Extracting Materials Science Data from Scientific Tables

Defne Circi[1], **Ghazal Khalighinejad**[2], **Anlan Chen**[1], **Bhuwan Dhingra**[2], **L.C. Brinson**[1]
[1]Department of Mechanical Engineering and Materials Science, Duke University, USA
[2]Department of Computer Science, Duke University, USA
{defne.circi, ghazal.khalighinejad, anlan.chen, bhuwan.dhingra, cate.brinson}@duke.edu

Abstract

Advances in materials science depend on leveraging data from the vast published literature. Extracting detailed data and metadata from these publications is challenging, leading current data repositories to rely on newly created data in narrow domains. Large Language Models (LLMs) offer a new opportunity to rapidly and accurately extract data and insights from the published literature, transforming it into structured formats for easy querying and reuse. This paper explores using LLMs for autonomous data extraction from materials science articles, focusing on polymer composites to demonstrate successes and challenges in 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, which includes polymer names, molecule names used as fillers, and their respective compositions. Additionally, it achieved 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.

1 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 (Horawalavithana et al., 2022). 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.

Therefore, automation of the data curation process has gained increasing attention to enable rapid growth of a robust repository of prior published data (Yang, 2022; Olivetti et al., 2020; Dunn et al., 2022; Foppiano et al., 2023; Shetty and Ramprasad, 2021; Xie et al., 2023; Gilligan et al., 2023; Cheung et al., 2023). 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 (Choi and Lee, 2023; Polak et al., 2023; Kononova et al., 2019; Wang et al., 2022; Shetty et al., 2023; Venugopal et al., 2021). 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 (Circi et al., 2023; Khalighinejad et al., 2024). 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. (2022) found that 85% of compositions and their associated properties are reported only in tables. Thus, tables in the materials science domain contain rich information about the properties and composition of materials. 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 (Sayeed et al., 2023).

There have been a number of efforts to extract data such as compositions and properties of materials from tables. Zhang et al. (2023) 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 (Jensen et al., 2019). Using the same tool to obtain raw XML tables and captions, DiSCoMaT introduced the task of composition information extraction from tables and developed a graph neural network based pipeline to extract glass compositions (Gupta et al., 2022). Zaki et al. (2023) found that using advanced LLMs such as GPT-4 to extract composition performed worse than a graph neural network model and suggested task specific prompting strategies and fine-tuning in domain-specific datasets. Bai et al. (2023) introduced the schema-driven information extraction task, which uses the source code of a table to produce a sequence of JSON objects, each representing a cell in the table. They considered various domains and showed that the data format affects the performance of information extraction from tables particularly in the domain of chemistry. Oka et al. (2021) 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. 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. Our task differs from Bai et al. (2023) in that we are not focusing on extracting information from each cell and its attributes. We aim to evaluate samples, where each attribute may come from multiple cells, and some cells may not contribute to any sample. Our task involves two implicit steps: information extraction from each cell and then aggregating that information into samples. We solve both steps using a single prompt. 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.

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. One key challenge in extracting information from tables in the polymer composite domain is dealing with the complexity and inconsistency in the literature of polymer molecule names. Our study confronted several other challenges, detailed in Appendix A, 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. (2023), 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.

2 Methods

2.1 Article and dataset preparation

The data for this study consists of tables containing information about polymer nano- and microcomposite samples. The articles were selected from MaterialsMine (McGuinness et al., 2022). 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 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.

2.2 Choosing inputs of table data

Here, we describe the approaches that were used for obtaining inputs of table data. All 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 Figure 1.

2.2.1 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. To extract and interpret the data from these table images, we utilized GPT-4 Turbo with vision capabilities.

2.2.2 GPT-4 on unstructured OCR extraction from table image

For digitizing table content using OCR, we chose OCRSpace (OCR). 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. Despite this limitation, OCRSpace's free API makes it a highly accessible for converting large volumes of data, with a rate limit of 500 requests within one day.

