Benchmarking LLMs on Extracting Polymer Nanocomposite Samples

Anonymous ACL submission

Abstract

This paper investigates the use of large language models (LLMs) for extracting sample lists of polymer nanocomposites (PNCs) from materials science research papers. The challenge lies in the complex nature of PNC samples, which have numerous attributes scattered throughout the text. To address this, we introduce a new benchmark and a novel evaluation technique for this task and examine different LLM prompting strategies: end-to-end prompting to directly generate entities and their relations, as well as a Named Entity Recognition and Relation Extraction (NER+RE) approach, where entities are first identified, followed by relation classification. We also incorporate self-consistency to improve LLM performance. Our findings show that even advanced LLMs, such as GPT-4 Turbo, struggle to extract all of the samples from an article. However, condensing the articles into the relevant sections can help. Finally, we analyze the errors encountered in this process, categorizing them into three main challenges, and discussing potential strategies for future research to overcome them.

1 Introduction

Research publications are the main source for the discovery of new materials in the field of materials science, providing a vast array of essential data. The creation of structured materials databases from these publications is essential for enhancing the speed and efficiency of material discovery. This is evident in the achievements of AI tools such as GNoME (Merchant et al., 2023). Yet, the unstructured presentation of this data in journals makes it challenging to extract valuable information and utilize it for future discoveries (Horawalavithana et al., 2022). Furthermore, manually sorting through articles to extract details about materials—such as their structure, processing, and properties—is time-consuming and prone to errors. Hence, there's a growing need for an automated system that can transform these valuable data into a structured, machine-readable format for more efficient retrieval and analysis (Yang, 2022).

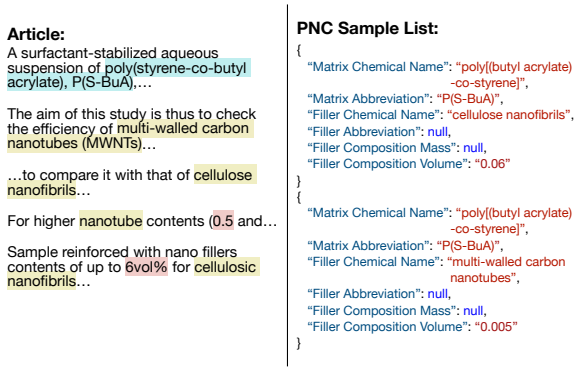


Figure 1: A snippet from a PNC research article (Dalmas et al., 2007) and the extracted PNC sample list from the NanoMine database. Note how information for a single sample is extracted from multiple parts of the article text.

Scientific papers on polymer nanocomposites (PNCs) include detailed descriptions of sample compositions, crucial for understanding their unique properties. PNCs are critical in material science, combining polymer matrices with nanoscale fillers to yield composites with tailored mechanical, thermal, and electrical characteristics. The diversity of PNCs is derived from various matrix and filler combinations, each altering the material's properties. Extracting this data is challenging due to the scattered nature of information across texts, figures, and tables, and the complexity of N-ary relationships where multiple attributes define each sample (Figure 1).

In this paper, we use the NanoMine (Zhao et al., 2018) data repository to construct PNCExtract, a benchmark designed for extracting PNC sample lists from scientific texts using large language mod-

els (LLMs). PNCExtract focuses on the systematic extraction of N-ary relations across different parts of full-length peer-reviewed PNC articles, capturing the unique combination of matrix, filler, and composition in each sample. Prior research on information extraction from materials science literature, such as the works of Dunn et al. (2022), Song et al. (2023a), and Xie et al. (2023), primarily focused on information extraction from specific sentences or passages. PNCExtract, on the other hand, requires models to analyze entire papers to aggregate information dispersed across the various sections of a paper, a key challenge highlighted by Hira et al. (2023). Consequently, we leverage the advanced token limits of recent LLMs like GPT-4 Turbo (OpenAI, 2023) and LongChat (Dacheng Li* and Zhang, 2023) in our study. We also introduce a dual-metric evaluation system comprising a partial metric for detailed analysis of each attribute within an N-ary extraction and a strict metric for assessing overall accuracy. Unlike prior works in materials science that focused solely on an assessment of binary relations (Dunn et al., 2022; Xie et al., 2023; Song et al., 2023a; Wadhwa et al., 2023) or used strict evaluation criteria (Cheung et al., 2023) without recognizing partial matches, our method provides a more comprehensive assessment that acknowledges the complexity of PNC samples.

We explore two prompting strategies for LLMs in a zero-shot context. The first approach aligns with the principles of Named Entity Recognition (NER) and Relation Extraction (RE), which we refer to as NER+RE which involves a two-stage pipeline: initially, entities within the text are identified, and subsequently, valid relations between these entities are extracted, a technique also explored by Zhou et al. (2022) and Tang et al. (2023). However, this approach can become expensive due to the complexity of PNC samples, which feature multiple attributes, leading to an exponential increase in the number of candidate relations. Our second prompting strategy adopts an end-to-end (E2E) method by directly generating the N-ary objects. We find that the E2E approach works better in terms of both accuracy and efficiency. Moreover, we present a simple extension to the self-consistency technique (Wang et al., 2023b) for list-based predictions by sampling multiple times from the LLM and aggregating the lists through majority voting. Our findings demonstrate that this approach improves the accuracy of sample extraction.

