Extracting Polymer Nanocomposite Samples from Full-Length Documents

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

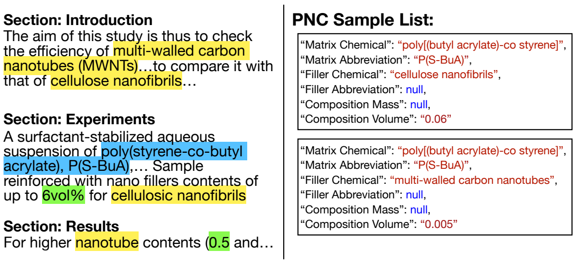
Abstract

This paper investigates the use of large language models (LLMs) for extracting sample lists of polymer nanocomposites (PNCs) from full-length materials science research papers. The challenge lies in the complex nature of PNC samples, which have numerous attributes scattered throughout the text. The complexity of annotating detailed information on PNCs limits the availability of data, making conventional document-level relation extraction techniques impractical due to the challenge in creating comprehensive named entity span annotations. To address this, we introduce a new benchmark and an evaluation technique for this task and explore different prompting strategies in a zero-shot manner. We also incorporate self-consistency to improve the performance. Our findings show that even advanced LLMs struggle to extract all of the samples from an article. Finally, we analyze the errors encountered in this process, categorizing them into three main challenges, and discuss 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. Creating structured databases from these publications enhances discovery efficiency, as evidenced by AI tools like GNoME (Merchant et al., 2023). Yet, the unstructured format of journal data complicates its extraction and use for future discoveries (Horawalavithana et al., 2022). Furthermore, the manual extraction of material details is inefficient and prone to errors, underlining the necessity for automated systems to transform this data into a structured format for better retrieval and analysis (Yang, 2022).

Scientific papers on polymer nanocomposites (PNCs) provide essential details on sample compositions, key to understanding their properties. PNCs, which blend polymer matrices with nanoscale fillers, are significant in materials science for their customizable mechanical, thermal, and electrical characteristics. The variety in PNCs comes from different matrix and filler combinations that modify the properties. However, extracting this data poses challenges due to its distribution across texts, figures, and tables, and the complexity of N-ary relationships defining each sample. An example in Figure 1 illustrates how sample details can be spread over various paper sections.



Section: Introduction
The aim of this study is thus to check the efficiency of multi-walled carbon nanotubes (MWNTs)…to compare it with that of cellulose nanofibrils…

Section: Experiments
A surfactant-stabilized aqueous suspension of poly(styrene-co-butyl acrylate), P(S-BuA),… Sample reinforced with nano fillers contents of up to 6vol% for cellulosic nanofibrils

Section: Results
For higher nanotube contents (0.5 and…

PNC Sample List:

“Matrix Chemical”: “poly[(butyl acrylate)-co styrene]”,
“Matrix Abbreviation”: “P(S-BuA)”,
“Filler Chemical”: “cellulose nanofibrils”,
“Filler Abbreviation”: null,
“Composition Mass”: null,
“Composition Volume”: “0.06”

“Matrix Chemical”: “poly[(butyl acrylate)-co styrene]”,
“Matrix Abbreviation”: “P(S-BuA)”,
“Filler Chemical”: “multi-walled carbon nanotubes”,
“Filler Abbreviation”: null,
“Composition Mass”: null,
“Composition Volume”: “0.005”

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.

In this paper, we construct PNCExtract, a benchmark designed for extracting PNC sample lists from scientific articles. PNCExtract focuses on the systematic extraction of N-ary relations across different parts of full-length PNC articles, capturing the unique combination of matrix, filler, and composition in each sample. Many works have explored N-ary relation extraction from materials science literature (Dunn et al., 2022; Song et al., 2023a,b; Xie et al., 2023; Cheung et al., 2023) and other domains (Giorgi et al., 2022). However, these studies primarily target abstracts and short texts, not addressing the challenge of extracting information from the entirety of full-length articles. PNCExtract addresses this by requiring models to process entire articles, identifying information

Findings of the Association for Computational Linguistics: ACL 2024, pages 13163–13175
August 11-16, 2024 ©2024 Association for Computational Linguistics

Task	Doc-level	N-ary RE	End-to-End
Materials Domain			
PNCExtract	✓	✓	✓
PolyIE	✗	✓	✗
Dunn et al. (2022)	✗	✓	✓
Xie et al. (2023)	✓	✗	✓
MatSci-NLP	✗	✗	✗
Other Domains			
PubMed	✓	✓	✗
SciREX	✓	✓	✗
NLP-TDMS	✓	✓	✗