2.2.3 GPT-4 on structured table output from pdf

We utilized the ExtractTable tool (Ext) 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. Although the tool does not include table captions, it does maintain the tabular format which makes information extraction efficient. We generated 2 types of 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.

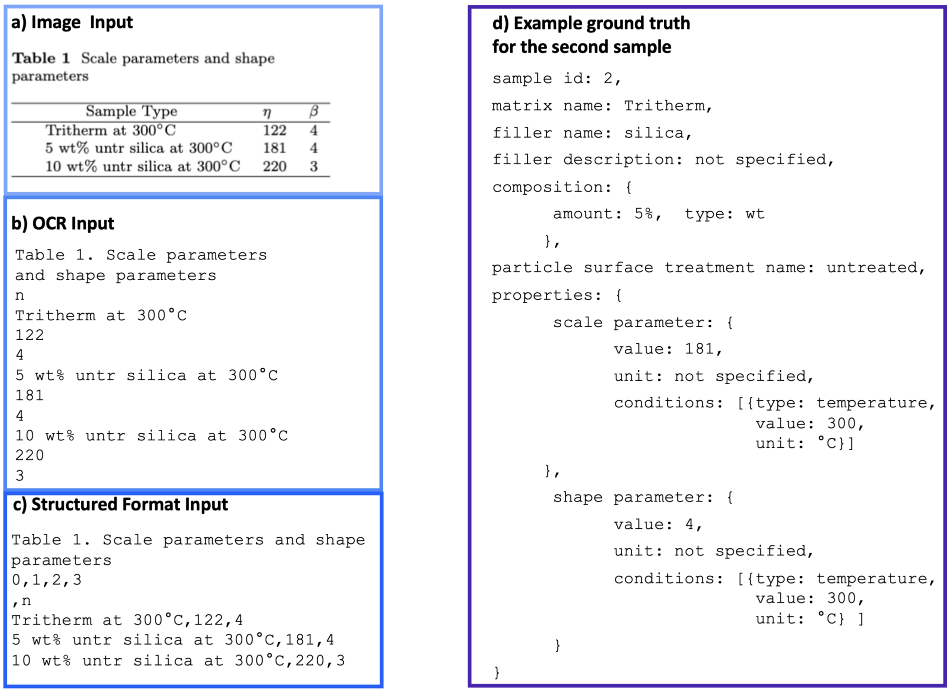
2.3 Prompt design

Polymer composites are a class of materials comprising a polymeric matrix with embedded nanoparticle or microparticle fillers. These fillers often have surface chemical groups added to improve the properties of the resulting composite. 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. In our study, we 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., 2023). 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 section 2.2, the type of input table to be incorporated in the prompt is determined. The prompt also includes two example samples to make the outputs more consistent.

2.4 Evaluation

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 (section 2.2)—image-based extraction, OCR, and structured data extraction—against the set of annotated ground truth tables. In our table extraction process, we observed that the sequence of samples extracted by the model usually aligns



a) Image Input

Table 1 Scale parameters and shape parameters

Sample Type	η	β
Tritherm at 300°C	122	4
5 wt% untr silica at 300°C	181	4
10 wt% untr silica at 300°C	220	3

b) OCR Input

```
Table 1. Scale parameters
and shape parameters
n
Tritherm at 300°C
122
4
5 wt% untr silica at 300°C
181
4
10 wt% untr silica at 300°C
220
3
```

c) Structured Format Input

```
Table 1. Scale parameters and shape
parameters
0,1,2,3
,n
Tritherm at 300°C,122,4
5 wt% untr silica at 300°C,181,4
10 wt% untr silica at 300°C,220,3
```

d) Example ground truth for the second sample

```
sample id: 2,
matrix name: Tritherm,
filler name: silica,
filler description: not specified,
composition: {
      amount: 5%,  type: wt
     },
particle surface treatment name: untreated,
properties: {
      scale parameter: {
            value: 181,
            unit: not specified,
            conditions: [{type: temperature,
                          value: 300,
                          unit: °C}]
     },
      shape parameter: {
            value: 4,
            unit: not specified,
            conditions: [{type: temperature,
                          value: 300,
                          unit: °C} ]
      }
}
```

Figure 1: Example of the three different input types: a) GPT-4-Vision on sample table image (simulated table inspired by (Travelpiece et al., 2009)) 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

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 have considered several factors affecting the evaluation, leading us to establish different criteria for evaluating various entities. This approach contrasts with Bai et al. (2023), where the same threshold or exact match is used for all entities. In the following sections, we discuss these factors in detail.

