Retrieval of synthesis parameters of polymer nanocomposites using LLMs

Defne Circi
Duke University
Durham, NC 27708
defne.circi@duke.edu

Ghazal Khalighinejad
Duke University
Durham, NC 27708
ghazal.khalighinejad@duke.edu

Shruti Badhwar
Reincarnate Inc.
shruti@reincarnateartificial.com

Bhuwan Dhingra
Duke University
Durham, NC 27708
bdhingra@cs.duke.edu

L. Catherine Brinson
Duke University
Durham, NC 27708
cate.brinson@duke.edu

Abstract

Automated materials synthesis requires historical data, but extracting detailed data and metadata from publications is challenging. We developed initial strategies for using large language models for rapid, autonomous data extraction from materials science articles in a format curatable by a materials database. We used the sub-domain of polymer nanocomposites as our example use case and demonstrated a proof of concept case study via manual validation. We used Claude 2 chat, Open AI GPT-3.5, and 4 API to extract characterization methods and general information about the samples, utilizing zero and few-shot prompting to elicit more detailed and accurate responses. We achieved the best results with an F1 score of 0.88 in the sample extraction task, using Claude 2 chat. Our findings demonstrate the utility of language models for more effective and practical retrieval of synthesis parameters from literature.

1 Introduction

Research publications are the current official repository of reliable information on a huge variety of materials data. It is important to connect data from different resources in materials science, as existing data directs future discoveries and research. However, due to the unstructured nature and highly unique writing and presentation styles, it is difficult to utilize the vast majority of materials data locked in journal articles and reports [1]. Sifting through the articles and determining the structure, processing steps, and properties of each material sample is tedious, time-consuming, and error prone. A robust structured data platform to store, visualize, and analyze materials data is critical for downstream tasks of material discovery, process optimization and virtual metrology/characterization [2], as recently demonstrated by Szymanski et al [3]. NanoMine, part of MaterialsMine, focuses on collecting experimental data from literature on the specific material system of polymer nanocomposites that meets these needs. It is a FAIR (Findable, Accessible, Interoperable, Reusable) [4] data resource, which is important for enabling reuse of data. To date, NanoMine data is collected manually. However, it is impractical to manually curate the data from more than 1 million published papers in

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

this subfield, much less data from more subfields in materials and science. Therefore, automation of the curation process has gained attention increasingly to enable rapid growth of a robust repository of prior published data. [2, 5–10]. Leveraging natural language processing (NLP) and large language models (LLMs) can make the information most important to scientists, such as material identification, composition, properties, or experimental details readily available in a machine-readable format [11–15]. LLMs can be used in other ways to accelerate materials research by classifying documents, correcting annotations, planning experiments, predicting properties and promising structures, making recommendations, and answering questions [12, 16–22]. There has been an effort to extract data such as compositions and properties of materials from tables and figures as important information is often presented in these formats [11, 23–25].

Polymer nanocomposites are a class of materials consisting of a polymeric matrix material in which one or more types of nanoparticle fillers are embedded. These fillers often have surface chemical groups added to them in order to improve the dispersion and properties of the resulting composite [26]. Loadings of particles as small as less than 1 percent by mass have demonstrated dramatic physical property enhancements [27]. Although the details of composition and processing leading to given output properties are still poorly understood, these materials show immense promise for numerous environmental and industrial applications [28]. Given the wealth of information in the literature on these materials (1M papers according to search on Google Scholar), successful data extraction from the published literature could allow for rapid new understanding and discoveries. Our study focuses on the beginning of the process, which is to extract polymer nanocomposite sample information, where each sample is identified by its composition (matrix name, filler name, composition fraction, filler surface treatment) and is associated with synthesis (input) and property (output) details. Here, the focus is on textual data due to additional challenges presented by other modalities, such as structural variations and the presence of partial information [29]. We employ LLMs GPT-3.5, GPT-4, and Claude 2 for this combined named entity recognition and relation extraction task, requiring finding the right entities and the relationships between them for each sample. It is a non-trivial task as it requires recognition of complicated patterns. We aimed to improve curation effectiveness and performance. By carefully constructing prompts guiding the language models to recognize and tag materials entities, we obtained promising results for sample extraction. Having a manually curated set from MaterialsMine allows us to validate the quality of the extracted dataset.

