison. The final match score for a condition pair is the average of these three scores, which means it can range from 0 (no match) to 1 (a perfect match). These highest match scores for all conditions in the prediction were then summed up to determine the total match score. Here, we aimed to ensure that each condition in the prediction was matched with its most similar counterpart in the ground truth. To obtain the *condition score*, the total match score was then normalized by dividing it by the larger of the two condition counts either in the predictions or the ground truth which we denote by *N*. Figure 5

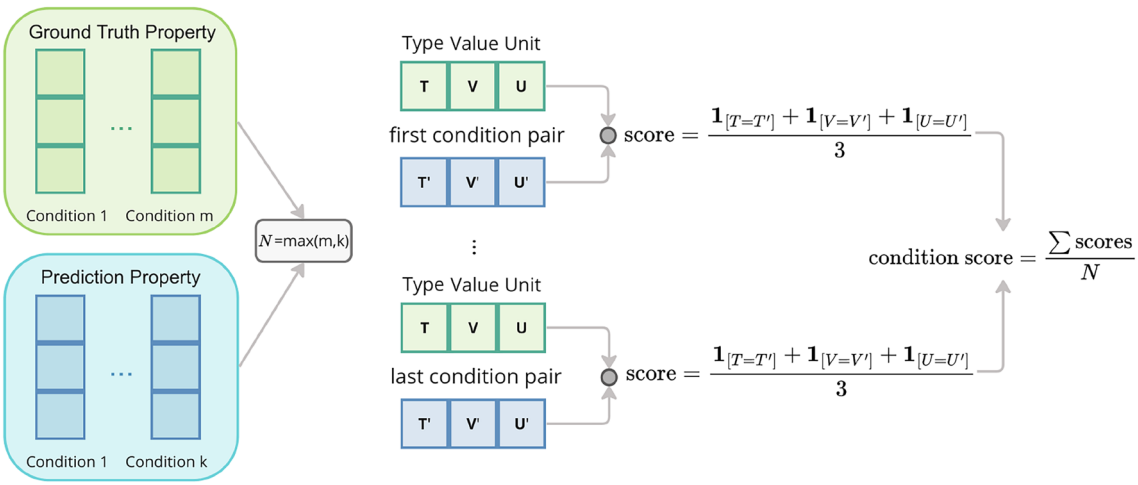

Fig. 5 Illustration of the process for calculating the condition score in a dataset. The method involves iterating through each condition in the predictions and matching it with the most similar condition in the ground truth. The match score for each pair is determined based on the comparison of three entities: type (T), value (V), and unit (U) The condition score is then computed by summing the highest match scores for all conditions in the prediction and normalizing this sum by the larger condition count in either the predictions or ground truth

illustrates this process. The maximum value observed for this dataset is 2 although the evaluation metric is valid for any value of N. Therefore, the condition score here takes values in $\{0, 1/6, 2/6, 3/6, 4/6, 5/6, 1\}$ as there are three entities in each condition. This normalization adjusts the final score of the conditions to fall within a range between 0 and 1.

Results and Discussion

Table 1 provides a breakdown of valid table predictions and number of samples obtained through different input types. Details of calculations are explained under handling missing samples in Sect. 2.4. Note that we obtained lists of valid JSONs for all tables when the image was used as an input. However, when OCR and structured format were used as an input, in some cases predictions were missing or the obtained JSONs were invalid. In all input cases, there were some missing samples.

Composition Information

Table 2 shows the accuracy scores of composition information. When the missing samples were not included, structured format with captions performed the best with an average accuracy score equal to 0.948. Image, OCR and structured format without captions have accuracy scores 0.917, 0.890 and 0.890, respectively. When the missing samples were included, image, structured format without captions, structured format with captions and OCR gave accuracy scores of 0.910, 0.832, 0.816 and 0.790, respectively.

We found that the predicted samples, when structured format with captions were used, had the highest average accuracy with a score of 0.948. Here, there is no penalty for not making the predictions (excluding missing samples). When a complete list is desired, it is necessary to penalize for missing some samples in the predictions or not giving any valid predictions. In this case, the image input performed the best with a score of 0.910. We observed that the strength of the image model lies in producing only valid tables and generating fewer invalid samples. This results highlights potential areas for improvement in other models by modifying the prompts.

Table 1 Fraction of invalid tables and fraction of samples that are missing

Category/input type	Image	OCR	Structured format	
			With captions	Without captions
Invalid tables	0.0	0.081	0.135	0.054
Missing samples	0.016	0.137	0.126	0.120

Table 2 Accuracy scores of composition information extraction using OCR, image, and structured format as an input with their 95% confidence intervals

Input type/Including missing samples	No	Yes
Image	0.917 ± 0.036	$\mathbf{0.910 \pm 0.037}$
OCR	0.890 ± 0.065	0.790 ± 0.107
Structured format (with captions)	$\mathbf{0.948 \pm 0.032}$	0.816 ± 0.113
Structured format (without captions)	0.890 ± 0.056	0.832 ± 0.089

The best performances are indicated in bold

Property Information

For the matching of property names between predicted samples and ground truth samples considering property names as explained in Sect. 2.4.2, manual inspection showed that using Levenshtein distance with a threshold as a similarity metric generally worked very well. Notable examples of successful matches through this method include "decomposition temperature" with "thermal decomposition temperature," "real relative permittivity at low field" with "real relative permittivity," and "dielectric permittivity" with "measured dielectric permittivity." There were few instances where this method failed to identify matches with equivalent meanings. An example was the mismatch between "nitrogen content" in the predictions and "element analysis nitrogen" in the ground truth, where the terms refer to the same property but were not recognized as a match due to the significant lexical differences.