Lastly, we discuss three primary challenges encountered when using LLMs for PNC sample extraction. First, many samples are located within tables and figures, indicating a need for multimodal approaches. Second, there is variability in the expression of chemical names; LLMs often use semantically correct but non-standard naming conventions. Finally, the complexity of PNC samples, with their various attributes and diverse chemical names, poses a difficulty as LLMs struggle to differentiate between them. Code for reproducing all experiments is available at `hidden.for.anonymity`.

In summary, we make the following contributions:

- We introduce the PNCExtract benchmark with a novel evaluation method to assess LLMs' ability to extract PNC samples' composition from full-length research articles.
- We explore two prompting strategies, NER+RE and E2E in a zero-shot context. Our findings show that the E2E approach is more accurate and efficient. Furthermore, we develop an extension to the self-consistency technique, tailored for this task, and demonstrate its effectiveness in improving accuracy.
- We identify and discuss three challenges faced in extracting PNC samples with LLMs and suggest potential strategies to address them.

2 PNCExtract Benchmark

In this section we first describe our dataset, including its source, information extraction tasks, preprocessing, and statistics. Then we describe a novel evaluation method for the described task.

2.1 Dataset

2.1.1 NanoMine Data Repository

NanoMine (Zhao et al., 2018) is a PNC data repository structured around an XML-based schema designed for the representation and distribution of nanocomposite materials data. The NanoMine database, manually curated using Excel templates provided to materials researchers, consists of a broad array of potential schema entries. These entries are categorized into several major sections, such as Materials Composition, Processing, and Properties. The Materials Composition section covers characteristics of the constituent materials, including the polymer matrix and the filler particle.

Processing details the description of chemical synthesis. The Properties section provides measured data on materials performance and response, with each section containing numerous entries.

A typical sample in NanoMine uses only a fraction of the possible 350 terms that keep evolving. NanoMine database currently contains a list of 240 full-length scholarly articles and their corresponding PNC sample lists. While NanoMine includes various subfields, our study focuses on the "Materials Composition" section. This section comprehensively details the characteristics of constituent materials in nanocomposites, including aspects like the polymer matrix, filler particles, and their compositions (expressed in volume or weight fractions). The reason for this focus is that determining which samples's composition were studied in a given paper is the essential first step towards identifying and understanding more complex properties of PNCs. Out of the 240 articles, we focus on 193 and disregard the rest due to having inconsistent format (see Appendix A). These 193 articles contain a total of 1052 samples.

2.1.2 Dataset Curation and Cleaning

During our curation process, we selectively disregard certain attributes from NanoMine based on three criteria:

- Complexity in Extraction and Evaluation: Attributes that cannot be directly extracted with a language model or evaluated are disregarded. For example, intricate descriptions (such as "an average particle diameter of 10 um") are excluded due to their complexity in evaluation.
- Rarity in the Dataset: We also disregard attributes infrequently occurring in NanoMine. For instance, "Tacticity" is noted in only 0.05% of samples. This rarity might stem from either its infrequent mention in research papers or oversights by annotators.
- Relative Importance: Attributes that are less important for our analysis, such as "Manufacturer Or Source Name", are also excluded. Our focus is on extracting attributes that are most relevant for identifying a nanocomposite sample.

This filtering process retains 6 out of the 43 total attributes in the Materials Composition of NanoMine.

2.1.3 Problem Definition

We define our dataset as $\mathcal{D} = \{D_1, D_2, \ldots, D_{193}\}$, where each D_i is a peer-reviewed paper included in our study. Corresponding to each paper D_i, there is an associated list of samples $\mathcal{S}_i$, comprising various PNC samples. Formally, $\mathcal{S}_i$ is defined as $\mathcal{S}_i = \{s_{i1}, s_{i2}, \ldots, s_{in_i}\}$, where s_{ij} represents the j-th PNC sample in the sample list of the i-th paper, and n_i denotes the total number of PNC samples in $\mathcal{S}_i$. Each paper has 5.72 samples on average. Each sample s_{ij} is a JSON object with six entries: Matrix Chemical Name, Matrix Chemical Abbreviation, Filler Chemical Name, Filler Chemical Abbreviation, Filler Composition Mass, and Filler Composition Volume. Table 1 presents the count of samples with each attribute marked as non-null. The primary task involves extracting a set of samples $\hat{\mathcal{S}}_i$ from a given paper D_i.

Attribute	Number of Samples
Matrix Chemical Name	1052
Matrix Chemical Abbreviation	864
Filler Chemical Name	1052
Filler Chemical Abbreviation	819
Filler Mass	624
Filler Volume	407

Table 1: Number of total samples for which each of the attributes is non-null.

2.2 Evaluation Metrics

Our task involves evaluating the performance of our model in predicting PNC sample lists. One natural approach, also utilized by Cheung et al. (2023), is to verify if there is an exact match between the predicted and the ground-truth samples. This method, however, has a notable limitation, particularly due to the numerous attributes that define a PNC sample. Under such strict evaluation criteria, a predicted sample is considered entirely incorrect if even one attribute is predicted inaccurately, which can be too strict considering the complexity and attribute-rich nature of PNC samples.