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.

- **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". (Bai et al., 2023) also observed that models encounter difficulties in accurately generating complete JSON records. 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."* 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.

- **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 method focused on the quality of the data that was actually extracted, disregarding the impact of the samples that were not extracted.

2.4.1 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: polymer molecule name (matrix), molecule name (filler), composition (amount and type), and a surface molecule attached to the filler (PST). 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 Figure 2:

- **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, polymer matrix names “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.

- **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.

- **Handling numeric values and percentages:** 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 Figure 2, the “no” branch.

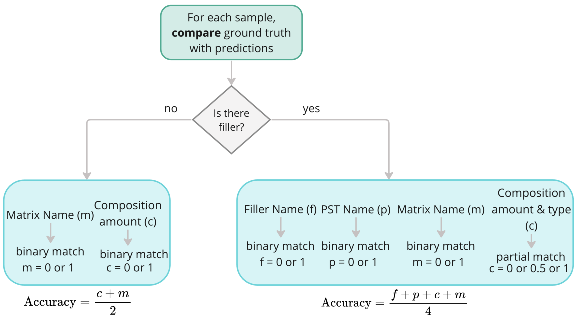
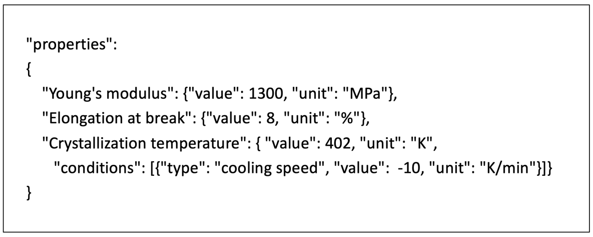


Figure 2: Flowchart illustrating calculation of accuracy score for composition information considering matrix name (m), filler name (f), PST name (p) and composition (c). Note that some flexibility is allowed in matching m, f, and p in that substring matches are allowed.