Table 1: Comparison of PNCExtract with other Information Extraction (IE) approaches within the materials science domain (Song et al., 2023a; Cheung et al., 2023; Dunn et al., 2022; Xie et al., 2023) and across various other scientific domains (Jain et al., 2020; Jia et al., 2019b; Hou et al., 2019). "End-to-End" indicates that, unlike previous methods that require task-specific supervision (e.g., named entity recognition, coreference resolution), PNCExtract relies on end-to-end supervision only.

across all sections, a challenge noted by Hira et al. (2023).

Compared to other document-level information extraction (IE) datasets like SciREX (Jain et al., 2020), PubMed (Jia et al., 2019b), and NLP-TDMS (Hou et al., 2019) which also demand the analysis of entire documents for N-ary relation extraction, our dataset marks the first initiative within the materials science domain. This distinction is important due to the unique challenges of IE in materials science, particularly with polymers. The field features a complex nomenclature with chemical compounds and materials having various identifiers such as systematic names, common names, trade names, and abbreviations, all with significant variability and numerous synonyms for single entities (Swain and Cole, 2016). Furthermore, there is a scarcity of annotated datasets with detailed information, which complicates the creation of effective IE models in this area.

In light of these challenges, our dataset is designed for a generative task to navigate the complexities of fully annotating entire PNC papers, which involve annotating named entity spans, coreferences, and negative examples (entity pairs without a relation). The complexity of PNC papers, due to their various entities and samples, makes manual annotation both time-consuming and prone to errors. Consequently, encoder-only models, which require extensive annotations, fall short for our purposes. In Table 1, we compare PNCExtract with previous IE approaches in the scientific domain.

We 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. This approach distinguishes itself from prior works in materials science, which either focused on the evaluation 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.

We further explore different prompting strategies, including one that aligns with the principles of Named Entity Recognition (NER) and Relation Extraction (RE) which involves a two-stage pipeline, as well as an end-to-end method to directly generate the N-ary object. 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. Our findings demonstrate that this approach improves the accuracy of sample extraction. Since the extended length of articles often exceeds the context limits of some LLMs, we also explore condensing them through a dense retriever (Ni et al., 2022) to extract segments most relevant to specific queries. Our findings indicate that condensing documents generally enhances accuracy. Since existing document-level IE models (Jain et al., 2020; Zhong and Chen, 2021) are not suited for our task, we employ GPT-4 with our E2E prompting on the SciREX dataset and benchmark it against the baseline model. Our analysis shows that GPT-4, even in a zero-shot setting, outperforms the baseline models that were trained with extensive supervision.

Lastly, we discuss three primary challenges encountered when using LLMs for PNC sample extraction. Code for reproducing all experiments is available at https://github.com/ghazalkhalighinejad/PNCExtract.

2 PNCExtract Benchmark

In this section, we first describe our dataset, including the problem definition, and the dataset preparation. Then we describe our evaluation method for the described task.

2.1 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 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 2 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 2: Number of total samples for which each of the attributes is non-null.

Statistics	Paper Length	#Samples/Doc
Avg.	6965	6
Med.	6734	4
Min.	238	1
Max.	16355	50

Table 3: Statistical summary of paper lengths and number of samples per document. Paper length is measured in tokens.

2.2 Dataset Preparation

NanoMine Data Repository We curate our dataset using the NanoMine data repository (Zhao et al., 2018). NanoMine is a PNC data repository structured around an XML-based schema designed for the representation and distribution of nanocomposite materials data. The NanoMine database is manually curated using Excel templates provided to materials researchers. 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 the polymer matrix, filler particles, and their compositions (expressed in volume or weight fractions). The reason for this focus is that determining which sample compositions are studied in a given paper is the essential first step toward 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. For each sample, we retain 6 out of the 43 total attributes in the Materials Composition of NanoMine (see Appendix B for details).