Table 3 shows the precision, recall and F_1 scores of property name information extraction. Image input performed the best with image, structured format with captions, OCR and structured format without captions giving average F_1 scores of 0.863, 0.682, 0.666 and 0.576, respectively. We believe the superior performance of the image model may be due to its ability to incorporate both textual and visual cues from images, enhancing its understanding of the table's structure and providing a richer context. For example,

Table 3 F_1, precision and recall scores of property name information extraction using image, OCR, and structured format as an input with their 95% confidence intervals for all tables

Input type	Precision	Recall	F_1
Image	$\mathbf{0.905 \pm 0.074}$	$\mathbf{0.844 \pm 0.086}$	$\mathbf{0.863 \pm 0.078}$
OCR	0.740 ± 0.113	0.639 ± 0.122	0.666 ± 0.117
Structured format (with captions)	0.740 ± 0.131	0.662 ± 0.131	0.682 ± 0.129
Structured format (without captions)	0.627 ± 0.139	0.556 ± 0.135	0.576 ± 0.134

The best performances are indicated in bold

hierarchies within the tables are often lost when converted to text. To assess the models' ability to extract property names from tables, we categorized them as either simple or hard based on their layout. We analyzed 20 simple tables with straightforward layouts and 17 hard tables with more complex arrangements. As expected, the F_1 scores for the simple tables were higher, as detailed in Table 4. The gains in processing simple tables were similar for both the image and the structured format with captions. OCR, which does not maintain the structure, exhibited a significantly higher improvement on simple tables compared to hard ones. We observed minimal difference in performance for the structured format without captions. This likely stems from the frequent mention of property names in captions; omitting them can notably degrade performance in both of the cases. We also observed that the average precision values are higher than the recall values in all cases.

The inclusion of captions with the structured format increased the scores of both composition and property name stressing the importance of this inclusion in information extraction.

For property details such as value, unit and conditions (Stage 2 of Property evaluation), we determined the property matches between ground truth samples and the predicted samples. We used a threshold to determine which properties should count as a true positive considering its value, unit and conditions. This threshold approach allowed for some degree of variation in the predicted output, acknowledging that perfect matches are not always feasible. In Fig. 6, we reported F_1 scores demonstrating how well the details of the properties were extracted after the properties were matched with varying thresholds when all the samples were considered.

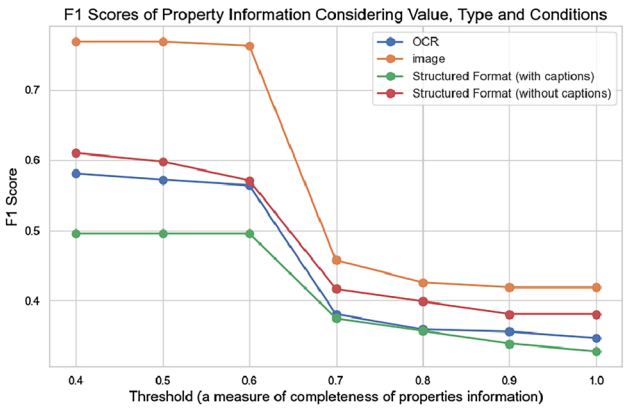

Fig. 6 F_1 scores of property information considering value, type, and conditions for input types image, OCR, structured format (with captions), and structured format (without captions) based on different thresholds

The value of the threshold determines the acceptable average of the correctness scores of the three fields in properties: value, type and conditions as explained in Sect. 2.4.2. The higher it is, the more strict the evaluation becomes; therefore, the scores are lower. Considering details of properties such as units and conditions is especially critical in scientific articles. There is a noticeable decrease after a threshold of 0.6 as after the threshold is 0.66, we expect at least two of the three detail fields to be correct which makes the evaluation much stricter than the lower thresholds. Only the conditions field can take a range of values as there can be multiple conditions with a varying number of correct sub fields, whereas value and unit fields are binary. This will

Table 4 F_1, precision, and recall scores of property name information extraction using image, OCR, and structured format as an input with their 95% confidence intervals

Simple Tables						
Input type	Precision	Recall	F_1	% ↑ in P	% ↑ in R	% ↑ in F_1
Image	0.932 ± 0.054	0.893 ± 0.078	0.905 ± 0.062	6.88%	13.61%	11.31%
OCR	0.815 ± 0.122	0.723 ± 0.148	0.745 ± 0.138	25.19%	33.89%	29.85%
Structured format						
(With captions)	0.752 ± 0.186	0.687 ± 0.193	0.700 ± 0.189	3.44%	11.52%	9.03%
(Without captions)	0.595 ± 0.202	0.572 ± 0.207	0.579 ± 0.205	−10.39 %	6.32%	1.40%
Hard Tables						
Input type	Precision	Recall	F_1			
Image	0.872 ± 0.157	0.786 ± 0.172	0.813 ± 0.162			
OCR	0.651 ± 0.208	0.540 ± 0.207	0.573 ± 0.204			
Structured format						
(With captions)	0.727 ± 0.176	0.616 ± 0.173	0.642 ± 0.171			
(Without captions)	0.664 ± 0.212	0.538 ± 0.190	0.571 ± 0.190			

The results for simple tables, including percent increases from hard to simple tables, are presented first followed by hard tables

cause smaller changes in the score. The reported value is the average of the three fields: value, unit and conditions.

Interestingly, the structured format without captions performed better than with captions as seen in Fig. 6. We believe this result arises because predictions of more samples were missing when captions are included and usually details such as values, units and conditions of the properties are reported inside the table, not in captions. This underscores the need to carefully consider different evaluation strategies and their results, as this example illustrates a trade-off between increasing the information details considered and maximizing F1 score.

Challenges of Information Extraction from Tables

This study has highlighted a number of important challenges in all input types. The challenges we addressed, which included some brought forth by Hira et al. [51]: extracting the same properties measured under different conditions and understanding the meaning of the rows or columns even if they are abbreviated or semantically similar to one another. Detailed analysis identified several additional challenges which we report below based on where they occur: Composition information, properties and both composition information and properties.

Composition Information

1. Differentiating between filler name and PST chemical name: Accurately identifying whether a chemical name refers to a filler material or a PST. This involves recognizing the context and classification of each chemical listed as shown in Fig. 7. This was also a challenge for human annotators as in this example they also made a mistake considering the PST as filler names. "UN" and "VS" are used as abbreviations for untreated and vinyl silane treatment but this can be only understood by reading the text of the article.
2. Handling extraneous information: Tables can contain additional information not relevant to the prompt, like processing methods. For example, processing methods "melt extrusion" and "SSSP" (solid-state shear pulverization) are mentioned in Fig. 8. At present, we are not requesting the model to extract processing information, and the model should ignore this text. However the model incorrectly attributed this extraneous information to PST. Gupta et al. also reported this challenge of filtering irrelevant information in composition extraction from tables [20]. This issue can be mitigated by crafting more detailed prompts that cover all details or, in this case, by a broader extraction goal including capturing processing features.
3. Implicit matrix names for the not specified ones: Identifying matrix names that are not explicitly mentioned but need to be inferred. (For example, matrix name "tritherm" is only mentioned in the unfilled sample in Fig. 9.) This complexity involves understanding the context.