2 Methods

We considered two pipelines for data extraction and evaluation. In Pipeline 1, we extracted the list of samples in each paper, as identified by their composition (matrix and filler names), the composition fraction of filler, and surface functional groups on the fillers; then we performed a manual evaluation. In Pipeline 2, we focused on an automated evaluation procedure, for which we simplified and changed the extraction goal; for a single paper we extracted a representative sample, as identified by a unique matrix name, and the high level processing method and characterization techniques used; in this second pipeline, comparison of higher level information across many papers can be achieved, although the detailed sample list for each paper will be lacking.

2.1 Article and dataset selection

The used article text dataset consists of the abstract, the full text organized with top-level section headers and content, and figure and table captions [30]. We utilized the material sample data extracted from articles in MaterialsMine [31] to develop the ground truth dataset for 42 articles. MaterialsMine contains 240 articles on nanocomposites with a total of 2,512 samples. Each article has an average of 10 samples with a minimum of 2 and a maximum of 49 samples in an article. The detailed sample information is available in rdf or XML files from MaterialsMine, which we downloaded and converted to JSON format. The fields include properties, processing details, conditions, and characterization methods. If all fields were filled, each JSON would contain 400 fields, but in practice most fields are empty, as only a small subset are relevant for a given study. This enabled us to create an accurate ground truth mapping between samples and source articles based on the information available in MaterialsMine.

2.2 Identifying relevant article sections

Due to the 8,192 token limitation of the GPT-4 model that impeded us from feeding the full length article, we needed to reduce the length of the full research articles.[1] Rather than using specific sections like the abstract or methods, or using paragraph classification methods such as using BERT-based classifiers [32], we identified the most relevant sections across the entire article using a vector database. For the pipeline with manual evaluation, we split each article into chunks of less than 1000 characters and then compared the vector embedding of each chunk to the embedding of our search query using a squared L2 similarity score. The embeddings for the article chunks were pre-computed using model "text-embedding-ada-002" provided by OpenAI and stored in the Chroma database [33]. The top 10 most similar chunks for each article were concatenated and used as the input to the GPT-4 model. This allowed us to provide a broad targeted context, while staying within the token limitations of the model.

For the workflow pipeline with automated evaluation, we developed another method for extracting comprehensive information. To capture all necessary data from polymer nanocomposite papers, including materials, processing methods, and characterization, while meeting the 8,192 token limit we kept all the sections but removed sentences that did not contain relevant information.

2.3 Prompt design

Since key differentiating fields are matrix, filler, composition and particle surface treatment (PST), we picked this minimal set to define the samples for the first pipeline. While processing details can also differentiate samples, this information was often sparse so we excluded it in this study. We utilized the strength of zero-shot learning models, which can perform well without any training data, combined with carefully crafted prompts. The models extract the entities and find the relations simultaneously. First, we tried using the generic extraction chain function from LangChain [34] which uses a provided schema. We put every field except the PST in the required list to be extracted since PST is not applied to all nanocomposites. However, this did not sufficiently capture most samples. Therefore, we designed custom prompts to provide more guidance for Claude 2 chat interface (the API access was not available), Open AI GPT-3.5 and 4 API to better guide the model. The prompts include a template JSON file to be filled along with a description of the task. For GPT-3.5 and 4, we also included example output to make the outputs more consistent. The schema used for the extraction function and custom prompts are included in the Appendix. These prompts are evaluated manually.

Given the necessity to evaluate a large number of papers, having an automated pipeline for evaluation is crucial. However, we noticed that evaluating the task of sample extraction has several challenges as it requires determining the most accurate alignment between each predicted sample and its corresponding ground truth sample, taking into account all fields that describe the samples simultaneously. Therefore, we considered a simpler task, focusing on some characteristics that are typically consistent across many samples in a given paper, including several distinguishing characteristics that can be classified categorically. We evaluated GPT-4 based on this new task with a predefined schema to extract n-tuple JSON objects from the papers. Because filler types and surface functional groups often have nearly infinite variation between sample sets in a given paper, we focused on extracting the polymer matrix materials names as a core identifier. In order to extract useful characteristics that can distinguish samples across papers, we captured the key entities of processing methods and characterization techniques. These entities can be identified from high level categories with a list of pre-defined entries, which allows models to find similar, but usually not identical, language in the paper. Processing methods give an overview of the synthesis methodologies used to create the samples and relate to fundamental output properties, and characterization methods give information about the type of figures in the paper and the properties measured for each sample. The prompt we used to prompt GPT-4 that achieved the highest F_1 score can be found in the Appendix.