2.4.2 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 Figure 3. 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"}]}
}
```

Figure 3: Property information example JSON illustrating a few of the wide variety of property names, parameters and conditions appearing in tables containing material property information.

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.

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 method (Levenshtein et al., 1966) 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 Figure 7. 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.

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 & Units:** We used an equality check, where a score of 1 is assigned for an exact match and 0 for a mismatch.

- **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. 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 4 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 3 entities in each condition. This normalization adjusts the final score to fall within a range between 0 and 1.

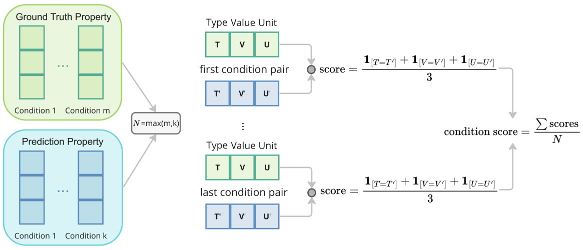

Ground Truth Property
Condition 1
Condition m
Prediction Property
Condition 1
Condition k
N=max(m,k)
Type Value Unit
T
V
U
first condition pair
T'
V'
U'
score = $\frac{\mathbf{1}_{[T=T']}+\mathbf{1}_{[V=V']}+\mathbf{1}_{[U=U']}}{3}$
Type Value Unit
T
V
U
last condition pair
T'
V'
U'
score = $\frac{\mathbf{1}_{[T=T']}+\mathbf{1}_{[V=V']}+\mathbf{1}_{[U=U']}}{3}$
condition score = $\frac{\sum \text{scores}}{N}$

Figure 4: 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.

3 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 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, in some cases predictions were missing or the obtained JSONs were invalid. In all input cases, there were some missing samples. Details of the challenges of different input types and examples of tables can be found in Appendix B and Appendix C.

Composition Information Extraction

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. 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.

Property Information Extraction

For the matching of property names between predicted samples and ground truth samples considering property names as explained in section 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”, 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 textual and visual cues from images, enhancing its understanding of the table’s structure providing a richer context.

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 Figure 5, we reported F_1 scores demon-

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

Input type/Including missing samples	no	yes
Image	0.917 ± 0.036	**0.910** ± **0.037**
OCR	0.890 ± 0.065	0.790 ± 0.107
Structured Format (with captions)	**0.948** ± **0.032**	0.816 ± 0.113
Structured Format (without captions)	0.890 ± 0.056	0.832 ± 0.089

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

strating how well the details of the properties were extracted after the properties were matched with varying thresholds when all the samples were considered. The value of the threshold determines the acceptable average of the correctness scores of the 3 fields in properties: value, type and conditions as explained in section 2.4.2. 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 2 of the 3 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 cause smaller changes in the score. The reported value is the average of the three fields: value, unit and conditions.