Properties

1. Differentiating between property name and its conditions: Distinguishing property names from the conditions under which they are measured or reported. For example, in Fig. 10, property name is reported as "dc characteristic breakdown strength @ 25°C" instead of separating the temperature as a condition. Providing models with a predefined list of potential properties can enhance their accuracy in identifying property names.
2. Different ways to refer to a property: Recognizing that very different terms can refer to the same property, both "loss tangent" and "tan delta" can be used as a property name for "tan δ" in Fig. 11. (This loose nomenclature issue also poses an evaluation challenge.)
3. Missing properties in the parentheses: Extracting properties that are listed in parentheses within another property column, rather than in a separate column (as in Fig. 10, where the Weibull parameter is included parenthetically in a column for the breakdown strength value).
4. Ambiguity of conditions: In this example shown in Fig. 7, it is unclear without context whether the reported temperature is the condition under which the property measurement is conducted or if it is an environmental condition to which the samples are exposed. Analysis of text paragraphs associated with a table together with the table may lead to reduced ambiguity.

Composition Information and Properties

1. Complex/non-traditional table structures: tables with irregular cell spans or merged cells that do not follow a typical row-column format can be challenging to the models. For example, in Fig. 12, the frequency is reported as a new column where the other columns are properties. It is also not very clear by just looking at the table which property is associated with the reported frequency. Upon careful inspection, we realized that both humans and LLMs labeled the frequency as a property name incorrectly. In Fig. 13, some of the elements in the table spans two rows. It is a complex task to associate the one element with multiple samples that are presented. Moreover, in Fig. 14, information about a single sample is spread across two rows, where each pair of rows reports properties under different temperature conditions.

2. Long sample list: tables with many samples reported (more than 5) are more likely to miss some samples in the output.
3. Unfilled samples can be missed by the model when table text is poorly constructed or overly abbreviated for space: Fig. 9 can be given as an example.
4. Understanding numerical values that are reported unconventionally: when unconventional formats are used for numerical values, such as scientific notation or mixed formats. For example, property value 1.9 x 10'8 in Fig. 11 is predicted to have a value of 1.9 when expected to be 1.9e8 and composition value 4–1/2 in Fig. 10 is predicted to be 4–1/2 when 4.5 is correct.

Advantages of Using LLMs

1. Understanding of property names: LLMs can comprehend the meaning of property names in tables, even when they are presented as abbreviations. This proficiency is evident in the interpretation of properties like "dielectric constant" and "dielectric loss" as demonstrated in Figs. 15 and 16.
2. Recognition of units of the properties: LLMs can recognize dimensionless nature without explicit mention in the table and correctly find the units mentioned in the table.
3. Expertise in complex properties: In cases involving complex properties that might fall outside the expertise of human curators, LLMs are often more reliable. For example, they successfully interpret tables with unusual syntax or specialized terms that may be challenging for human experts. An instance of this can be seen in Fig. 17, which includes properties with complex descriptions like impulse strength voltage. Human annotators mistakenly categorized this property as duration of the impulse strength.
4. Can be used as a validation: When the LLM result disagrees with the human curator, it might be more correct. For example, in Fig. 18, silver NP content which is reported as approximate NP content is predicted correctly as the filler composition instead of the composition value which is included in the sample description in the first column. Identifying weight percentages that pertain to the composition of the sample versus other weight percentages can be complex when tables include various types of weight data. LLMs can help us catch these kinds of mistakes (Fig. 19).

Advantages of Our Approach

Focusing on the text only, without considering figures and tables, it is possible to capture the subset of all samples that have the best performance, or the worst performance. The "middle of the pack" samples are rarely called out explicitly in the written text. By focusing on the tables, we were able to extract a wider selection of samples for more comprehensive data extraction. Incorporating numerical values, such as property values, lays the groundwork for future quantitative analysis.

Furthermore, it is important to note that extracting sample information from an experimental paper is a persistent challenge. Our flexible approach can be applied in sample extraction across various domains. This adaptability is achievable by modifying the template defined in the prompt and incorporating a few examples. It does not require a fully supervised dataset. While each domain might present its unique challenges, the general approach remains applicable throughout various realms within materials science.

The tables in our study encompass a diverse range of properties. This diversity poses challenges for evaluation. To navigate this complexity, we implemented an evaluation approach which first matches the property names in the ground truth and the predictions, and then considers the details of the properties to count them as a correct match with varying thresholds. This approach provides a nuanced assessment of performance.

Limitations and Future Opportunities

While in this work we focused on few-shot prompting, we believe designing better prompts and using chain-of-thought prompting may further improve performance. Future work could consider extending these approaches to extract sample information from figures. Future work could also explore including process details that could better guide materials design. However, a notable limitation in our current approach is the separate evaluation of each table in an article. A more integrated method that merges information across all tables could offer a holistic view of each sample's properties, leading to a more comprehensive understanding. Additionally, our current methodology does not include the extraction of variations in numerical property values. Moreover, we assume a direct match in the ordering of samples, implying that each sample's position in the model output corresponds to the same position in the human-annotated dataset, an assumption that could be avoided in future work. Due to the highly detailed comparisons of ground truth and model prediction, a relatively small number of tables were examined. Armed with the methods and findings in this work, we believe we will be able to deploy the extraction and analysis on a larger set of tables.

We also acknowledge the challenges faced in property matching, particularly highlighted in cases such as not being

able to match "nitrogen content" and "element analysis nitrogen." This underscores the need for a more sophisticated evaluation approach, perhaps through the exploration of alternative similarity scores better suited for this nuanced task. The exploration of different similarity metrics could significantly enhance the precision of our matching algorithms, reducing the margin of error and paving the way for more accurate data extraction. By addressing these challenges and exploring these new directions, we aim to push the boundaries of what is possible in information extraction from scientific tables.