Document-level information extraction requires understanding the entire document to accurately annotate entities, their relations, and saliency. These make the annotation of scientific articles time-consuming and prone to errors. We found that NanoMine also contains errors. Given the challenge of reviewing all 1052 samples and reading through 193 articles, we adopted a semi-automatic approach to correct samples. Specifically, for an article, we consider both the predicted and ground truth sample list of a document. Using our partial metric (detailed in Section 2.3), we match predicted samples with their ground-truth counterparts and assign a similarity score to each pair. Matches are classified as exact, partial, or unmatched—either true samples or predictions. We then focus on re-annotating samples with the most significant differences between prediction and ground truth, especially those partial matches with lower similarity scores and unmatched samples. This method accelerates re-annotation by directing annotators towards specific attributes and samples based on GPT-4 predictions. Following this strategy, we made three types of adjustments to the dataset: deleting 20 samples, adding 15, and editing 19 entities.[1](See Appendix G for details).

2.3 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.

[1] This work includes contributions from polymer experts, under whose mentorship all authors received their training.

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.

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 uses a standardized format for chemical names. To align the predicted names with this standard, we use resources by Hu et al. (2021), which list 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 Our evaluation incorporates an attribute aggregation method. For both the "Matrix" and "Filler" categories, a prediction is considered accurate if the model successfully identifies either the chemical name or the abbreviation. For the "Composition", 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

As mentioned in Section 1, our dataset is designed for a generative task, making encoder-only models unsuitable for two main reasons. First, these models demand extensive annotations, such as named entity spans, coreferences, and negative examples, a process that is both time-consuming and error-prone. Second, encoder-only models struggle with processing long documents efficiently. While some studies have successfully used these models for long documents (Jain et al., 2020), they had access to significantly larger datasets. Our dataset, however, contains detailed domain-specific information, making it challenging to obtain a similarly extensive dataset.

Consequently, within a zero-shot context[2], we explore two prompting methods: Named Entity Recognition plus Relation Extraction (NER+RE) and an End-to-End (E2E) approach.

[2] The context length is prohibitive for attempting few-shot approaches.

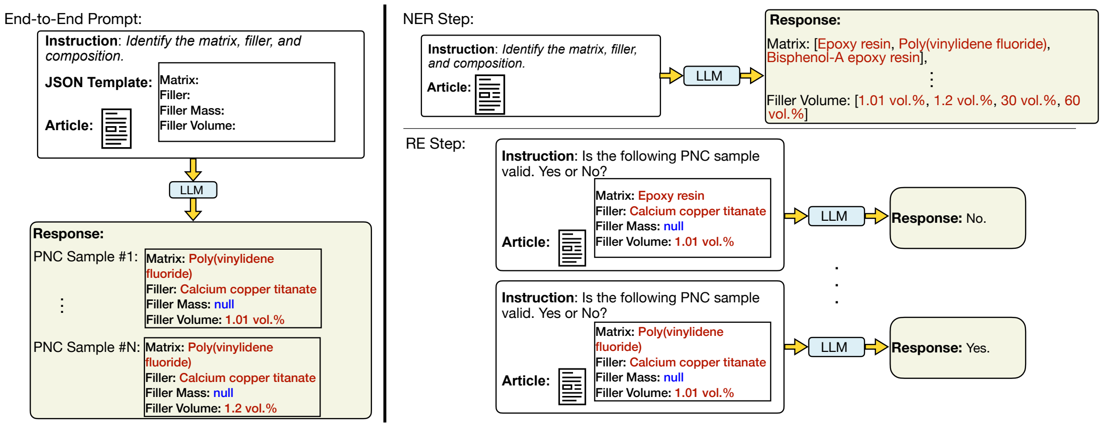

End-to-End Prompt:
Instruction: Identify the matrix, filler, and composition.
JSON Template:
Matrix:
Filler:
Filler Mass:
Filler Volume:
Article:
LLM
Response:
PNC Sample #1:
Matrix: Poly(vinylidene fluoride)
Filler: Calcium copper titanate
Filler Mass: null
Filler Volume: 1.01 vol.%
PNC Sample #N:
Matrix: Poly(vinylidene fluoride)
Filler: Calcium copper titanate
Filler Mass: null
Filler Volume: 1.2 vol.%
NER Step:
Instruction: Identify the matrix, filler, and composition.
Article:
LLM
Response:
Matrix: [Epoxy resin, Poly(vinylidene fluoride), Bisphenol-A epoxy resin],
Filler Volume: [1.01 vol.%, 1.2 vol.%, 30 vol.%, 60 vol.%]
RE Step:
Instruction: Is the following PNC sample valid. Yes or No?
Matrix: Epoxy resin
Filler: Calcium copper titanate
Filler Mass: null
Filler Volume: 1.01 vol.%
Article:
LLM
Response: No.
Instruction: Is the following PNC sample valid. Yes or No?
Matrix: Poly(vinylidene fluoride)
Filler: Calcium copper titanate
Filler Mass: null
Filler Volume: 1.01 vol.%
Article:
LLM
Response: Yes.