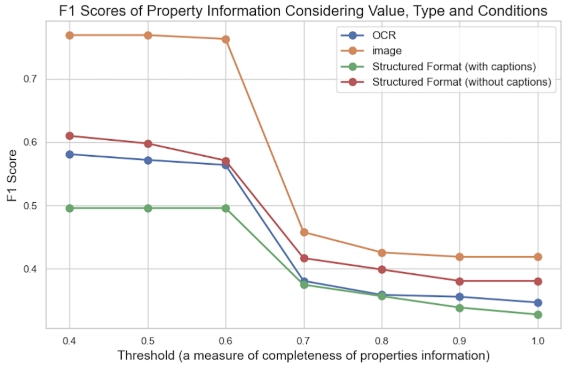


Figure 5: F_1 scores of property information considering value, type and conditions for input types image, OCR, structured format (with captions) and structured format (without captions) based on different thresholds

.

Interestingly, the structured format without captions performed better than with captions as seen in Figure 5. We believe this is 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 F_1 score.

3.1 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. 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. 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, posing challenges for evaluation. To navigate this complexity, we implemented an evaluation approach that first matches the property names in the ground truth and the predictions.

Input type	precision	recall	F_1
Image	**0.905 ± 0.074**	**0.844 ± 0.086**	**0.863 ± 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

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

It 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.

4 Limitation and future opportunities

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. Future work could explore extending these approaches to extract relevant information from figures as well. Additionally, incorporating more flexibility around the extraction of polymer and molecular names would be valuable, allowing for variations that might appear in different articles.

5 Conclusion

Our work developed a method to compare different methodologies for materials science data extraction from tables using GPT-4 offering insights into the effectiveness 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. The code and data are made available in the GitHub repository of this work.

Acknowledgments

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.

References

Extracttable. Accessed: Dec 2023.

Ocrspace. Accessed: Dec 2023.

Fan Bai, Junmo Kang, Gabriel Stanovsky, Dayne Freitag, and Alan Ritter. 2023. Schema-driven information extraction from heterogeneous tables. *arXiv preprint arXiv:2305.14336*.

Jerry Junyang Cheung, Yuchen Zhuang, Yinghao Li, Pranav Shetty, Wantian Zhao, Sanjeev Grampurohit, Rampi Ramprasad, and Chao Zhang. 2023. Polyie: A dataset of information extraction from polymer material scientific literature. *arXiv preprint arXiv:2311.07715*.

Jaewoong Choi and Byungju Lee. 2023. Accelerated materials language processing enabled by gpt. *arXiv preprint arXiv:2308.09354*.

Defne Circi, Ghazal Khalighinejad, Shruti Badhwar, Bhuwan Dhingra, and L Brinson. 2023. Retrieval of synthesis parameters of polymer nanocomposites using llms. In *AI for Accelerated Materials Design-NeurIPS 2023 Workshop*.

Alexander Dunn, John Dagdelen, Nicholas Walker, Sanghoon Lee, Andrew S Rosen, Gerbrand Ceder, Kristin Persson, and Anubhav Jain. 2022. Structured information extraction from complex scientific text with fine-tuned large language models. *arXiv preprint arXiv:2212.05238*.

Luca Foppiano, Pedro Baptista Castro, Pedro Ortiz Suarez, Kensei Terashima, Yoshihiko Takano, and Masashi Ishii. 2023. Automatic extraction of materials and properties from superconductors scientific literature. *Science and Technology of Advanced Materials: Methods*, 3(1):2153633.

Luke PJ Gilligan, Matteo Cobelli, Valentin Taufour, and Stefano Sanvito. 2023. A rule-free workflow for the automated generation of databases from scientific literature. *arXiv preprint arXiv:2301.11689*.

