Benchmarking LLMs on Extracting Polymer Nanocomposite Samples

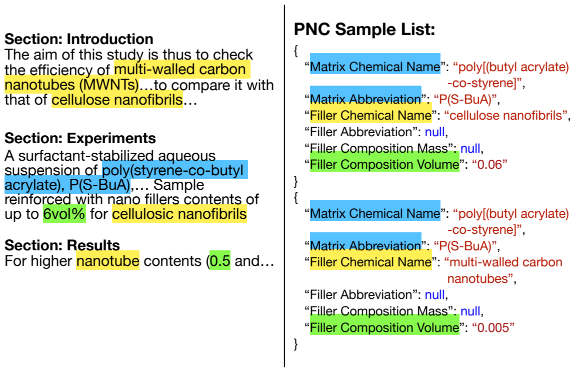
Ghazal Khalighinejad[1], **Defne Circi**[2], **L.C. Brinson**[2], **Bhuwan Dhingra**[1]
[1]Department of Computer Science, Duke University, USA
[2]Department of Mechanical Engineering and Materials Science, Duke University, USA
{ghazal.khalighinejad, defne.circi, cate.brinson, bhuwan.dhingra}@duke.edu

Abstract

This paper investigates the use of large language models (LLMs) for extracting sample lists of polymer nanocomposites (PNCs) from 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.

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 extracting material details from articles is inefficient and error-prone. 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).



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 Name”: “poly[(butyl acrylate)
                          -co-styrene]”,
  “Matrix Abbreviation”: “P(S-BuA)”,
  “Filler Chemical Name”: “cellulose nanofibrils”,
  “Filler Abbreviation”: null,
  “Filler Composition Mass”: null,
  “Filler Composition Volume”: “0.06”
}
{
  “Matrix Chemical Name”: “poly[(butyl acrylate)
                          -co-styrene]”,
  “Matrix Abbreviation”: “P(S-BuA)”,
  “Filler Chemical Name”: “multi-walled carbon
                          nanotubes”,
  “Filler Abbreviation”: null,
  “Filler Composition Mass”: null,
  “Filler 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.

Scientific literature on polymer nanocomposites (PNCs) provides key insights into their compositions and properties, essential for material science innovation. PNCs, made by combining polymer matrices with nanoscale fillers, offer customizable mechanical, thermal, and electrical characteristics. The variety in PNCs comes from different matrix-filler combinations, each impacting the composite's properties. However, extracting this information is challenging due to its dispersion across texts, figures, tables, and the complexity of defining each sample by multiple attributes. An example in Figure 1 illustrates how sample details can be spread over various paper sections.

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 models (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 (see Appendix A.1 for details). Prior research on information extraction from materials science literature, such as the works of Dunn et al. (2022), Song et al. (2023), and Xie et al. (2023), primarily focused on informa-

Model	Strict			Partial		
	Prec.	Rec.	F1	Prec.	Rec.	F1
Condensed Papers						
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						
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 1: Precision, Recall, and F_1 of GPT-4 Turbo 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).

tion 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 GPT-4 Turbo (OpenAI, 2023) in our study.

We also introduce a dual-metric evaluation system, featuring a partial metric for attribute-level analysis within an N-ary extraction and a strict metric for overall accuracy, addressing the limitations of prior works that either focused on binary relations (Dunn et al., 2022; Xie et al., 2023; Song et al., 2023; Wadhwa et al., 2023) or applied strict criteria without acknowledging partial matches (Cheung et al., 2023). This system provides a comprehensive assessment by recognizing the complexity of PNC samples. Evaluating model performance involves checking for exact and partial matches between predicted and ground-truth samples. The partial metric rewards predictions for partial accuracy, employing a maximum weight bipartite matching algorithm to optimally align predicted and ground-truth samples. This approach ensures a balanced evaluation that appreciates the detailed, attribute-rich nature of PNC samples (see Appendix A.2 for details).

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 (see Appendix B for details). 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., 2023) for list-based predictions by sampling multiple times from the LLM and aggregating the lists through majority voting.

In Table 1, we report that models significantly improve when analyzing condensed versions of papers (see Appendix C.1 for details). The E2E prompting method exceeds the NER+RE technique in terms of accuracy. Additionally, applying self-consistency leads to performance improvements in both condensed and full paper settings.

2 Conclusion and Future Works

We introduced PNCExtract, a benchmark for extracting PNC samples from scientific articles, exploring NER+RE and E2E prompting strategies, alongside adapting self-consistency for list-based predictions. We also developed a novel partial evaluation method. Our results indicate that future research should explore more sophisticated retrieval methods. Additionally, there could be significant benefits from adopting multimodal strategies that combine text and visual data, as well as experimenting with few-shot learning or fine-tuning techniques to enhance the precision of chemical name identification.

3 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.

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.

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.

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. 2023. 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)*.

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

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.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. 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.

A PNCExtract Benchmark

A.1 Dataset

A.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. These 193 articles contain a total of 1052 samples.

A.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.

A.1.3 Problem Definition

We define our dataset as $\mathcal{D} = \{D_1, D_2, \dots, D_{193}\}$, 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 $\mathcal{S}_i = \{s_{i1}, s_{i2}, \dots, s_{in_i}\}$, 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 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.