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.

3.1 NER+RE Prompt

Building on previous research (Peng et al., 2017; Jia et al., 2019a; Viswanathan et al., 2021), which treated N-ary relation extraction as a binary classification task, our NER+RE method treats 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.[3]

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 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.

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 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.

3.4 Condensing Articles with Dense Retrieval

LLMs, such as LLaMA2 with its token limit of $4,096$, face challenges in maintaining performance with longer input lengths. Recent advancements have extended these limits (Dacheng Li* and Zhang, 2023; Tworkowski et al., 2023); however, an increase in input length often leads to a

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

decline in model performance. This raises the question of whether condensing articles could serve as an effective strategy to address such limitations. We, therefore, employ the Generalizable T5-based Dense Retrievers (GTR-large) (Ni et al., 2022) to retrieve relevant parts of the documents.

This process involves dividing each document C_i into segments ($\{C_{i1}, C_{i2}, ..., C_{iN}\}$) and formulating four queries (Q_j) to extract targeted information regarding an entity.[4] On average, each segment consists of 60 tokens. We then calculate the similarity between each pair of segments and queries (C_{ik}, Q_j). For every query Q_j, we select the top k segments ($TopK(Q_j, C_i)$) based on their similarity scores. These top segments from all four queries are then combined to form a condensed version of the original document ($\bigcup_{j=1}^{4} TopK(Q_j, C_i)$).

4 Experiments

4.1 Benchmarking LLMs on PNCExtract

Models In our experiments, we employ LLaMA-7b-Chat (Touvron et al., 2023), LongChat-7B-16K (Dacheng Li* and Zhang, 2023), Vicuna-7B-v1.5 and Vicuna-7B-v1.5-16K (Chiang et al., 2023), and GPT-4 Turbo (OpenAI, 2023). The LongChat-7B-16K and Vicuna-7B-16K models are fine-tuned for context lengths of 16K tokens, and GPT-4 Turbo for 128K tokens.

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.

4.1.1 Results

In Table 4 we report the partial and strict metrics for multiple models and settings. We report the best results for each model in the condensed paper setting, selected across different $k = \{5, 10, 30\}$, which correspond to average token counts per document of 790, 1420, and 3310, respectively. Further details on the results across various levels of document condensation are available in the Appendix E. The results highlight several key observations:

Model	Strict			Partial		
	P	**R**	**F$_1$**	**P**	**R**	**F$_1$**
			Condensed Papers			
LLaMA2 C	21.7	0.6	1.2	60.0	1.5	3.0
Vicuna	5.8	2.6	3.6	49.9	19.5	28.1
Vicuna-16k	17.7	5.9	8.9	60.4	19.9	29.9
LongChat	6.6	3.5	4.6	47.3	24.4	32.2
GPT-4	43.6	32.0	36.9	64.5	**47.7**	54.8
			Full Papers			
Vicuna-16k	18.4	1.5	2.7	65.7	4.6	8.5
LongChat	5.4	4.2	4.7	36.6	29.6	32.7
GPT-4	44.8	30.2	36.0	64.9	43.8	52.3
GPT-4 (NR)	28.4	**37.2**	32.2	-	-	-
GPT-4 + SC	**51.6**	31.1	**38.8**	**73.5**	43.8	**54.9**

Table 4: 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 different prompting methods (NER+RE, E2E, and E2E with self-consistency [SC]). "NR" denotes NER+RE prompting. "LLaMA2 C" represents the LLaMA2-7b-chat model. Models with limited context lengths are evaluated only in the condensed paper scenario.

Effect of Document Length on the Performance Table 4 demonstrates that shortening documents proves beneficial in most cases. Additionally, Figure 3 shows the trend of partial F$_1$ scores as document length increases. We observe that GPT-4's performance decreases in extremely shortened settings but is optimal when documents are shortened to the top 30 segments. This indicates that while reducing document length is beneficial, excessive shortening may result in the loss of sample information. Additionally, Table 5 provides bootstrap analysis from 1000 resamplings, indicating that GPT-4 Turbo has a higher mean F$_1$ score on shorter full-length documents.