Tanishq Gupta, Mohd Zaki, NM Krishnan, et al. 2022. Discomat: distantly supervised composition extraction from tables in materials science articles. *arXiv preprint arXiv:2207.01079*.

Lesley M Hamming, Rui Qiao, Phillip B Messersmith, and L Catherine Brinson. 2009. Effects of dispersion and interfacial modification on the macroscale properties of tio2 polymer–matrix nanocomposites. *Composites science and technology*, 69(11-12):1880–1886.

Kausik Hira, Mohd Zaki, Dhruvil Sheth, NM Krishnan, et al. 2023. Reconstructing materials tetrahedron: Challenges in materials information extraction. *arXiv preprint arXiv:2310.08383*.

Sameera Horawalavithana, Ellyn Ayton, Shivam Sharma, Scott Howland, Megha Subramanian, Scott Vasquez, Robin Cosbey, Maria Glenski, and Svitlana Volkova. 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*, pages 160–172.

Le Hui, Linda S Schadler, and J Keith Nelson. 2013. The influence of moisture on the electrical properties of crosslinked polyethylene/silica nanocomposites. *IEEE Transactions on Dielectrics and Electrical Insulation*, 20(2):641–653.

Zach Jensen, Edward Kim, Soonhyoung Kwon, Terry ZH Gani, Yuriy Román-Leshkov, Manuel Moliner, Avelino Corma, and Elsa Olivetti. 2019. A machine learning approach to zeolite synthesis enabled by automatic literature data extraction. *ACS central science*, 5(5):892–899.

Ghazal Khalighinejad, Defne Circi, L Catherine Brinson, and Bhuwan Dhingra. 2024. Extracting polymer nanocomposite samples from full-length documents. *arXiv preprint arXiv:2403.00260*.

Olga Kononova, Haoyan Huo, Tanjin He, Ziqin Rong, Tiago Botari, Wenhao Sun, Vahe Tshitoyan, and Gerbrand Ceder. 2019. Text-mined dataset of inorganic materials synthesis recipes. *Scientific data*, 6(1):203.

Vladimir I Levenshtein et al. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710. Soviet Union.

Deborah McGuinness, Cate Brinson, Wei Chen, Chiara Daraio, Cynthia Rudin, Linda Schadler, Rebecca Cowan, Jamie McCusker, Samuel Stouffer, Neha Keshan, et al. 2022. Materialsmine: An open-source, user-friendly materials data resource guided by fair principles.

Hiroyuki Oka, Atsushi Yoshizawa, Hiroyuki Shindo, Yuji Matsumoto, and Masashi Ishii. 2021. Machine extraction of polymer data from tables using xml versions of scientific articles. *Science and Technology of Advanced Materials: Methods*, 1(1):12–23.

Elsa A Olivetti, Jacqueline M Cole, Edward Kim, Olga Kononova, Gerbrand Ceder, Thomas Yong-Jin Han, and Anna M Hiszpanski. 2020. Data-driven materials research enabled by natural language processing and information extraction. *Applied Physics Reviews*, 7(4).

Maciej P Polak, Shrey Modi, Anna Latosinska, Jinming Zhang, Ching-Wen Wang, Shanonan Wang, Ayan Deep Hazra, and Dane Morgan. 2023. Flexible, model-agnostic method for materials data extraction from text using general purpose language models. *arXiv preprint arXiv:2302.04914*.

Monoj Pramanik, Suneel Kumar Srivastava, Biswas Kumar Samantaray, and Anil Kumar Bhowmick. 2003. Rubber–clay nanocomposite by solution blending. *Journal of Applied Polymer Science*, 87(14):2216–2220.

Hasan M Sayeed, Wade Smallwood, Sterling G Baird, and Taylor D Sparks. 2023. Nlp meets materials science: Quantifying the presentation of materials data in scientific literature.