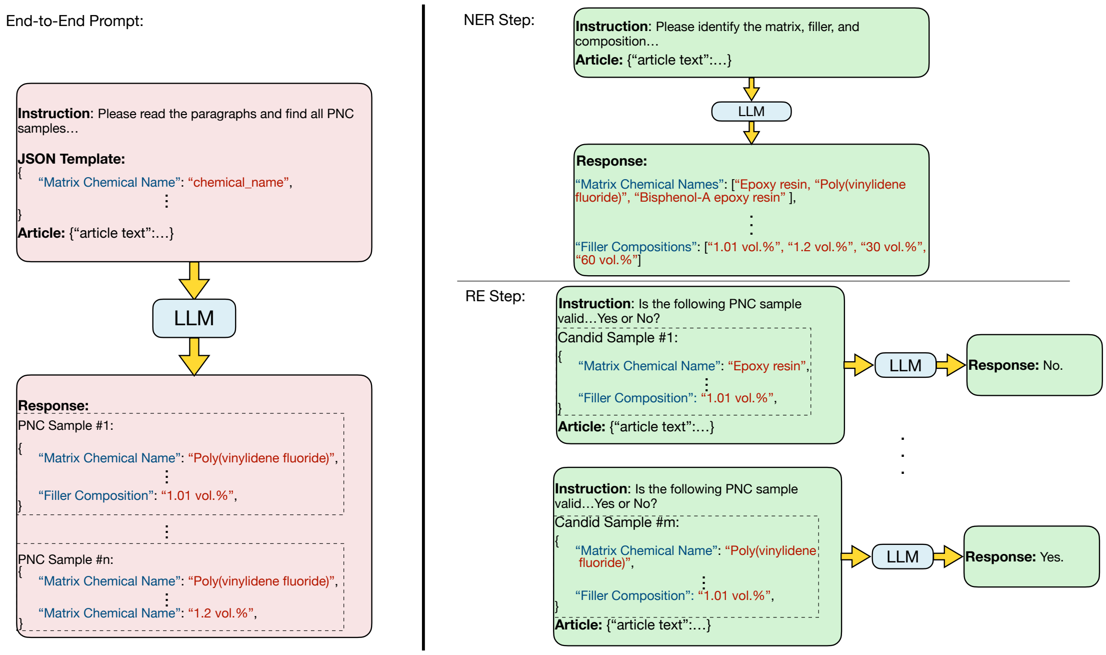

End-to-End Prompt:
Instruction: Please read the paragraphs and find all PNC samples…
JSON Template:
{
"Matrix Chemical Name": "chemical_name",
⋮
}
Article: {"article text":…}
LLM
Response:
PNC Sample #1:
{
"Matrix Chemical Name": "Poly(vinylidene fluoride)",
⋮
"Filler Composition": "1.01 vol.%",
}
⋮
PNC Sample #n:
{
"Matrix Chemical Name": "Poly(vinylidene fluoride)",
⋮
"Matrix Chemical Name": "1.2 vol.%",
}
NER Step:
Instruction: Please identify the matrix, filler, and composition…
Article: {"article text":…}
LLM
Response:
"Matrix Chemical Names": ["Epoxy resin, "Poly(vinylidene fluoride)", "Bisphenol-A epoxy resin"],
⋮
"Filler Compositions": ["1.01 vol.%", "1.2 vol.%", "30 vol.%", "60 vol.%"]
RE Step:
Instruction: Is the following PNC sample valid…Yes or No?
Candid Sample #1:
{
"Matrix Chemical Name": "Epoxy resin",
⋮
"Filler Composition": "1.01 vol.%",
}
Article: {"article text":…}
LLM
Response: No.
Instruction: Is the following PNC sample valid…Yes or No?
Candid Sample #m:
{
"Matrix Chemical Name": "Poly(vinylidene fluoride)",
⋮
"Filler Composition": "1.01 vol.%",
}
Article: {"article text":…}
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.

A.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.

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.

B Modeling Sample List Extractions from Articles with LLMs

Our approach involves the application of LLMs to the task defined in section A.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.

B.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.

B.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.

B.3 Self-Consistency

The self-consistency method (Wang et al., 2023), 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

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

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.

C Experiments

C.1 Experimental Setup

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.

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.

C.2 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 3, 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 3 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 3 shows an example of this.

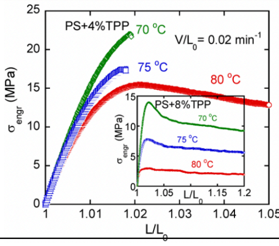
Challenging Example	Ground-truth Sample	Predicted Sample	Explanation
Compositions in Tables and Figures			
	{'Matrix Chemical Name': 'Polystyrene', 'Matrix Abbreviation': 'PS', 'Filler Chemical Name': 'Triphenyl phosphate', 'Filler Abbreviation': 'TPP', 'Filler Mass': '0.08', 'Filler Volume': null}	{'Matrix 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 3: 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."

D Prompts

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

D.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]
```

D.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]
```

D.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]
```