Conclusion

Our work developed a rigorous method to compare different methodologies for materials science data extraction from tables using GPT-4 offering insights into the effectiveness and applicability of various techniques. We introduced an automated evaluation technique tailored to assess the accuracy and efficiency of these extraction methods, contributing to a nuanced understanding of their performance. We also compiled, annotated and analyzed a dataset of tables in the polymer composite domain, providing a resource for further research and application in this domain. Our results indicate that using GPT-4-Vision for table extraction with appropriate prompting results in the best performance compared to structured and unstructured table input methods. Through prompt design, we captured essential sample composition and property details such as values, units, and conditions. This study also highlighted a number of detailed challenges that occur for tabular data extraction from typical materials science papers. These results underscore the complexities involved in information extraction and also pave the way for future research to address these issues.

Supplementary Information

Challenges of Different Inputs

- **GPT-4-Vision on table image:** We spent 80.752 tokens of which 51.453 are context tokens and 29.299 are generated tokens with a total of 1.39$. Out of 37 tables, all of them gave valid list of JSON outputs. When missing samples are excluded, the number of samples considered went down to 179.
- **GPT-4 on unstructured OCR extraction from table image:** We spent 78.728 tokens of which 46.246 are context tokens and 32.482 are generated tokens with a total of 1.44$. Out of 37 tables, 34 of them had valid list of JSON outputs as predictions. When three of these tables and other missing samples are excluded the number of samples considered went down to 157.
- **GPT-4 on structured table output from PDF files:** We spent 100.523 tokens of which 59.585 are context tokens and 40.938 are generated tokens with a total of 1.82$. We found that when considering the structured format of JSON outputs, 32 tables yielded valid results with captions included, and 35 were valid with captions excluded. Initially, sample size was 182. Upon excluding non-valid JSON files and missing samples resulted in final sample counts of 159 (with captions) and 160 (without captions).

Appendix A: Prompt

```
"Identify and document detailed information about each nano and micro-composite
sample listed in a provided table, using a JSON format. Detailed Instructions:
1. Sample Identification:
o Review each sample in the table.
2. JSON Template Completion:
o For each sample, fill out the following JSON template:

{
    \"sample_id\": [Sample ID Number],
    \"matrix_name\": [Matrix Name],
    \"filler_name\": [Filler Name],
    \"filler description\": [Filler Description],
    \"composition\": {
        \"amount\": [Amount of Filler],
        \"type\": [Type of Composition]
    },
    \"particle_surface_treatment_name\": [Particle Surface Treatment Name],
    \"properties\": {
        [Property Name 1]: {
            \"value\": [Value],
            \"unit\": [Unit],
            \"conditions\": [
                {\"type\": [Condition Type], \"value\": [Condition Value],
                \"unit\": [Condition Unit]}
            ]
        },
        [Property Name 2]: {
            \"value\": [Value],
            \"unit\": [Unit],
            \"conditions\": [
                {\"type\": [Condition Type], \"value\": [Condition Value],
                \"unit\": [Condition Unit]}
            ]
        }
        // Add more properties as needed
    }
}

Data Entry Guidelines:
o Matrix Name: Enter the matrix's material name. Exclude any descriptors related
to size or treatment.
o Filler Name: Enter only the chemical name of the filler. Exclude descriptors like
nano/micro, treated/non-treated, and size.
o Filler Description: Indicate whether the filler is nano or micro. If not specified,
use "not specified".
o Composition: Include the filler's amount (eg: 3%) and type (vol or wt or not
specified). If no filler is present, enter "none" for filler name and "0.0%" for
composition. If there are reported in both of the types, just write the value and
type of one of them.
o Particle Surface Treatment Name: enter chemical treatment name if known, "treated"
if particles are treated but name is unknown, "untreated" if no treatment is applied,
"not specified" if treatment status is unknown.
o Properties: Document all properties listed for each sample. Use full names for
properties instead of abbreviations. Include value, unit, and any conditions
specified. Exclude the conditions from the property name. Ignore the deviations
if reported.
o If any information is missing in the table, use "not specified" in the JSON.
Please extract all relevant information from the table and generate a complete
JSON output, encompassing each nano and micro composite sample detailed in the
provided table. Do not put any comments in the JSON output.
Here is an example:
```

```
[
    {
        "sample_id": 1,
        "matrix_name": "PP",
        "filler_name": "none",
        "filler_description": "nano",
        "composition": {"amount": "0.0%", "type": "not specified"},
        "particle_surface_treatment_name": "not specified",
        "properties": {
            "Young's modulus": {"value": 910, "unit": "MPa"},
            "Yield strength": {"value": 28, "unit": "MPa"},
            "Elongation at break": {"value": 810, "unit": "%"},
            "Absorbed energy per thickness": {"value": 3.09, "unit": "J/cm"},
            "Crystallization temperature": {
                "value": 390, "unit": "K",
                "conditions": [{"type": "cooling speed", "value": "-10",
                "unit": "K/min"}]
            },
            "Half Life of Crystallization": {
                "value": "120", "unit": "min",
                "conditions": [{"type": "temperature", "value": -413,
                "unit": "K"}]
            }
        }
    },
    {
        "sample_id": 2,
        "matrix_name": "PP",
        "filler_name": "graphite",
        "filler_description": "not specified",
        "composition": {"amount": "2%", "type": "wt"},
        "particle_surface_treatment_name": "not specified",
        "properties": {
            "Young's modulus": {"value": 1300, "unit": "MPa"},
            "Yield strength": {"value": "N/A", "unit": "MPa"},
            "Elongation at break": {"value": 8, "unit": "%"},
                   "Absorbed energy per thickness": {"value": 0.84,
                   "unit": "J/cm"},
                   "Crystallization temperature": {
                       "value": 402, "unit": "K",
                       "conditions": [{"type": "cooling speed", "value": "-10",
                       "unit": "K/min"}]
                   },
                   "Half Life of Crystallization": {
                       "value": 9.5, "unit": "min",
                       "conditions": [{"type": "temperature", "value": -413,
                       "unit": "K"}]
                   }
               }
           }
       ]


       Insert [TABLE]
       "
```

Appendix B: Examples of Tables

See Figs. 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,17, 18, 19.