Hence, we also propose a partial metric which rewards predicted samples for partial matches to a ground truth sample. However, computing such a metric first requires identifying the optimal matching between the predicted and ground truth sample lists, for which we employ a maximum weight bipartite matching algorithm. This approach acknowledges the accuracy of a prediction even if not all attributes are perfectly matched.

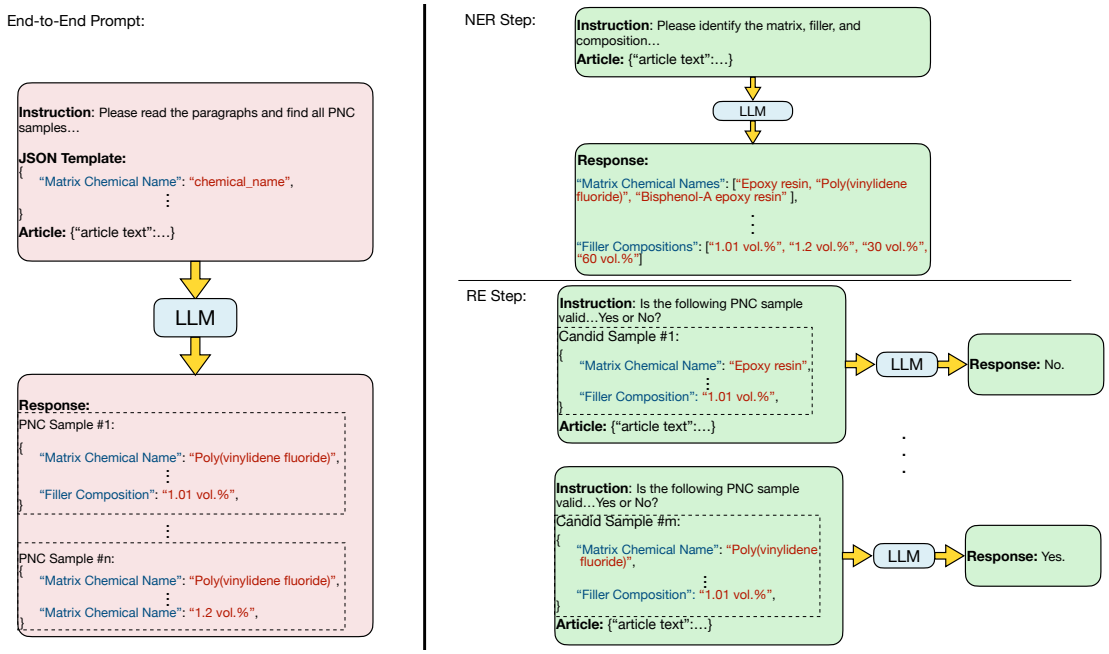


Figure 2: Two prompting strategies for PNC sample extraction with LLM are presented. On the left, the end-to-end (E2E) approach uses a single prompt to directly extract PNC samples. On the right, the NER+RE approach first identifies relevant entities and then classifies their relations through yes/no prompts to validate PNC samples.

Additionally, we also apply a strict metric, similar to the approach of Cheung et al. (2023), where a prediction is considered correct only if it perfectly matches with the ground truth across all attributes of a PNC sample.

Standardization of Prediction To accurately calculate the partial and strict metrics, standardizing predictions is essential. The variability in polymer name expressions in scientific literature makes uniform evaluation challenging. For example, "silica" and "silicon dioxide" are different terms for the same filler. Our dataset from NanoMine uses a standardized format for chemical names. To align the predicted names with this standard, we use resources by Hu et al. (2021), which lists 89 matrix names with their standard names, abbreviations, synonyms, and trade names, as well as, 159 filler names with their standard names. We standardize predicted chemical names by matching them to the closest names in these lists and converting them to their standard forms. Furthermore, our dataset exclusively uses numerical values to represent compositions (e.g., a composition of "0.5vol.%" should be listed as "0.005"). Predictions in percentage format (like "0.5vol.%") are thus converted to the numerical format to align with the dataset's representation.

Attribute Aggregation We implement an attribute aggregation approach in our evaluation. For the "Matrix" category, a prediction is considered accurate if the model correctly identifies either the "Matrix Chemical Name" or the "Matrix Abbreviation". Similarly, in the "Filler" category, accuracy is determined by the correct prediction of either the "Filler Chemical Name" or the "Filler Abbreviation". Lastly, for the "Composition" category, a correct prediction may be based on either the "Filler Composition Mass" or the "Filler Composition Volume". This approach allows for a broader assessment, capturing any correct form of attribute identification without focusing on the finer details of each attribute.