Pranav Shetty, Arunkumar Chitteth Rajan, Chris Kuenneth, Sonakshi Gupta, Lakshmi Prerana Panchumarti, Lauren Holm, Chao Zhang, and Rampi Ramprasad. 2023. A general-purpose material property data extraction pipeline from large polymer corpora using natural language processing. *npj Computational Materials*, 9(1):52.

Pranav Shetty and Rampi Ramprasad. 2021. Automated knowledge extraction from polymer literature using natural language processing. *Iscience*, 24(1).

RC Smith, C Liang, M Landry, JK Nelson, and LS Schadler. 2008. The mechanisms leading to the useful electrical properties of polymer nanodielectrics. *IEEE Transactions on Dielectrics and Electrical Insulation*, 15(1):187–196.

Yuan Sui, Mengyu Zhou, Mingjie Zhou, Shi Han, and Dongmei Zhang. 2023. Evaluating and enhancing structural understanding capabilities of large language models on tables via input designs. *arXiv preprint arXiv:2305.13062*.

AM Travelpiece, JK Nelson, LS Schadler, and Daniel Schweickart. 2009. Dielectric integrity of silica-pai nanocomposites at elevated temperature. In *2009 IEEE Conference on Electrical Insulation and Dielectric Phenomena*, pages 535–538. IEEE.

Vineeth Venugopal, Sourav Sahoo, Mohd Zaki, Manish Agarwal, Nitya Nand Gosvami, and NM Anoop Krishnan. 2021. Looking through glass: Knowledge discovery from materials science literature using natural language processing. *Patterns*, 2(7).

Katsuyuki Wakabayashi, Cynthia Pierre, Dmitriy A Dikin, Rodney S Ruoff, Thillaiyan Ramanathan, L Catherine Brinson, and John M Torkelson. 2008. Polymer- graphite nanocomposites: effective dispersion and major property enhancement via solid-state shear pulverization. *Macromolecules*, 41(6):1905–1908.

Zheren Wang, Olga Kononova, Kevin Cruse, Tanjin He, Haoyan Huo, Yuxing Fei, Yan Zeng, Yingzhi Sun, Zijian Cai, Wenhao Sun, et al. 2022. Dataset of solution-based inorganic materials synthesis procedures extracted from the scientific literature. *Scientific Data*, 9(1):231.

Tong Xie, Yuwei Wa, Wei Huang, Yufei Zhou, Yixuan Liu, Qingyuan Linghu, Shaozhou Wang, Chunyu Kit, Clara Grazian, and Bram Hoex. 2023. Large language models as master key: Unlocking the secrets of materials science with gpt. *arXiv preprint arXiv:2304.02213*.

Huichen Yang. 2022. Piekm: 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*, pages 57–62.

Mohd Zaki, NM Krishnan, et al. 2023. Mascqa: A question answering dataset for investigating materials science knowledge of large language models. *arXiv preprint arXiv:2308.09115*.

Zian Zhang, Haoxuan Tang, and Zhiping Xu. 2023. Fatigue database of complex metallic alloys. *Scientific Data*, 10(1):447.

A 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., 2023), included: 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 Figure 6. 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 Figure 13. 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., 2022) also reported this challenge of filtering irrelevant information in composition extraction from tables . 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 Figure 7). 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 Figure 8 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 Figure 9. (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 Figure 8, where the Weibull parameter is included parenthetically in a column for the breakdown strength value).