Table A Saturated moisture content (with standard deviation) after months of exposure to humid environments (9 months for 100% rh conditioned samples, and 2 months for 75% rh conditioned samples) [14]

	Moisture content wt%			
	25 °C 100% rh	50 °C 100% rh	80 °C 100% rh	50 °C 75% rh
XLPE	0.01 ± 0.01	0.02 ± 0.01	0.04 ± 0.01	–
5 wt% VS nano	0.22 ± 0.01	0.35 ± 0.03	1.05 ± 0.04	0.05 ± 0.01
5 wt% UN nano	0.28 ± 0.01	0.38 ± 0.05	1.19 ± 0.04	0.07 ± 0.01

Fig. 7 Example of two challenges: **a** differentiating between filler name and PST chemical name. "UN" and "VS" (red boxes) are used as abbreviations for untreated and vinyl silane treatment. However, they are labeled as filler names by both humans and LLMs across all input types. **b** ambiguity of conditions. Without context, it is unclear whether the reported temperature and humidity (blue box) are the conditions under which the property measurement is conducted or if they are environmental conditions to which the samples are exposed. Simulated table, after [41]

Table A AC breakdown scale parameters, shape parameters, and % decrease

	η	β	% decrease
Tritherm at 300°C	122	4	56
5 wt% untr silica at 300°C	181	4	9
10 wt% untr silica at 300°C	220	3	2
5 wt% tr silica at 300°C	158	5	29
10 wt% tr silica at 300°C	251	4	–

Fig. 9 Example of two challenges: **a** unfilled samples not included. The sample in the first row (highlighted in red) which does not contain any fillers is omitted in the predictions. **b** matrix names are not specified, but implied to be the same as the first row, "tritherm," for the unfilled sample. While humans knew that other filled samples have the same matrix name, LLMs across all input types failed to label it as a matrix name. Simulated table, after [36]

Table 1. Thermal and Mechanical Property Enhancement in PP−Graphite Composites[a]

	tensile properties			impact strength	crystallization behavior	
samples	Young's modulus, E (MPa)	yield strength, σ_y (MPa)	elongation at break, ϵ_B (%)	absorbed energy per thickness, W (J/cm)	crystallization temp, $T_{c,onset}$, at −10 K/min (K)	isothermal crystallization half-time, $\tau_{1/2}$, at 413 K (min)
neat PP	910 ± 30	28 ± 2	810 ± 30	3.09 ± 0.49	390	>120
PP/2.8 wt % graphite melt extrusion	1300 ± 50	N/A	8 ± 1	0.84 ± 0.20	402	9.5
PP/2.5 wt % graphite SSSP	1870 ± 170	43 ± 3	560 ± 60	1.21 ± 0.15	411	3.6

[a] The values following ± are errors of one standard deviation. The complete data set is included in Table S1 of the Supporting Information.

Fig. 8 Example of the challenge of having extra information. Processing methods "melt extrusion" and "SSSP" which stands for solid-state shear pulverization are mentioned in the table which are not relevant to the prompt. When image is used as an input, "melt extrusion" is incorrectly labeled as particle surface treatment. Reprinted with permission from [47]. © 2008, American Chemical Society

Table 2. Breakdown strength for unfilled and nanoparticle-filled resins showing that the addition of nanoparticles increases the dielectric breakdown strength. The Weibull shape parameters are given in parentheses.

Material [Ref]	dc Characteristic Breakdown Strength @ 25°C in kV/mm (β)	dc Characteristic Breakdown Strength @ 80°C in kV/mm (β)
Unfilled XLPE [7]	270 (2.5)	79 (3.8)
5 wt% untreated 12nm nanosilica-filled XLPE [7]	315 (2.0)	83 (3.1)
5 wt% vinyl silane-treated 12nm nanosilica-filled XLPE [7]	446 (1.7)	220 (2.9)
Unfilled ether-bisphenol epoxy resin [24]	332 (10.56)	------
10 wt% untreated 22 nm nanotitania-filled epoxy resin [24]	391 (10.39)	------
Unfilled ether-bisphenol epoxy resin [25]	347	------
4-1/2 wt% nanoclay (MMT)-filled epoxy resin [25]	531	------

Fig. 10 Example of three challenges: **a** differentiating between property name and its conditions. The property "dc characteristic breakdown strength" is predicted, where "at 25°C" should be recognized as a condition, not part of the property's name. **b** missing properties in the parentheses. The Weibull shape parameters, ideally requiring a distinct column, are instead embedded within the "characteristic breakdown strength" column. This leads to inconsistencies, such as these parameters being mistakenly categorized as conditions or omitted in predictions. **c** understanding numerical values that are reported unconventionally. Composition value "4–1/2" is inaccurately predicted as "4–1/2" instead of the correct notation "4.5" across all input types. Reprinted with permission from reference [35]. © 2008, IEEE

TABLE II
Dynamic Mechanical Properties of EVA and Its Nanocomposites

Sample	T_g (°C)	E' (Pa) at T_g	E' (Pa) at 30°C	tan δ at T_g	tan δ at 30°C
Pure EVA	−27	05×10^7	1.5×10^6	0.95	0.17
EVA + 4 wt % 12Me-MMT	−30	1.9×10^8	04×10^6	0.68	0.16
EVA + 6 wt % 12Me-MMT	−32	06×10^8	07×10^6	0.55	0.17

Fig. 11 Example of the challenge of understanding numerical values that are reported unconventionally. Property value "1.9 x 10'8" is inaccurately predicted as "1.9" instead of the correct notation "1.9e8" when OCR and structured format are used as an input. Reprinted with permission from reference [33]. © 2003, Wiley

Table 1. Lichtenecker-Rother predictions of composite material dielectric permittivity (ε') and measured values at 60 Hz at 25 °C [17-19], at 30 °C [20]

Material	f(Hz)	ε'(L-R)	Measured ε'
Unfilled ether-bisphenol epoxy resin	1k	----	10.0
Untreated 23 nm nanotitania	1k	----	99
10 wt% (3.0 vol%) untreated 22 nm nanotitania-filled epoxy resin	1k	10.1	13.8
Unfilled polyimide (BTDA-ODA)	100k	----	3.5
Untreated 12 nm nanoalumina	100k	----	9.8
5 vol% untreated 12 nm nanoalumina-filled polyimide	100k	3.7	6.0
Unfilled crosslinked polyethylene (XLPE)	100k	----	2.4
Untreated 12 nm nanosilica	100k	----	4.5
5 wt% (1.9 vol%) untreated 12 nm nanosilica-filled XLPE	100k	2.4	2.0
Unfilled low-density polyethylene (LDPE)	10k	----	2.3
Untreated 30 nm ZnO nanoparticles	10k	----	8
10 wt% (1.7 vol%) untreated 30 nm ZnO nanoparticle-filled LDPE	10k	2.35	2.52

Fig. 12 Example of the challenge of complex/non-traditional table structures. Frequency is reported in a separate column, distinct from other property columns, leading to ambiguity regarding its association with specific properties. Despite careful review, both human evaluators and language models erroneously identified frequency as a property name. Reprinted with permission from reference [35]. © 2008, IEEE

Table 1
Summary of T_g, and quality of dispersion for two samples of each type of composite.