Partial-F1 This metric employs the F_1 score in its calculation, which proceeds in two steps. Initially, an accuracy score is computed for each pair of predicted and ground truth samples where we compute the fraction of matches in the <Matrix, Filler, Composition> trio across the two samples. This process results in $\hat{k} \times k$ score combinations, where $\hat{k}$ and k represent the counts of predicted and ground truth samples. The next step involves translating these comparisons into an assignment problem within a bipartite graph. Here, one set of vertices symbolizes the ground truth samples, and the other represents the predicted samples, with

edges denoting the F_1 scores between pairs. The objective is to identify a matching that optimizes the total F_1 score, which can be computed using the Kuhn-Munkres algorithm (Kuhn, 1955)). in $O(n^3)$ time (where $n = max(\hat{k}, k)$). Note that if $\hat{k} \neq k$, a one-to-one match for each prediction may not be necessary. Once matching is done, we count all the correct, false positive, and false negative predicted attributes (the attributes of all the unmatched predicted samples and ground-truth samples are considered false positives and false negatives, respectively). Subsequently, we calculate the micro-average Precision, Recall, and F_1.

Strict-F1 For a stricter assessment, a sample is labeled correct only if it precisely matches one in the ground truth. Predictions not in the ground truth are false positives, and missing ground truth samples are false negatives. This metric emphasizes exact match accuracy.

3 Modeling Sample List Extractions from Articles with LLMs

Our approach involves the application of LLMs to the task defined in section 2.1.3. We adopt two prompting methods: NER+RE and an End-to-End (E2E) approach in a zero-shot context. Figure 2 illustrates both of these.

3.1 NER+RE Prompt

Building on previous research (Peng et al., 2017; Jia et al., 2019; Viswanathan et al., 2021), which treated N-ary relation extraction as a binary classification task, our NER+RE method treats Relation Extraction (RE) as a question-answering process, following the approach in Zhang et al. (2023). This process is executed in two stages. Initially, the model identifies named entities within the text. Subsequently, it classifies N-ary relations by transforming the task into a series of yes/no questions about these entities and their relations. For evaluation, we apply only the strict metric, as the partial metric is not suitable in this binary classification context.[1]

The NER+RE approach becomes computationally expensive during inference, especially as the number of entities increases. This leads to an exponential growth in potential combinations, expanding the candidate space for valid compositions and consequently extending the inference time.

3.2 End-to-End Prompt

To address this challenge, we develop an End-to-End (E2E) prompting strategy that directly extracts JSON-formatted sample data from articles. This E2E prompt method is designed to efficiently handle the complexity and scale of extracting N-ary relations from scientific texts, bypassing the limitations of binary classification frameworks in this context.

3.3 Self-Consistency

The self-consistency method (Wang et al., 2023b), aims to enhance the reasoning abilities of LLMs. Originally, this method relied on taking a majority vote from several model outputs. For our purposes, since the output is a set of answers rather than a single one, we apply the majority vote principle to the elements within these sets.

To implement this, we generate t predictions from the model, each at a controlled temperature of 0.7. Our objective is to identify which samples appear frequently across these multiple predictions as a sign of higher confidence from the model.

During the evaluation, each model run generates a list of predicted samples from a specific paper. We refer to each list as the k-th prediction, denoted $S_k = \{a_1^k, a_2^k, ..., a_m^k\}$. For each predicted element a_j^i, we determine its match score match_j^i, by counting how frequently it appears across all predictions $\{S_1, S_2, ..., S_t\}$. This score can vary from 1, meaning it appeared in only one prediction, to t, indicating it was present in all predictions.

We then apply a threshold α to filter the samples. Those with a match_j^i at or above α are retained, as they were consistently predicted by the model. Samples falling below this threshold suggest less confidence in the prediction and are removed.

4 Experiments

4.1 Experimental Setup

Models We use LLaMA2 models (Touvron et al., 2023), LongChat model (Dacheng Li* and Zhang, 2023), GPT-4, and GPT-4 Turbo (OpenAI, 2023) models for our experiments. LongChat is finetuned from LLaMA models, which were originally pretrained with 2048 context length. LongChat is finetuned to a context length of 16384. GPT-4 also has a context length of 8000 tokens but the Turbo

[1]While partial evaluation is theoretically possible by considering all potential samples identified in the NER step, such an approach would yield limited insights.

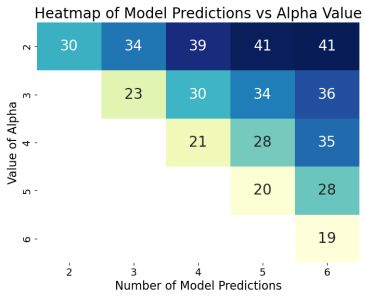


Figure 3: Heatmap showing the strict sample-level F_1 scores achieved by applying self-consistency across varying numbers of GPT-4 Turbo (E2E) predictions at different α values.

version increases this to 128k tokens. The experiments are also done on two different setting where we prompt a full-length paper and where we prompt with a condensed paper.