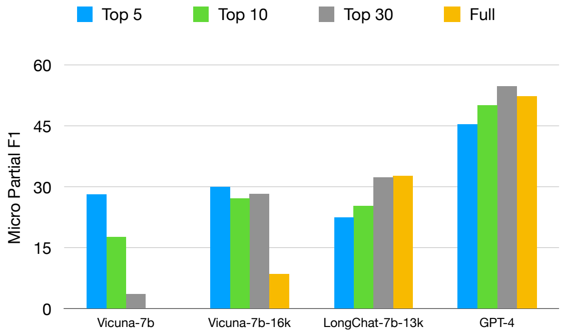


Figure 3: Comparison of Micro Partial F1 Scores Across Different Models and Document Lengths. "Top 5", "Top 10", and "Top 30" indicate document summaries retrieved with k set to 5, 10, and 30 respectively.

[4] The queries are: "What chemical is used in the polymer matrix?", "What chemical is used in the polymer filler?", "What is the filler mass composition?", and "What is the filler volume composition?".

Length Interval	Mean F_1	SD	95% CI
(0, 8000)	44.2	04.0	(35.2, 51.2)
(8000, 20000)	35.2	05.7	(24.4, 46.7)

Table 5: Comparison of mean F_1 scores, standard deviations, and 95% confidence intervals for different token length intervals.

E2E vs. NER+RE: The E2E prompting method shows better performance compared to the NER+RE approach, which is attributed to the higher precision of E2E. Furthermore, the inference time of the GPT-4 Turbo (E2E) is 28 sec/article, 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 on the validation set (see Appendix D). Based on that, we employ $\alpha = 3$ and 8 predictions on the test set. Table 4 shows that self-consistency enhances the strict and partial F_1.

Influence of the Partial Metric Adopting the partial metric has several advantages. First, it helps identify specific challenge areas. For example, in Table 6, we show the model faces the most challenges in accurately predicting Composition. Furthermore, human annotations for PNC samples are often error-prone (Himanen et al., 2019; McCusker et al., 2020), hence one potential use of an LLM like GPT-4 would be to identify errors and send them back for re-annotation. The partial metric can help prioritize which samples to re-annotate.

Attributes	P	R	F_1
Matrix	50.2	23.5	32.1
Filler	53.1	25.0	34.0
Composition	44.4	20.4	28.0

Table 6: Micro average precision, recall, and F_1 across the attributes.

4.2 Comparing with Baselines

Previous works on document-level N-ary IE (Jain et al., 2020; Jia et al., 2019b; Hou et al., 2019), have relied on encoder-only models, making them unsuitable for our specific task. For comparative purposes, we prompt GPT-4 on the SciREX dataset (Jain et al., 2020), which comprises 438 annotated full-length machine learning papers. As shown in Table 7, when prompted in a zero-shot, end-to-end manner, GPT-4 Turbo outperforms the baseline methods. Note that the baseline model, trained on 300 papers, received extensive supervision in the form of mention, coreference, binary relation, and salient mention identification. This suggests that we would need to expend a large amount of annotation effort on PNCExtract to build a supervised pipeline comparable to the zero-shot GPT-4 approach presented here.

Model	Prec.	Rec.	F_1
Jain et al. (2020)	0.7	17.3	0.8
Zero-shot GPT-4	5.0	8.5	**5.5**

Table 7: Micro average precision, recall, and F_1. The baseline results are taken from the referenced paper.

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.

Disentangling the Complex Components in PNC Samples The composition of 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.

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, pre-

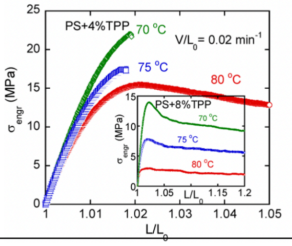
Challenging Example	Ground-truth Sample	Predicted Sample	Explanation
Compositions in Tables and Figures			
	{'Matrix Chemical Name': 'Polystyrene', 'Matrix Abbreviation': 'PS', 'Filler Chemical Name': 'Triphenyl phosphate', 'Filler Abbreviation': 'TPP', 'Filler Mass': '0.08', 'Filler Volume': null}	{'Matrix 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."

dicted 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.

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. MatSciBERT was not applicable to our task, as detailed in Section 3, and HoneyBee's model weights were not accessible during our research phase. Other contributions in this field include studies by Shetty et al. (2023), Hiroyuki Oka and Ishii (2021), and Tchoua et al., focusing on the extraction of polymer-related data from scientific articles.