Type of sample	Sample	T_g (°C)	$\bar{A}$ mean distance between agglomerates (μm)
2 wt% modified TiO_2 in PMMA	1	119.2 ± 0.47	4.13 ± 0.25
	2	120.7 ± 1.58	3.78 ± 0.18
PMMA	1	116.4 ± 1.07	N/A
2 wt% TiO_2 in PMMA	1	113.8 ± 0.55	3.98 ± 0.03
	2	115.0 ± 0.65	3.84 ± 0.33
3 wt% TiO_2 in PMMA	1	110.5 ± 0.78	4.16 ± 0.14
	2	116.6 ± 0.69	4.60 ± 0.29

Fig. 13 Example of the challenge of complex/non-traditional table structures. The first and the forth row of the type of the sample column spans two rows as there are two types of each sample. This can be understood by looking at the other two columns. Reprinted with permission from reference [48]. © 2009, Elsevier

Table A DC breakdown scale parameters, shape parameters, and % decrease

	η	β	% decrease
Tritherm at 200°C	257	6	21
5 wt% untr silica at 200°C	380	3	17
10 wt% untr silica at 200°C	290	2	14
Tritherm at 300°C	120	2	63
5 wt% untr silica at 300°C	275	4	40
10 wt% untr silica at 300°C	282	14	16

Fig. 14 Example of the challenge of different rows need to be merged. Information pertaining to the same samples is spread across multiple rows (the control sample in rows 1 and 4 (red boxes), the 5wt% sample in rows 2 and 5 (blue boxes), the 10 wt% sample in rows 3 and 6 (green boxes)), where each pair of rows reports properties under varying conditions. While the table contains data for three unique samples, structured format and image-based input method predicts six samples. Simulated table, after [36]

Table 6. Impulse test breakdown fields for the XLPE and 12-1/2% nanocomposite materials. The Weibull shape parameters are given in parentheses.

Material	1.2x50 μs Impulse strength @ 25 °C in kV/mm (β)
Unfilled XLPE	254 (3.6)
12½ wt% untreated 12nm nanosilica-filled XLPE	311 (4.9)
12½ wt% vinyl silane-treated 12nm nanosilica-filled XLPE	332 (5.2)

Fig. 17 A table featuring "impulse strength voltage," mistakenly identified by human curators as "impulse duration" due to the reporting in microseconds. Reprinted with permission from reference [35]. © 2008, IEEE

Fig. 15 This table lists the surface properties, where "ΘH_2O" represents the water contact angle and "γs" denotes the surface tension components. Understanding the abbreviations requires domain-specific knowledge. Reproduced with permission from reference [37]. © 2006, Wiley

Table 3. Surface Properties of PI/OFG Nanocomposites

OFG in PI (Feed wt %)	Θ (° ± σ)				
	H_2O	Glycerol	γ_s (mN/m)[a]	γ_s^d (mN/m)[a]	γ_s^p (mN/m)[a]
0 %	60.5 ± 1.7	71.1 ± 1.2	50.7	3.06	47.7
3 %	67.3 ± 1.0	70.4 ± 1.0	45.8	3.96	35.7
7 %	69.5 ± 0.8	72.3 ± 0.9	44.0	5.75	33.6
10 %	71.9 ± 1.2	74.0 ± 1.1	43.4	6.43	29.9
15 %	73.1 ± 0.7	74.8 ± 0.6	41.1	6.48	29.0

[a] Calculated with Wu's harmonic mean method.[40]

Table 1. Concentration of Each Component in the BT-Ag/PVDF Composites and Comparison of Dielectric Properties

sample	BT-Ag (wt %)	BT-Ag (vol %)	BT (vol %)	Ag[a] (vol %)	D_k/tan δ 1 kHz	D_k/tan δ 100 kHz
PVDF	0	0	0	0	10.1/0.03	9.45/0.05
BT-Ag/PVDF-1	20	7.6	6.5	1.1	13.6/0.03	12.5/0.05
BT-Ag/PVDF-2	40	18.0	15.4	2.6	20.6/0.03	18.8/0.05
BT-Ag/PVDF-3	60	33.0	28.2	4.8	54.0/0.06	46.0/0.07
BT-Ag/PVDF-4	70	43.4	37.1	6.3	94.3/0.06	81.0/0.06
BT-Ag/PVDF-5	80	56.8	48.6	8.2	160.0/0.11	127.3/0.06

[a]The calculated content of Ag in the BT-Ag hybrid particles was 28.6 wt %.

Fig. 16 This table lists the electrical properties of materials, where D_k represents the dielectric constant and tan δ denotes the loss tangent. Understanding the abbreviations requires domain-specific knowledge. Reprinted with permission from reference [46]. © 2014, American Chemical Society

Table 2. NP content of nanocomposites calculated by TGA analysis.

Cured sample	Experimental char content	Approximate NP content
	wt.-%	wt.-%
CE	0.7	0
CE + 3 wt.-% $AgSbF_6$	1.3	0.6
CE + 5 wt.-% $AgSbF_6$	1.8	1.1
CE + 7 wt.-% $AgSbF_6$	2.3	1.6
CE + 10 wt.-% $AgSbF_6$	3.2	2.5
CE + 15 wt.-% $AgSbF_6$	4.8	4.1
CE + 20 wt.-% $AgSbF_6$	5.8	5.1

Fig. 18 Example of the challenge of differentiating between other content weight percentages and composition. Silver NP content which is reported as approximate NP content is predicted instead of the filler composition which is included in the sample description in the first column when image and structured format with captions are used as an input. Reproduced with permission from reference [39]. © 2010, Wiley

Table 5. Breakdown strength for unfilled and filled crosslinked polyethylene showing that the addition of nanoparticles incresases the dielectric breakdown strength. The Weibull shape parameters are given in parentheses.