Heuristics for Condensing Research Papers within LLMs Token Limit LLMs come with token limits, such as 8,192 tokens for the GPT-4 API and 4,096 for LLaMA2. These limits pose a challenge in processing entire research papers, which often exceed these token counts. To address this, we employ simple heuristics to condense the articles effectively. We first divide each paper into distinct sections - the abstract, introduction, experiments, main text, results, and the captions for figures and tables. We keep the title, abstract, and captions for figures and tables unchanged due to their conciseness and rich information content. For the introduction, experiments, main text, and results, we selectively retain only those sentences that contain a digit, which typically indicate crucial composition details. The conclusion section is completely left out, as it often contains repetitive information.[2]

Setup We divide our dataset into 52 validation articles and 141 test articles. We assess the performance using micro average Precision, Recall, and F1 scores, considering both strict and partial metrics at the sample and property levels. We also compare two different prompting strategies NER+RE and E2E. Moreover, we consider the self-consistency technique.

[2]We initially explored retrieval methods but our preliminary results suggested the heuristic-based approach is more effective (Appendix D).

4.2 Results

In Table 2 we report the partial and strict metrics to evaluate multiple models and settings. The results highlight several key observations:

Performance Improvement with Condensed Papers The results indicate that models perform better when provided with condensed versions of papers. In particular, our optimal model, GPT-4 Turbo with self-consistency (SC), achieves a strict F_1 score that is 3.4 points higher and a partial F_1 score that is 4.0 points higher in the condensed paper compared to the full paper setting. Moreover, Table 3 reports the bootstrap analysis from 1000 resamplings, indicating a higher mean F_1 score of 37.4 for GPT-4 Turbo on shorter documents ($0 - 8000$ tokens) compared to a mean F_1 score of 28.5 on longer documents ($8000 - 20000$ tokens).

Comparative Performance: E2E vs. NER+RE: In both condensed and full paper settings, the E2E prompting method shows better performance compared to the NER+RE approach. Specifically, E2E exceeds NER+RE by 4.5 F_1 points in the condensed setting. This performance gap is attributed to the higher precision of E2E. Furthermore, the inference time of the GPT-4 Turbo (E2E) is 28 sec/article in the condensed paper setting, significantly faster than 45 sec/article for GPT-4 Turbo (NER+RE).

Impact of Self-consistency on PNC Sample Extraction: To optimize the application of self-consistency, we first determine the most effective number of predictions to sample and the optimal value for α. Figure 3 shows that the optimal performance is achieved with α at 2 and by sampling 6 predictions. Consequently, we employ these optimal settings for self-consistency on the test set. The results, as reported in Table 2, show that self-consistency enhances the strict and partial F_1 by 2.30 points and 3.3 points, respectively, in the condensed setting. In the full paper setting, the improvement is 5.5 points for strict and 3 points for partial metric.

In addition to our evaluation centered on sample extraction, we present the F_1 scores for different attributes. Table 4 details the performance of GPT-4 Turbo indicating that the model predicts "Filler" attributes more accurately than "Composition", which has lower performance metrics.

Model	Strict			Partial		
	Prec.	Recall	F1	Prec.	Rec.	F1
Condensed Papers						
LLaMA2-7b (E2E)	0.0	0.0	0.0	0.0	0.0	0.0
LLaMA2-7b Chat (E2E)	0.5	0.2	0.2	0.9	0.9	0.9
LongChat-7b-13k (E2E)	5.1	5.6	4.8	31.9	27.1	29.1
GPT-4 (E2E)	30.6	**34.1**	31.8	44.4	43.5	43.8
GPT-4 Turbo (E2E)	43.3	29.4	35.0	66.9	44.0	53.1
GPT-4 Turbo (NER+RE)	27.0	35.2	30.5	-	-	-
GPT-4 Turbo + SC (E2E)	**45.0**	31.8	**37.3**	**69.7**	**47.4**	**56.4**
Full Papers						
LongChat-7b-13k (E2E)	5.3	5.5	4.5	25.8	22.3	23.7
GPT-4 Turbo (E2E)	37.8	22.9	28.5	66.4	39.1	49.2
GPT-4 Turbo (NER+RE)	31.9	34.3	33.0	-	-	-
GPT-4 Turbo + SC (E2E)	42.5	28.3	33.9	65.7	43.4	52.3

Table 2: Precision, Recall, and F_1 of different LLMs on condensed and full papers using strict and partial metrics. The table includes GPT-4 Turbo with NER+RE and E2E prompting, as well as an enhancement on E2E using self-consistency (SC). Models with limited context lengths are evaluated only in the condensed paper scenario.

Length Interval	Mean F_1	SD	95% CI
(0, 8000)	37.4	02.2	(33.0, 41.7)
(8000, 20000)	28.5	05.0	(19.3, 37.8)

Table 3: Comparison of mean F_1 scores, standard deviations, and 95% confidence intervals for different token length intervals.

Attributes	Prec.	Recall	F_1
Matrix	50.2	23.5	32.1
Filler	53.1	25.0	34.0
Composition	44.4	20.4	28.0

Table 4: Micro average precision, recall, and F_1 across the attributes.

4.3 Analysis of Errors

Accurately extracting PNC samples is a complex task, and even state-of-the-art LLMs fail to capture all the samples. We find that out of 1052 ground-truth samples, 773 were not identified in the model's predictions. Furthermore, 364 of the 664 predictions were incorrect. This section discusses three categories of challenges faced by current models in sample extraction and proposes potential directions for future improvements.