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 full-length papers, not just selected sentences, and we extend our evaluation to partial assessment of N-ary relations, rather than limiting it to binary assessments.

6 Discussion and Future Works

We introduced PNCExtract, a benchmark focused on the extraction of PNC samples from full-length materials science articles. To the best of our knowledge, this is the first benchmark enabling detailed N-ary IE from full-length materials science articles. We hope that this effort encourages further research into generative end-to-end methods for scientific information extraction from full-length documents. Future investigations should also consider more advanced techniques for condensing entire scientific papers. 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 fine-tuning methods could lead to more precise chemical name generation.

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.

Furthermore, despite our efforts to correct NanoMine, another limitation of our study is the potential presence of inaccuracies within the 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.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

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.

John Giorgi, Gary Bader, and Bo Wang. 2022. A sequence-to-sequence approach for document-level relation extraction. In *Proceedings of the 21st Workshop on Biomedical Language Processing*, pages 10–25, Dublin, Ireland. Association for Computational Linguistics.

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.

Lauri Himanen, Amber Geurts, Adam Stuart Foster, and Patrick Rinke. 2019. Data-driven materials science: Status, challenges, and perspectives. *Advanced Science*, 6(21).

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

Hiroyuki Shindo Yuji Matsumoto Hiroyuki Oka, Atsushi Yoshizawa 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.

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.

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

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.

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

Chen Jia, Xiaobo Liang, and Yue Zhang. 2019a. 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.

Robin Jia, Cliff Wong, and Hoifung Poon. 2019b. Document-level n-ary relation extraction with multi-scale representation learning. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3693–3704, Minneapolis, Minnesota. Association for Computational Linguistics.

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

Jamie P. McCusker, Neha Keshan, Sabbir Rashid, Michael Deagen, Cate Brinson, and Deborah L. McGuinness. 2020. Nanomine: A knowledge graph for nanocomposite materials science. In *The Semantic Web – ISWC 2020*, pages 144–159, Cham. Springer International Publishing.

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.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernandez Abrego, Ji Ma, Vincent Zhao, Yi Luan, Keith Hall, Ming-Wei Chang, and Yinfei Yang. 2022. Large dual encoders are generalizable retrievers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9844–9855, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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.

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).

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.

Matthew C. Swain and Jacqueline M. Cole. 2016. Chemdataextractor: A toolkit for automated extraction of chemical information from the scientific literature. *Journal of Chemical Information and Modeling*, 56(10):1894–1904. PMID: 27669338.

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

Roselyne B. Tchoua, Aswathy Ajith, Zhi Hong, Logan T. Ward, Kyle Chard, Alexander Belikov, Debra J. Audus, Shrayesh Patel, Juan J. de Pablo, and Ian T. Foster. Creating training data for scientific named entity recognition with minimal human effort.

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.

Szymon Tworkowski, Konrad Staniszewski, Mikołaj Pacek, Yuhuai Wu, Henryk Michalewski, and Piotr Miłoś. 2023. Focused transformer: Contrastive training for context scaling.

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.

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

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.

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.

B 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.

C Terms of Use

We used OpenAI (gpt-4 and gpt-4-1106-preview), LLaMA2, LongChat, and Vicuna models, and NanoMine data repository in accordance with their licenses and terms of use.

D Computational Experiments Details

Models Details All of the open-sourced models used in our experiments (e.g. LLaMA2, LongChat, and Vicuna) have 7 billion parameters.

Computational Budget We perform all of the experiments with one NVIDIA RTX A6000 GPU. Each of the experiments with LLaMA2, LongChat, and Vicuna took $2 - 3$ hours.

α	#Predictions	F_1
2	9	39.3
	8	39.2
	7	41.2
	6	40.8
	5	41.4
	4	39.9
3	9	41.8
	8	**43.4**
	7	39.7
	6	39.0
	5	36.0

Table 8: F1 scores for alpha levels 2 and 3, with various numbers of predictions.

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. Table 8 shows that the optimal performance is achieved with α at 3 and by sampling 8 predictions.

E Model Performance on Condensed and Full Papers

Table 9 presents an evaluation of various LLMs across different condensation levels and their performance on full-length papers.