2.4 Evaluation

Manual evaluation, Pipeline 1 For the extraction of samples, we compared the sample information extracted by GPT-3.5 and 4, and Claude 2 to the ground truth samples from the papers. The ground truth samples are a subset of all the samples that can be found in the paper since we only considered

[1]The experiments were done in September 2023.

the samples that were mentioned in the text, for a fair comparison. A sample was considered correct if the values in the matrix, filler, composition, and PST fields matched any ground truth sample. To allow for flexibility in polymer terminology, exact word matches were not required. F_1 scores were calculated based on the number of samples captured in the outputs. Repeated extractions of the same sample were excluded, as some samples have an identical minimal information set, but differ in other information such as processing details. Samples with empty or non-numerical composition values were also excluded, and not counted as false positives. Therefore, the number of ground truth samples considered was lower than the total number of complete samples. In complete samples, samples that come from figures and tables or have the same minimal information but are different in terms of other processing methods or detailed information such as the particle size of the filler are also present.

Automated evaluation, Pipeline 2 We evaluated the performance of GPT-4 using the F_1 metric for the extraction of consistent sample information task. Our evaluation method compared the predicted JSON objects to the ground truth. Specifically, an entity was correctly predicted if it exactly matches its corresponding ground truth entity. It is noteworthy that incorporating semantic similarity may increase the F1 score. If there is a mismatch between the predicted entity and the ground truth, it is counted as both a false positive and a false negative. If the prediction is null, but there is a ground truth entity, it is classified as a false negative.

3 Results and discussion

Initial experiments for pipeline 1 using the generic Langchain's create extraction chain function and GPT-4 resulted in an F_1 score of 0.26 as shown in Table 1. The low score was likely because it often extracted just one sample, despite most papers having more than one sample using this method. Our customized prompts significantly improved the performance. GPT-3.5 achieved an F_1 of 0.36, GPT-4 0.62 and Claude 0.88. Therefore, the prompts play a key role in guiding the model to extract the relevant information. It was expected that Claude would perform better given its access to the full text, while GPT-4 and GPT-3.5 used selected sections which may have lacked some sample details. We also expected GPT-4 to outperform GPT-3.5 as it is more capable of understanding intricate instructions.

Table 1: The F_1 scores of sample extraction task

Method	F_1
Claude 2	0.88
GPT-4, custom prompt	0.62
GPT-3.5-16k, custom prompt	0.36
GPT-4, create extraction chain function	0.26

Table 2 shows the results of GPT-4's predictions for Pipeline 2 based on various prompts focusing on the properties that are consistent across samples. Parsable accuracy looks at whether the predicted output matches the given template. Our findings indicate that while it is relatively straightforward for the model to extract characterization methods, identifying the correct processing method is more challenging and therefore less accurate. This is because characterization methods are typically explicitly stated in papers. In contrast, processing methods such as melt mixing or solution processing often require reasoning based on the content of the paper.

Table 2: The F_1 score and Parsable accuracy scores of GPT-4 for extracting information that is consistent across samples

Precision	Recall	F_1	Parsable
60.71	60.71	60.71	73.69

A limitation of our current preliminary approaches is that the extraction is biased towards samples that exhibit the desired properties that scientists deliberately report. It is noted from our manual evaluation that samples mentioned explicitly in the text, and therefore extractable by these methods, are the ones that exhibit the desired properties and/or the most noteworthy ones. Focusing on the text only, without considering figures and tables, we capture the subset of all samples that have the

best performance, or the worst performance. The "middle of the pack" samples are rarely called out explicitly in the written text and will require more nuanced work to extract from the papers.

An important criterion to contribute data to a materials database is ensuring faithfulness of the extractions. Our future research includes plans to incorporate additional verification steps prior to storing extracted information to ensure the information stored is reliable. Nonetheless, the approaches described here can accelerate curation by automating parts of the process. Finding the most relevant parts and extracting sample sets from text adds computational steps to curation, thus making it semi-automatic rather than fully manual.