Compositions in Tables and Figures NanoMine aggregates samples from the literature, including those presented in tables and visual elements within research articles. As demonstrated in the first example of Figure 4, a sample is derived from the inset of a graph. Our present approach relies solely on language models. Future research could focus on advancing models to extract information from both textual and visual data through multimodal methods.

Disentangling the Complex Components in PNC Samples The composition of polymer nanocomposites (PNC) includes a variety of components such as hardeners and surface treatment agents. A common issue in our model's predictions is incorrectly identifying these auxiliary components as the main attributes. For example, the second row in Figure 4 shows the model predicting the filler material along with its surface treatments instead of recognizing the filler by itself. Going forward, enhancing the model to accurately distinguish and classify the diverse elements in a PNC sample is a key area for development.

Non-standard/Uncommon Chemical Name Predictions The expression of chemical names is inherently complex, with multiple names often existing for the same material. In some cases, predicted chemical names are conceptually accurate yet challenging to standardize. This suggests the necessity for more sophisticated approaches that can handle the diverse and complex representations of chemical compounds. The third example in Figure 4 shows an example of this.

5 Related Work

Early works have focused on training models specifically for the tasks of NER and RE. Building on this, recently Wadhwa et al. (2023) and Wang et al. (2023a) show that LLMs can effectively carry out these tasks through prompting. Inspired by

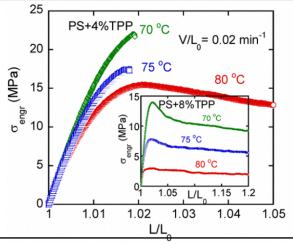
Challenging Example	Ground-truth Sample	Predicted Sample	Explanation
Compositions in Tables and Figures			
	{'Matrix Chemical Name': 'Polystyrene', 'Matrix Abbreviation': 'PS', 'Filler Chemical Name': 'Triphenyl phosphate', 'Filler Abbreviation': 'TPP', 'Filler Mass': '0.08', 'Filler Volume': null}	{'Matrix Chemical Name': 'Polystyrene', 'Matrix Abbreviation': 'PS', 'Filler Chemical Name': 'Triphenyl phosphate', 'Filler Abbreviation': 'TPP', 'Filler Mass': '0.04', 'Filler Volume': null}	The ground-truth sample with a filler mass of 0.08, sourced from a figure inset, was not mentioned in the text and thus not captured.
Disentangling the Complex Components in PNC Samples			
Copolymer grafted SiO2 nanoparticles with a rubbery PHMA inner layer and a matrix compatible PGMA outer layer were prepared	{'Matrix Chemical Name': 'DGEBA Epoxy Resin', 'Matrix Abbreviation': 'epoxy', 'Filler Chemical Name': 'Silicon dioxide', 'Filler Abbreviation': 'SiO2', 'Filler Mass': null, 'Filler Volume': '0.006'}	{'Matrix Chemical Name': 'DGEBA Epoxy Resin', 'Matrix Abbreviation': 'epoxy', 'Filler Chemical Name': 'SiO2/PHMA/PGMA', 'Filler Abbreviation': null, 'Filler Mass': null, 'Filler Volume': '0.006'}	PHMA and PGMA are the chemicals used in particle surface treatment, not the main filler.
Non-standard/Uncommon Chemical Name Predictions			
...the preparation of organophilic clay through the cationic exchange reactions with Na+-montmorillonite clay. The organophilic clay was prepared by a cationic exchange method, which is a reaction between the sodium cations of MMT clay and both intercalation agents of dodecyltriphenyl-phosphonium bromide	{'Matrix Chemical Name': 'Epoxy resin', 'Matrix Abbreviation': 'EPR', 'Filler Chemical Name': 'Montmorillonite', 'Filler Abbreviation': null, 'Filler Mass': '0.01', 'Filler Volume': null}	{'Matrix Chemical Name': 'Epoxy resin', 'Matrix Abbreviation': 'EPR', 'Filler Chemical Name': 'Organophilic clay modified with dodecyltriphenyl-phosphonium', 'Filler Abbreviation': null, 'Filler Mass': '0.01', 'Filler Volume': null}	The predicted filler name in this case is conceptually correct. However, it is not a standard or commonly used chemical name.

Figure 4: Examples of challenges for LLMs, showcasing three categories of challenges encountered in capturing accurate PNC sample compositions. Each row demonstrates a specific challenge, the ground-truth sample, the model's prediction, and a brief explanation of the issue."

these findings, our paper investigates the zero-shot performance of LLMs in materials science.

In the specific area of models trained on a materials science corpus, MatSciBERT (Gupta et al., 2022) employs a BERT (Devlin et al., 2018) model trained specifically on a materials science corpus. Song et al. (2023b) further developed HoneyBee, a fine-tuned Llama-based model for materials science. Our approach differs as we did not engage in fine-tuning, focusing instead on zero-shot performance. Additionally, we did not incorporate MatSciBERT due to its restricted token context and HoneyBee, as its model weights were not accessible during our research phase.