F Prompts

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

F.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]
```

F.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]
```

F.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]
```

Model	Strict			Partial		
	Prec.	Rec.	F_1	Prec.	Rec.	F_1
Condensed Papers (Top 5)						
LLaMA2-7b Chat	9.4	0.4	0.7	41.5	0.9	1.8
LongChat-7b-13k	3.7	1.4	2.1	43.3	15.2	22.5
Vicuna-7b-v1.5	5.8	2.6	3.6	49.9	19.5	28.1
Vicuna-7b-v1.5-16k	17.7	5.9	8.9	60.4	19.9	29.9
GPT-4 Turbo	31.9	18.6	23.5	63.1	35.6	45.5
Condensed Papers (Top 10)						
LLaMA2-7b Chat	21.7	0.6	1.2	60.0	1.5	3.0
LongChat-7b-13k	2.0	0.8	1.1	45.0	17.6	25.3
Vicuna-7b-v1.5	14.7	3.0	5.0	60.0	10.4	17.7
Vicuna-7b-v1.5-16k	15.0	4.9	7.4	58.3	17.8	27.3
GPT-4 Turbo	33.7	23.0	27.3	61.5	42.3	50.1
Condensed Papers (Top 30)						
LongChat-7b-13k	4.7	7.0	3.5	48.2	24.3	32.4
Vicuna-7b-v1.5	6.5	0.2	0.5	55.7	1.8	3.6
Vicuna-7b-v1.5-16k	17.3	5.6	8.4	62.2	18.3	28.2
GPT-4 Turbo	43.6	**32.0**	**36.9**	64.5	**47.7**	**54.8**
Full Papers						
Vicuna-7b-v1.5-16k	18.4	1.5	2.7	65.7	4.6	8.5
LongChat-7b-13k	5.4	4.2	4.7	36.6	29.6	32.7
GPT-4 Turbo	**44.8**	30.2	36.0	**64.9**	43.8	52.3

Table 9: Precision, Recall, and F_1 of different LLMs on condensed and full papers using strict and partial metrics. The results are segmented based on the degree of paper condensation (Top 5, Top 10, Top 30 segments) and for full paper length

G Re-Annotation Example Text

Below, we provide an example of the text that is automatically generated which facilitates the re-annotation.

```
File name: L381

True sample 0 is matched with predicted
    sample 0
But there's a discrepancy between the
    predicted sample and the true sample
      Filler Composition Volume.
True sample: {'Matrix Chemical Name': '
    Polystyrene', 'Matrix Abbreviation':
      'PS', 'Filler Chemical Name': '
    Reduced graphene oxide', 'Filler
    Abbreviation': 'rGO', 'Filler
    Composition Mass': None, 'Filler
    Composition Volume': '0.00428'}
Predicted sample: {'Matrix Chemical Name
    ': 'Polystyrene', 'Matrix Chemical
    Abbreviation': 'PS', 'Filler
    Chemical Name': 'Reduced Graphene
    Oxide', 'Filler Chemical
    Abbreviation': 'rGO', 'Filler
    Composition Mass': 'null', 'Filler
    Composition Volume': '2.10%'}

True sample 5 is matched with predicted
    sample 5
But there's a discrepancy between the
    predicted sample and the true sample
      Filler Composition Volume.
True sample: {'Matrix Chemical Name': '
    Polystyrene', 'Matrix Abbreviation':
      'PS', 'Filler Chemical Name': '
    Reduced graphene oxide', 'Filler
    Abbreviation': 'rGO', 'Filler
    Composition Mass': None, 'Filler
    Composition Volume': '0.0127'}
Predicted sample: {'Matrix Chemical Name
    ': 'Polystyrene', 'Matrix Chemical
    Abbreviation': 'PS', 'Filler
    Chemical Name': 'Reduced Graphene
    Oxide', 'Filler Chemical
    Abbreviation': 'rGO', 'Filler
    Composition Mass': 'null', 'Filler
    Composition Volume': '0.053%'}
Standardized predicted sample: {'Matrix
    Chemical Name': 'Polystyrene', '
    Matrix Chemical Abbreviation': 'PS',
      'Filler Chemical Name': 'Reduced
    Graphene Oxide', 'Filler Chemical
    Abbreviation': 'rGO', 'Filler
    Composition Mass': 'null', 'Filler
    Composition Volume': '0.053%'}

True sample 1 is exactly matched with
    predicted sample 3.

True sample 2 is exactly matched with
    predicted sample 2.

True sample 3 is exactly matched with
    predicted sample 1.

True sample 4 is exactly matched with
    predicted sample 4.
```