Material	dc Characteristic Breakdown Strength @ 25°C in kV/mm (β)	ac (60 Hz) Characteristic Breakdown Strength @ 25°C in peak kV/mm (β)
Unfilled XLPE	184 (5.1)	178 (4.5)
12½ wt% 6μm microsilica-filled XLPE	162 (5.9)	139 (5.4)
12½ wt% untreated 12nm nanosilica-filled XLPE	191 (4.8)	186 (5.0)
12½ wt% vinyl silane-treated 12nm nanosilica-filled XLPE	239 (5.2)	193 (5.8)

Fig. 19 Example of the challenge of differentiating between "untreated" and "not specified" for particle surface treatment. The microsilica-filled sample is predicted as "untreated" across all input types although the status is unknown. Reprinted with permission from reference [35]. © 2008, IEEE

Acknowledgements We thank NSF DGE-2022040 and NSF CSSI-1835677 for funding support. We also thank Dr. Jamie McCusker and Dr. Shruti Badhwar for their insightful discussions.

Code and Data Availability The code and data are available on GitHub at the following address: https://github.com/defnecirci/MatSciTableExtract.

Declarations

Conflict of Interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. Horawalavithana S, Ayton E, Sharma S, Howland S, Subramanian M, Vasquez S, Cosbey R, Glenski M, Volkova S (2022) Foundation models of scientific knowledge for chemistry: Opportunities, challenges and lessons learned. In: Proceedings of BigScience Episode# 5–Workshop on Challenges & Perspectives in Creating Large Language Models, pp. 160–172
2. Piekm YH (2022) Ml-based procedural information extraction and knowledge management system for materials science literature. In: Proceedings of the 2nd conference of the Asia-pacific chapter of the association for computational linguistics and the 12th international joint conference on natural language processing: system demonstrations, pp. 57–62
3. Szymanski NJ, Rendy B, Fei Y, Kumar RE, He T, Milsted D, McDermott MJ, Gallant M, Cubuk ED, Merchant A, et al (2023) An autonomous laboratory for the accelerated synthesis of novel materials. Nature, pp. 1–6
4. McCusker JP, Keshan N, Rashid S, Deagen M, Brinson C, McGuinness DL Nanomine: A knowledge graph for nanocomposite materials science. In: International semantic web conference, pp. 144–159 (2020). Springer
5. Olivetti EA, Cole JM, Kim E, Kononova O, Ceder G, Han TY-J, Hiszpanski AM (2020) Data-driven materials research enabled by natural language processing and information extraction. Appl Phys Rev 7(4):2–16
6. Dunn A, Dagdelen J, Walker N, Lee S, Rosen AS, Ceder G, Persson K, Jain A (2022) Structured information extraction from complex scientific text with fine-tuned large language models. arXiv preprint arXiv:2212.05238)
7. Foppiano L, Castro PB, Ortiz Suarez P, Terashima K, Takano Y, Ishii M (2023) Automatic extraction of materials and properties from superconductors scientific literature. Sci Technol Adv Mater Methods 3(1):2153633
8. Shetty P, Ramprasad R (2021) Automated knowledge extraction from polymer literature using natural language processing. Iscience 24(1):1–9
9. Xie T, Wa Y, Huang W, Zhou Y, Liu Y, Linghu Q, Wang S, Kit C, Grazian C, Hoex B (2023) Large language models as master key: Unlocking the secrets of materials science with gpt. arXiv preprint arXiv:2304.02213
10. Gilligan LP, Cobelli M, Taufour V, Sanvito S (2023) A rule-free workflow for the automated generation of databases from scientific literature. arXiv preprint arXiv:2301.11689
11. Cheung JJ, Zhuang Y, Li Y, Shetty P, Zhao W, Grampurohit S, Ramprasad R, Zhang C (2023) Polyie: A dataset of information extraction from polymer material scientific literature. arXiv preprint arXiv:2311.07715
12. Choi J, Lee B (2023) Accelerated materials language processing enabled by gpt. arXiv preprint arXiv:2308.09354
13. Polak MP, Modi S, Latosinska A, Zhang J, Wang C-W, Wang S, Hazra AD, Morgan D (2023) Flexible, model-agnostic method for materials data extraction from text using general purpose language models. arXiv preprint arXiv:2302.04914
14. Kononova O, Huo H, He T, Rong Z, Botari T, Sun W, Tshitoyan V, Ceder G (2019) Text-mined dataset of inorganic materials synthesis recipes. Sci. Data 6(1):203
15. Wang Z, Kononova O, Cruse K, He T, Huo H, Fei Y, Zeng Y, Sun Y, Cai Z, Sun W et al (2022) Dataset of solution-based inorganic materials synthesis procedures extracted from the scientific literature. Sci. Data 9(1):231
16. Shetty P, Rajan AC, Kuenneth C, Gupta S, Panchumarti LP, Holm L, Zhang C, Ramprasad R (2023) A general-purpose material property data extraction pipeline from large polymer corpora using natural language processing. NPJ Comput Mater 9(1):52