Similar to Dunn et al. (2022), Xie et al. (2023), Tang et al. (2023) and Cheung et al. (2023) our study also focuses on extracting N-ary relations from materials science papers. However, our approach diverges in two significant aspects: we analyze entire papers for PNC sample extraction, not just selected sentences or passages, and we extend our evaluation to partial assessment of N-ary relations, rather than limiting it to binary assessments.

Moreover, Song et al. (2023a) develops a benchmark for BERT models on a materials science dataset, focusing on traditional NLP tasks like NER and RE. Our work, however, evaluates LLMs for sample extraction from full-length papers, a domain where traditional NER and RE methods fall short, and where models like BERT are not viable.

6 Conclusion and Future Works

We introduced PNCExtract, a benchmark focused on extraction of PNC samples from materials science articles. We evaluated NER+RE and E2E prompting strategies on this benchmark and adapted the self-consistency technique for list-based predictions. Our results indicate that condensing materials science papers can notably improve PNC sample extraction tasks. This finding encourages future research to explore more sophisticated retrieval methods for this task.

To overcome the challenges in PNC sample extraction discussed in Section 4.3, future studies could investigate multimodal strategies that integrate text and visual data. Additionally, experimenting with few-shot learning or fine-tuning methods could lead to more precise chemical name generation. Implementing these advancements could significantly enhance the performance of LLMs in extracting PNC samples.

7 Limitation

Although our dataset comprises samples derived from figures within the papers, the current paper is confined to the assessment of language models exclusively. We acknowledge that incorporating multimodal models, which can process both text and visual information, has the potential to enhance the results reported in this paper. Another limitation is that the NanoMine dataset, employed in our analysis, is subject to human curation errors. Consequently, our evaluation assumes the correctness of the NanoMine dataset as ground truth, which could influence the accuracy of our results. Future research could enhance the validity of the evaluation by correcting the errors in the NanoMine dataset. Additionally, our paper selectively examines a subset of attributes from PNC samples. Consequently, we do not account for every possible variable, such as "Filler Particle Surface Treatment." This limited attribute selection means we do not distinguish between otherwise identical samples when this additional attribute could lead to differentiation. Acknowledging this, including a broader range of attributes in future work could lead to the identification of a more diverse array of samples.

8 Ethics Statement

We do not believe there are significant ethical issues associated with this research.

References

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.

Anze Xie Ying Sheng Lianmin Zheng Joseph E. Gonzalez Ion Stoica Xuezhe Ma Dacheng Li*, Rulin Shao* and Hao Zhang. 2023. How long can open-source llms truly promise on context length?

Florent Dalmas, Jean-Yves Cavaillé, Catherine Gauthier, Laurent Chazeau, and Rémy Dendievel. 2007. Viscoelastic behavior and electrical properties of flexible nanofiber filled polymer nanocomposites. influence of processing conditions. *Composites Science and Technology*, 67(5):829–839. Carbon Nanotube (CNT) - Polymer Composites.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Alex Dunn, John Dagdelen, Nicholas Thomas Walker, Sanghoon Lee, Andrew S. Rosen, Gerbrand Ceder, Kristin A. Persson, and Anubhav Jain. 2022. Structured information extraction from complex scientific text with fine-tuned large language models. *ArXiv*, abs/2212.05238.

Tanishq Gupta, Mohd Zaki, N. M. Anoop Krishnan, and Mausam. 2022. Matscibert: A materials domain language model for text mining and information extraction. *npj Computational Materials*, 8(1):102.

Kausik Hira, Mohd Zaki, Dhruvil Sheth, Mausam, and N M Anoop Krishnan. 2023. Reconstructing materials tetrahedron: Challenges in materials information extraction.

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.

Bingyin Hu, Anqi Lin, and L. Catherine Brinson. 2021. Chemprops: A restful api enabled database for composite polymer name standardization. *Journal of Cheminformatics*, 13(1):22.

Chen Jia, Xiaobo Liang, and Yue Zhang. 2019. Cross-domain NER using cross-domain language modeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2464–2474, Florence, Italy. Association for Computational Linguistics.

H. W. Kuhn. 1955. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97.

Amil Merchant, Simon Batzner, Samuel S. Schoenholz, Muratahan Aykol, Gowoon Cheon, and Ekin Dogus Cubuk. 2023. Scaling deep learning for materials discovery. *Nature*, 624(7990):80–85.

OpenAI. 2023. Gpt-4 technical report.

Nanyun Peng, Hoifung Poon, Chris Quirk, Kristina Toutanova, and Wen-tau Yih. 2017. Cross-sentence n-ary relation extraction with graph LSTMs. *Transactions of the Association for Computational Linguistics*, 5:101–115.

Yu Song, Santiago Miret, and Bang Liu. 2023a. Matscinlp: Evaluating scientific language models on materials science language tasks using text-to-schema modeling. *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*.

Yu Song, Santiago Miret, Huan Zhang, and Bang Liu. 2023b. Honeybee: Progressive instruction finetuning of large language models for materials science.

Ruixiang Tang, Xiaotian Han, Xiaoqian Jiang, and Xia Hu. 2023. Does synthetic data generation of llms help clinical text mining?

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

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

Somin Wadhwa, Silvio Amir, and Byron Wallace. 2023. Revisiting relation extraction in the era of large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15566–15589, Toronto, Canada. Association for Computational Linguistics.