While in this work we focused on zero-shot and few-shot learning, we believe designing better prompts and chain of thought prompting may further improve performance. Moreover, more experiments should be done to find the optimal chunking method. Although our current research presents a generalized overview with an aggregated F1 score metric, yet we recognize the need for a more granular analysis of the results to understand how the model performs in retrieving different entities and providing insights into specific strengths and areas for improvement. Future work could consider extending these approaches to extract sample information from non-textual inputs such as tables and figures. Additionally, developing evaluation techniques tailored to sample extraction from literature is needed. We will also explore fine-tuning the models and designing prompts to include process details and properties that could better guide materials design. This method allows scientist to explore previously tested process/batch variations and guide them in process optimization.

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

4 Conclusions

We showed the potential for large language models to be effective in extracting sample information from polymer nanocomposite literature. Our prompting approaches improved the models' ability to identify relevant sample details compared to generic extraction methods. While Claude achieved the best performance by leveraging full text access and capturing 123 samples from 42 articles, GPT-4 showed strong capabilities when guided by specific prompts. Further refinement of prompting strategies, inclusion of non-textual data, and development of tailored evaluation metrics are important next steps. Overall, this work demonstrates the promise of leveraging LLMs to expedite the extraction of key polymer nanocomposite sample information from text, thereby accelerating and facilitating database curation. Access to such curated data will enable downstream capabilities such as classification and regression tasks to find relationships between synthesis steps and output properties.

Acknowledgments

We thank NSF DGE-2022040 for the aiM NRT funding support.

References

[1] Sameera Horawalavithana, Ellyn Ayton, Shivam Sharma, Scott Howland, Megha Subramanian, Scott Vasquez, Robin Cosbey, Maria Glenski, and Svitlana Volkova. 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, 2022.

[2] Huichen Yang. 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, 2022.

[3] Nathan J Szymanski, Bernardus Rendy, Yuxing Fei, Rishi E Kumar, Tanjin He, David Milsted, Matthew J McDermott, Max Gallant, Ekin Dogus Cubuk, Amil Merchant, et al. An autonomous laboratory for the accelerated synthesis of novel materials. *Nature*, pages 1–6, 2023.

[4] Mark D Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E Bourne, et al. The fair guiding principles for scientific data management and stewardship. *Scientific data*, 3(1):1–9, 2016.

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

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

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

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

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

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

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

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

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

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

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

[16] Daniil A Boiko, Robert MacKnight, and Gabe Gomes. Emergent autonomous scientific research capabilities of large language models. *arXiv preprint arXiv:2304.05332*, 2023.

[17] Maciej P Polak and Dane Morgan. Extracting accurate materials data from research papers with conversational language models and prompt engineering–example of chatgpt. *arXiv preprint arXiv:2303.05352*, 2023.

[18] Andres M Bran, Sam Cox, Andrew D White, and Philippe Schwaller. Chemcrow: Augmenting large-language models with chemistry tools. *arXiv preprint arXiv:2304.05376*, 2023.

[19] Taicheng Guo, Kehan Guo, Zhengwen Liang, Zhichun Guo, Nitesh V Chawla, Olaf Wiest, Xiangliang Zhang, et al. What indeed can gpt models do in chemistry? a comprehensive benchmark on eight tasks. *arXiv preprint arXiv:2305.18365*, 2023.

[20] Yeonghun Kang and Jihan Kim. Chatmof: An autonomous ai system for predicting and generating metal-organic frameworks. *arXiv preprint arXiv:2308.01423*, 2023.

[21] Jiaxing Qu, Yuxuan Richard Xie, and Elif Ertekin. A language-based recommendation system for material discovery. In *1st Workshop on the Synergy of Scientific and Machine Learning Modeling@ ICML2023*, 2023.

[22] Kevin Maik Jablonka, Qianxiang Ai, Alexander Al-Feghali, Shruti Badhwar, Joshua D Bocarsly, Andres M Bran, Stefan Bringuier, L Catherine Brinson, Kamal Choudhary, Defne Circi, et al. 14 examples of how llms can transform materials science and chemistry: a reflection on a large language model hackathon. *Digital Discovery*, 2023.

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

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

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

[26] AAF Zikry. Dielectric behavior of silica/polyacrylamide nanocomposites. *International Journal of Polymeric Materials*, 57(4):383–395, 2008.

[27] Benjamin J Ash, Richard W Siegel, and Linda S Schadler. Glass-transition temperature behavior of alumina/pmma nanocomposites. *Journal of Polymer Science Part B: Polymer Physics*, 42(23):4371–4383, 2004.