4. Ambiguity of conditions: In this example shown in Figure 6, 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 & 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 Figure 10, 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 Figure 11, 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 Figure 12, 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: Figure 7 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 Figure 9 is predicted to have a value of 1.9 when expected to be 1.9e8 and composition value 4-1/2 in Figure 8 is predicted to be 4-1/2 when 4.5 is correct.

B 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 3 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).

C Examples of Tables

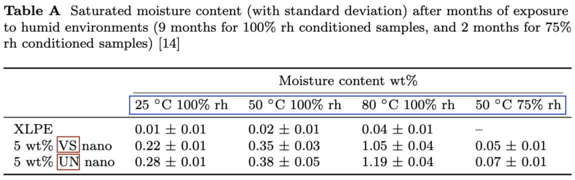


Table A Saturated moisture content (with standard deviation) after months of exposure to humid environments (9 months for 100% rh conditioned samples, and 2 months for 75% rh conditioned samples) [14]

	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

Figure 6: 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 (Hui et al., 2013).

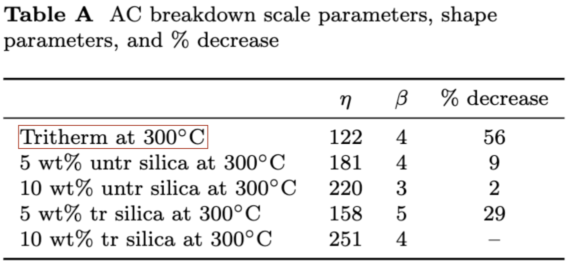


Table A AC breakdown scale parameters, shape parameters, and % decrease

	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	–

Figure 7: 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 (Travelpiece et al., 2009).

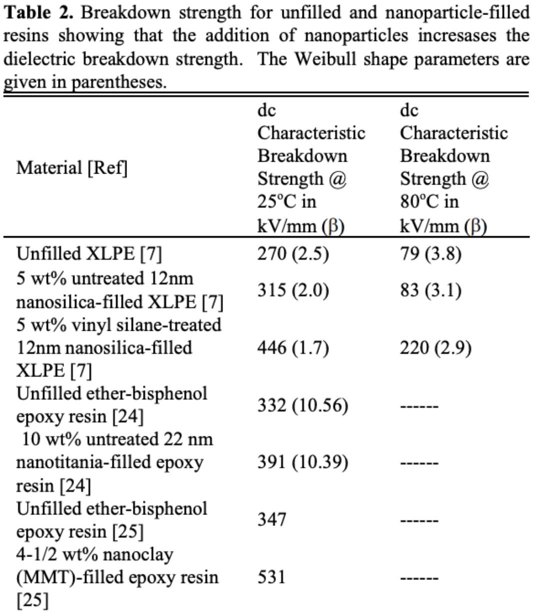


Table 2. Breakdown strength for unfilled and nanoparticle-filled resins showing that the addition of nanoparticles increases the dielectric breakdown strength. The Weibull shape parameters are given in parentheses.

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	------

Figure 8: 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 unconventionally reported numerical values. 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 (Smith et al., 2008). © 2008, IEEE.

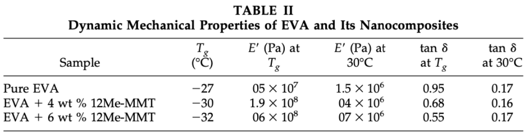


TABLE II
Dynamic Mechanical Properties of EVA and Its Nanocomposites

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

Figure 9: Example of the challenge of understanding unconventionally reported numerical values. 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 (Pramanik et al., 2003). © 2003, Wiley

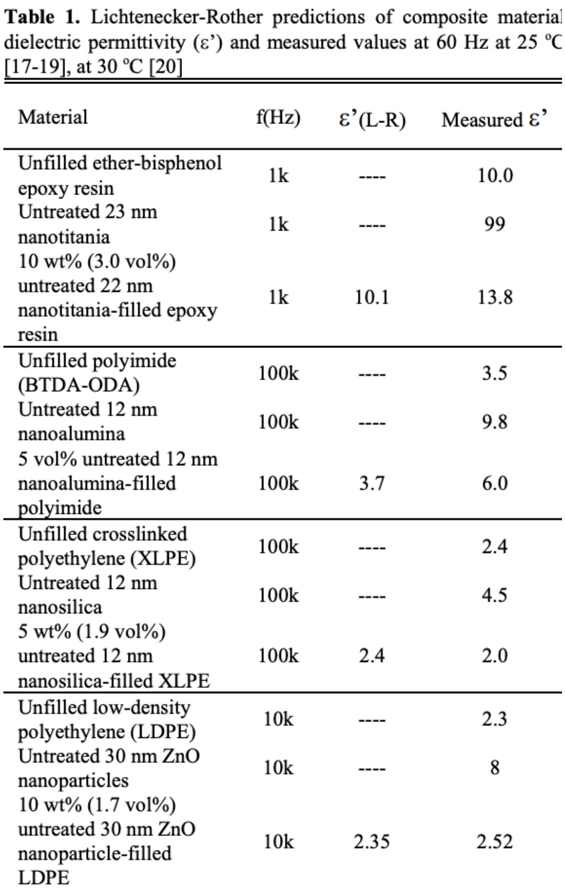


Table 1. Lichtenecker-Rother predictions of composite material dielectric permittivity (ε') and measured values at 60 Hz at 25 °C [17-19], at 30 °C [20]

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

Figure 10: 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 (Smith et al., 2008). © [2008] IEEE.

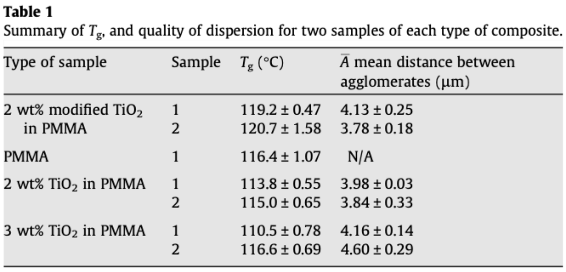


Table 1
Summary of T_g, and quality of dispersion for two samples of each type of composite.

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

Figure 11: 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 2 columns. Reprinted with permission from reference (Hamming et al., 2009). © 2009, Elsevier.

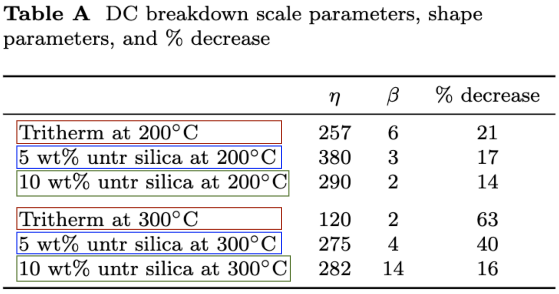


Table A DC breakdown scale parameters, shape parameters, and % decrease

	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

Figure 12: 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 3 unique samples, structured format and image-based input method predicts 6 samples. Simulated table, after (Travelpiece et al., 2009).

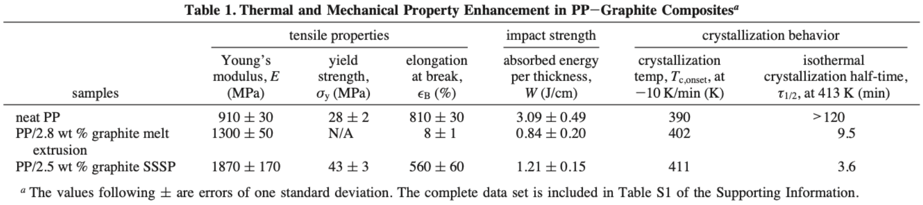


Table 1. Thermal and Mechanical Property Enhancement in PP−Graphite Composites[a]

	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.

Figure 13: 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 (adapted) with permission from (Wakabayashi et al., 2008). © 2008, American Chemical Society.