Shuhe Wang, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, and Guoyin Wang. 2023a. Gpt-ner: Named entity recognition via large language models.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023b. Self-consistency improves chain of thought reasoning in language models.

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

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.

Kai Zhang, Bernal Jimenez Gutierrez, and Yu Su. 2023. Aligning instruction tasks unlocks large language models as zero-shot relation extractors. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 794–812, Toronto, Canada. Association for Computational Linguistics.

He Zhao, Yixing Wang, Anqi Lin, Bingyin Hu, Rui Yan, James McCusker, Wei Chen, Deborah L. McGuinness, Linda Schadler, and L. Catherine Brinson. 2018. NanoMine schema: An extensible data representation for polymer nanocomposites. *APL Materials*, 6(11):111108.

Jinfeng Zhou, Bo Wang, Minlie Huang, Dongming Zhao, Kun Huang, Ruifang He, and Yuexian Hou. 2022. Aligning recommendation and conversation via dual imitation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 549–561, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

PNC Sample:


```
{
  "Matrix Chemical Name": "polystyrene",
  "Matrix Abbreviation": "PS",
  "Filler Chemical Name": ["octyldimethylmethoxysilane",
                           "silica"]
  "Filler Abbreviation": "ODMMS",
  "Filler Composition Mass": null,
  "Filler Composition Volume": null
}
```


Figure 5: An inconsistent sample in NanoMine that we exclude from our dataset.

A Processing NanoMine

In the sample composition section of NanoMine, various attributes describe the components of a sample. For our analysis, we focus on six specific attributes. Nonetheless, we encounter instances where the formatting in NanoMine is inconsistent. We excluded those articles. This is because our data processing and evaluation require a uniform structure. For example, in Figure 5, we identify an example of an inconsistency where the "Filler Chemical Name" is presented as a list rather than a single value, which deviates from the standard JSON format we expect. This inconsistency makes the sample incompatible with our dataset's format, leading to its removal from our analysis.

B Terms of Use

We used OpenAI (gpt-4 and gpt-4-1106-preview), Llama2, LongChat models, and NanoMine data repository in accordance with their licenses and terms of use.

C Computational Experiments Details

Models Details All of the open-sourced models used in our experiments (e.g. Llama2 and LongChat) have 7 billion parameters.

Computational Budget We perform all of the experiments with one NVIDIA RTX A6000 GPU. Each of the experiments with Llama2 and LongChat took $2-3$ hours.

Hyperparameter Settings For all experiments, except those involving self-consistency, the temperature parameter is set to zero to ensure consistent evaluation of the models. In the case of the self-consistency experiment, we determine the optimal value for the α threshold by tuning α on the validation set, where we explore within the range of $\{2, 3, 4, 5, 6\}$ to identify the optimal value of α. Based on this tuning process, we set the α threshold to 2.

D Condensing Papers with Retrieval Methods

Initially, we employed the dense passage retrieval method for condensing research articles. This technique involved segmenting the articles into smaller chunks and then using OpenAI embeddings to generate embedding vectors with the GPT-4 model. These vectors were subsequently used to perform a semantic search, identifying chunks closest to a specified query. Although this method was considered, the heuristics-based approach proved more effective, as the results from the retrieval models did not perform as well when used to prompt LLMs for our extraction tasks. This suggests that future work could explore developing more advanced retrieval modls for this purpose.

E Prompts

In this section, we present all the prompts used in our experiments.

E.1 E2E Prompt

```
Please read the following paragraphs,
    find all the nano-composite samples,
     and then fill out the given JSON
    template for each one of those
    nanocomposite samples. If there are
    multiple Filler Composition Mass/
    Volume for a unique set of Matrix/
    Filler Chemical Name, please give a
    list for the Composition. If an
    attribute is not mentioned in the
    paragraphs fill that section with "
    null". Mass and Volume Composition
    should be followed by a %.

{
    "Matrix Chemical Name": "
        chemical_name",
    "Matrix Chemical Abbreviation": "
        abbreviation",
    "Filler Chemical Name": "
        chemical_name",
    "Filler Chemical Abbreviation": "
        abbreviation",
    "Filler Composition Mass": "
        mass_value",
    "Filler Composition Volume": "
        volume_value"
}

[PAPER SPLIT]
```

E.2 NER prompt

```
Please identify the matrix name(s),
    filler name(s), and filler
    composition fraction(s). Here is an
    example of what you should return:

{
    "Matrix Chemical Names": ["Poly(
        vinyl acetate)", "Glycerol"],
    "Matrix Chemical Abbreviation": ["
        PVAc"],
    "Filler Chemical Names": ["Silicon
        dioxide"],
    "Filler Chemical Abbreviation": ["
        SiO2"],
    "Filler Composition Fraction":
        ["6%", "12%", "20%", "23%",
        "32%"]
}

[PAPER SPLIT]
```

E.3 RE Prompt

```
Is the following sample a valid polymer
    nanocomposite sample mentioned in
    the article? Yes or No?

Sample:
[JSON OBJECT]

Article:
[PAPER SPLIT]
```