[28] Mohamed SA Darwish, Mohamed H Mostafa, and Laila M Al-Harbi. Polymeric nanocomposites for environmental and industrial applications. *International Journal of Molecular Sciences*, 23(3):1023, 2022.

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

[30] Bingyin Hu. *Data Curation of a Findable, Accessible, Interoperable, Reusable Polymer Nanocomposites Data Resource-Materialsmine*. PhD thesis, Duke University, 2022.

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

[32] Nicholas Walker, John Dagdelen, Kevin Cruse, Sanghoon Lee, Samuel Gleason, Alexander Dunn, Gerbrand Ceder, A Paul Alivisatos, Kristin A Persson, and Anubhav Jain. Extracting structured seed-mediated gold nanorod growth procedures from literature with gpt-3. *arXiv preprint arXiv:2304.13846*, 2023.

[33] Chroma. `https://docs.trychroma.com/`.

[34] Harrison Chase. Langchain. `https://github.com/langchain-ai/langchain`, 2022.

A Appendix

A.1 Query used for similarity match

```
query = """
Read the following paragraphs, find all the nanocomposite samples, and fill out the
JSON template for each one:

{
  "sample ID": "",
  "matrix chemical name": "",
  "filler chemical name": "",
  "composition (mass % or vol % )": "",
  "particle surface treatment name": "",
  "properties": ""
}
"""
```

A.2 Schema

```
schema = {
    "properties": {
        "sample ID": {"type": "integer"},
        "matrix chemical name": {"type": "string"},
        "filler chemical name": {"type": "string"},
        "composition (mass % or vol % )": {"type": "string"},
        "particle surface treatment name": {"type": "string"},
    },
    "required": ["sample ID", "matrix chemical name","filler chemical name",
    "composition (mass % or vol % )"],
}
```

A.3 Prompts

A.3.1 GPT-3.5 & GPT-4

```
query = """
Read the following paragraphs, find all the nanocomposite samples, and fill out the
JSON template for each one. Samples should be unique. Composition is how much filler
is inside the matrix. it should be a
numerical number and unit (%mass or %weight):

//insert text

{
  "sample ID": "",
  "matrix chemical name": "",
  "filler chemical name": "",
  "composition (mass % or vol % )": "",
  "particle surface treatment  chemical name": "",
}

example output
[
{
  "sample ID": "1",
  "matrix chemical name": "Diglycidyl ether of bisphenol-A (DGEBA) epoxy resin",
  "filler chemical name": "BaTiO3 nanoparticles",
  "composition (mass % or vol % )": "50 vol%",
  "particle surface treatment  chemical name": "N/A"
},
{
  "sample ID": "2",
  "matrix chemical name": "Diglycidyl ether of bisphenol-A (DGEBA) epoxy resin",
  "filler chemical name": "BaTiO3 nanoparticles",
  "composition (mass % or vol % )": "50 vol%",
  "particle surface treatment  chemical name": "silane coupling agents"
},
{
  "sample ID": "3",
  "matrix chemical name": "Diglycidyl ether of bisphenol-A (DGEBA) epoxy resin",
  "filler chemical name": "BaTiO3 nanoparticles",
  "composition (mass % or vol % )": "30 vol%",
  "particle surface treatment  chemical name": "silane coupling agents"
}
]
"""
%
```

A.3.2 Claude 2

```
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 and output
complete json that consists of the sample jsons. Composition is the amount of filler
found in the composite. nano-composites can have different compositions, filler names,
matrix names or particle surface treatment names

[PAPER]

JSON template
  {
     "sample ID number:              ,
      "matrix chemical name":      ,
      "filler chemical name":          ,
      "composition (weight % or mass %)'':          ,
     "particle surface treatment name":
   }
```

A.4 Automated Evaluation

```
JSON Template:
  {
     ‘‘Matrix Chemical Name’’: " " ,
      ‘‘Processing Method’’: ‘‘Solution Processing/Melt Mixing/In Situ Polymerization’’ ,
      ‘‘Characterization’’: "Transmission electron microscopy/
      Scanning electron microscopy/
      Atomic force microscopy/
      Optical microscopy/
      Fourier transform infrared spectroscopy/
      Dielectric and impedance spectroscopy analysis/
      Raman spectroscopy/
      Differential scanning calorimetry"
   }

Please read the following paragraphs, and fill out the given JSON template.
For processing method and characterization, choose between the given options.

[PAPER SPLIT]
```