17. Venugopal V, Sahoo S, Zaki M, Agarwal M, Gosvami NN, Krishnan NA (2021) Looking through glass: knowledge discovery from materials science literature using natural language processing. Patterns 2(7):1–10
18. Circi D, Khalighinejad G, Badhwar S, Dhingra B, Brinson L (2023) Retrieval of synthesis parameters of polymer nanocomposites using llms. In: AI for accelerated materials design-NeurIPS 2023 workshop
19. Khalighinejad G, Circi D, Brinson LC, Dhingra B (2024) Extracting polymer nanocomposite samples from full-length documents
20. Gupta T, Zaki M, Krishnan N, et al (2022) Discomat: distantly supervised composition extraction from tables in materials science articles. arXiv preprint arXiv:2207.01079
21. Sayeed HM, Smallwood W, Baird SG, Sparks TD (2023) Nlp meets materials science: quantifying the presentation of materials data in scientific literature. Matter 7(3):723–727https://doi.org/10.26434/chemrxiv-2023-wd5cr-v3
22. Zhang Z, Tang H, Xu Z (2023) Fatigue database of complex metallic alloys. Sci Data 10(1):447
23. Jensen Z, Kim E, Kwon S, Gani TZ, Román-Leshkov Y, Moliner M, Corma A, Olivetti E (2019) A machine learning approach to zeolite synthesis enabled by automatic literature data extraction. ACS Central Sci 5(5):892–899
24. Zaki M, Krishnan N, et al (2023) Mascqa: A question answering dataset for investigating materials science knowledge of large language models. arXiv preprint arXiv:2308.09115
25. Oka H, Yoshizawa A, Shindo H, Matsumoto Y, Ishii M (2021) Machine extraction of polymer data from tables using xml versions of scientific articles. Sci Technol Adv Mater Methods 1(1):12–23
26. Sui Y, Zhou M, Zhou M, Han S, Zhang D (2023) Evaluating and enhancing structural understanding capabilities of large language models on tables via input designs. arXiv preprint arXiv:2305.13062
27. Zikry A (2008) Dielectric behavior of silica/polyacrylamide nanocomposites. Int J Polym Mater 57(4):383–395
28. Prabhune P, Comlek Y, Shandilya A, Sundararaman R, Schadler LS, Brinson LC, Chen W (2023) Design of polymer nanodielectrics for capacitive energy storage. Nanomaterials 13(17):2394
29. Darwish MS, Mostafa MH, Al-Harbi LM (2022) Polymeric nanocomposites for environmental and industrial applications. Int J Mol Sci 23(3):1023
30. Brinson LC, Deagen M, Chen W, McCusker J, McGuinness DL, Schadler LS, Palmeri M, Ghumman U, Lin A, Hu B (2020) Viewpoint: polymer nanocomposite data: curation, frameworks, access, and potential for discovery and design. ACS Macro Lett 9:1086–1094. https://doi.org/10.1021/acsmacrolett.0c00264
31. Singha S, Thomas MJ (2008) Dielectric properties of epoxy nanocomposites. IEEE Trans Dielectr Electr Insul 15(1):12–23
32. Singha S, Thomas MJ (2009) Influence of filler loading on dielectric properties of epoxy-zno nanocomposites. IEEE Trans Dielectr Electr Insul 16(2):531–542
33. Pramanik M, Srivastava SK, Samantaray BK, Bhowmick AK (2003) Rubber-clay nanocomposite by solution blending. J Appl Polym Sci 87(14):2216–2220
34. Nelson J, Hu Y (2005) Nanocomposite dielectrics-properties and implications. J Phys D Appl Phys 38(2):213
35. Smith R, Liang C, Landry M, Nelson J, Schadler L (2008) The mechanisms leading to the useful electrical properties of polymer nanodielectrics. IEEE Trans Dielectr Electr Insul 15(1):187–196
36. Travelpiece A, Nelson J, Schadler L, Schweickart D (2009) Dielectric integrity of silica-pai nanocomposites at elevated temperature. In: 2009 IEEE conference on electrical insulation and dielectric phenomena, pp. 535–538 . IEEE
37. Ye Y-S, Chen W-Y, Wang Y-Z (2006) Synthesis and properties of low-dielectric-constant polyimides with introduced reactive fluorine polyhedral oligomeric silsesquioxanes. J Polym Sci, Part A: Polym Chem 44(18):5391–5402
38. Holt AP, Griffin PJ, Bocharova V, Agapov AL, Imel AE, Dadmun MD, Sangoro JR, Sokolov AP (2014) Dynamics at the polymer/nanoparticle interface in poly (2-vinylpyridine)/silica nanocomposites. Macromolecules 47(5):1837–1843
39. Vescovo L, Sangermano M, Scarazzini R, Kortaberria G, Mondragon I (2010) In-situ-synthetized silver/epoxy nanocomposites: Electrical characterization by means of dielectric spectroscopy. Macromol Chem Phys 211(17):1933–1939
40. Gao L, He J, Hu J, Li Y (2014) Large enhancement in polarization response and energy storage properties of poly (vinylidene fluoride) by improving the interface effect in nanocomposites. J Phys Chem C 118(2):831–838
41. Hui L, Schadler LS, Nelson JK (2013) The influence of moisture on the electrical properties of crosslinked polyethylene/silica nanocomposites. IEEE Trans Dielectr Electr Insul 20(2):641–653
42. Virtanen S, Krentz TM, Nelson JK, Schadler LS, Bell M, Benicewicz B, Hillborg H, Zhao S (2014) Dielectric breakdown strength of epoxy bimodal-polymer-brush-grafted core functionalized silica nanocomposites. IEEE Trans Dielectr Electr Insul 21(2):563–570
43. Wang Z, Nelson JK, Miao J, Linhardt RJ, Schadler LS, Hillborg H, Zhao S (2012) Effect of high aspect ratio filler on dielectric properties of polymer composites: a study on barium titanate fibers and graphene platelets. IEEE Trans Dielectr Electr Insul 19(3):960–967
44. Roy M, Nelson J, MacCrone R, Schadler LS, Reed C, Keefe R (2005) Polymer nanocomposite dielectrics-the role of the interface. IEEE Trans Dielectr Electr Insul 12(4):629–643
45. Roy M, Nelson JK, MacCrone R, Schadler L (2007) Candidate mechanisms controlling the electrical characteristics of silica/xlpe nanodielectrics. J Mater Sci 42:3789–3799
46. Luo S, Yu S, Sun R, Wong C-P (2014) Nano ag-deposited batio3 hybrid particles as fillers for polymeric dielectric composites: toward high dielectric constant and suppressed loss. ACS Appl Mater Interf 6(1):176–182
47. Wakabayashi K, Pierre C, Dikin DA, Ruoff RS, Ramanathan T, Brinson LC, Torkelson JM (2008) Polymer- graphite nanocomposites: effective dispersion and major property enhancement via solid-state shear pulverization. Macromolecules 41(6):1905–1908
48. Hamming LM, Qiao R, Messersmith PB, Brinson LC (2009) Effects of dispersion and interfacial modification on the macroscale properties of tio2 polymer-matrix nanocomposites. Compos Sci Technol 69(11–12):1880–1886
49. OCRSpace. Accessed: Dec 2023. https://ocr.space/
50. ExtractTable. Accessed: Dec 2023. https://www.extracttable.com/
51. Hira K, Zaki M, Sheth D, Krishnan N, et al (2023) Reconstructing materials tetrahedron: Challenges in materials information extraction. arXiv preprint arXiv:2310.08383
52. Levenshtein VI, et al (1966) Binary codes capable of correcting deletions, insertions, and reversals. In: Soviet Physics Doklady, vol. 10, pp. 707–710. Soviet Union