JCIM JOURNAL OF CHEMICAL INFORMATION AND MODELING

pubs.acs.org/jcim Article

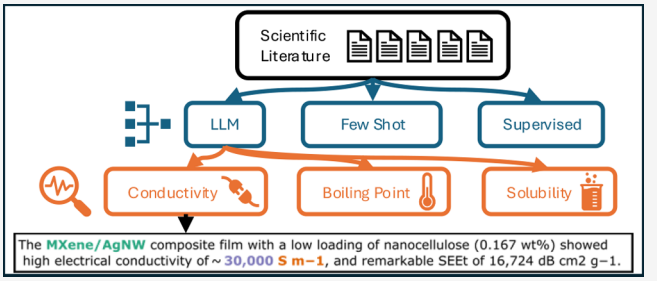
Extracting Material Property Measurements from Scientific Literature with Limited Annotations

Jessica Kong,*,‡ Gihan Panapitiya, and Emily Saldanha*

Cite This: J. Chem. Inf. Model. 2025, 65, 4906−4917 Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: Extracting material property data from scientific text is pivotal for advancing data-driven research in chemistry and materials science; however, the extensive annotation effort required to produce training data for named entity recognition (NER) models for this task often makes it a barrier to extracting specialized data sets. In this work, we present a comparative study of the conventional, supervised NER methodology to alternative few-shot learning architectures and large language model (LLM)-based approaches that mitigate the need to label large training data sets. We find that the best-performing LLM (GPT-4o) not only excels in directly extracting relevant material properties based on limited examples but also enhances supervised learning through data augmentation. We supplement our findings with error and data quality assessments to provide a nuanced understanding of factors that impact property measurement extraction.



INTRODUCTION

Named entity recognition (NER) is the primary method to automatically extract material property data from scientific literature, playing a key role in the information retrieval processes within the closely related fields of chemistry and materials science.[12,27,11] The extraction of such data for developing and maintaining data sets is crucial for scientific modeling and underpins data-driven research.[24] However, applying NER at a large scale requires extensive, expert manual annotations, which presents a significant obstacle.

In research environments, highly specialized data sets tailored to specific study areas and types of information are required. However, the process of labeling data sets for model training can be excessively laborious and time-consuming. This task becomes particularly daunting when data extraction is merely a fraction of an overall research pipeline. In such scenarios, conventional data extraction methods may not be feasible or efficient.

In addition to the annotation challenges, compared with traditional NER applications, the task of material property extraction has several other unique complications. For example, extraction requires the parsing and understanding of numerical entities, and numerical reasoning has been a weak point of a range of natural language processing models.[29] Additionally, the technical language and domain-specific knowledge that are needed for understanding the scientific text in this field present a barrier to using models pretrained on more general types of language. Given these challenges, it is not clear that methods that perform well on more generic NER tasks will also work well on these more niche extraction tasks, motivating the need for domain-specific evaluations of a range of methods and custom solutions.

In settings with limited annotation availability, one potential solution is to utilize few-shot learning (FSL) architectures, which have the ability to classify fine-grained entity types using only a few examples.[9] These models offer an attractive alternative by mitigating the need for extensively labeled data sets.

While FSL is a task-specific approach to obtaining material property data through NER, generative pretrained language models offer a more general, transfer-learning-based alternative. Within chemistry, the generative pretrained transformer (GPT)[6] series of models have not only performed competitively across a variety of classification tasks, including property and yield prediction,[13] but also in metal−organic framework synthesis, where they achieved >0.9 precision, recall, and F1 scores across key parameters for several hundred such structures.[32] In addition, they possess the ability to autonomously design, plan, and conduct experiments, indicating an advanced understanding of complex chemical contexts.[5] The deep contextual comprehension of these models, combined with their demonstrated FSL ability,[6] makes them particularly promising for material data extraction.

In this work, we perform a comprehensive evaluation of multiple possible approaches to mitigate the need for large,

Received: November 11, 2024
Revised: March 5, 2025
Accepted: April 3, 2025
Published: May 13, 2025

https://doi.org/10.1021/acs.jcim.4c01352
J. Chem. Inf. Model. 2025, 65, 4906−4917

annotated data sets in the materials science domain. Our work provides crucial guidance to material scientists and chemists aiming to perform information extraction for targeted use cases. We evaluate the efficacy of three methods for material property extraction: supervised learning, FSL, and large language model (LLM)-based techniques. For the supervised and FSL methods, we focus on techniques to automatically generate text annotations for the training data and leverage only small sets of manually labeled data for evaluating model performance in realistic sentences. We also focus on how LLMs, through data augmentation, can boost supervised learning performance. Our findings reveal that while supervised and FSL methods show comparable results, they are significantly outperformed by LLM-based approaches. Additionally, we conduct both error and data quality analyses to offer qualitative insights into the models' performance across different properties.

Related Work. While a considerable amount of effort has been made to automate the extraction of data sets containing material property measurements, these endeavors rely primarily on a supervised approach, requiring extensive manual labeling. This process, although effective, is resource-intensive. The development of the NLM-Chem corpus, which includes 38,342 chemical mentions of 4867 unique chemicals from 150 full-text articles, demonstrates this challenge.[16] It required double annotation by 10 experts over three rounds, with highly specific annotation guidelines developed over the course of 10 weeks and resolution of all annotation disagreements. This corpus was designed to aid the development of NER for chemicals, showcasing the utility and necessity of such detailed manual annotation in supervised learning models.[16,26]

Similarly, efforts to develop data sets for solid-state inorganic materials,[19] organic chemical reactions,[12] and nanomaterials made of various compounds with different morphologies[14] all involve intensive annotation efforts. In the work by Kononova et al., 834 paragraphs from 740 articles were labeled to identify four different tags related to solid-state synthesis.[19] Likewise, work by Guo involved 280−320 h by 15 domain experts to obtain 800 annotated and refined reactions.[12] Hiszpanski needed to hand-label 99 articles and 27,000 individual sentences to develop an active learning approach to extract chemicals used in nanomaterial synthesis from 35,000 articles.[14]

These aforementioned initiatives are directed toward enhancing the performance of chemistry-focused NER tools for specialized applications, primarily through the creation and utilization of high-quality, labeled data sets and the training of supervised models. While details from these works are provided to give concrete examples, numerous additional studies from various fields of materials science also require extensive annotation efforts to improve supervised approaches for extracting information.[20,27,22] The focus of our work in this paper is to identify potential methods for material property extraction that mitigate or avoid the need for these types of extensive annotation efforts. Therefore, in contrast to the existing work described above, we focus on the evaluation of methods that require limited manual labeling at either the training or inference phases by leveraging a combination of automated labeling methods and low-data requirement learning and inference methods.

Generative LLMs, such as GPT, have recently been adapted and evaluated for NER tasks. For example, Wang evaluated GPT on traditional NER entity type extraction (e.g., person, location, organization) and found that GPT exceeds supervised performance in the low-resource regime.[30] Chen performed meta-function pretraining of LLMs to extract entities from news, social media, biology, and financial domains in the few-shot setting.[7] Additionally, some initial work has been performed by applying LLMs to material science extraction tasks. For example, Ajith leveraged GPT to identify whether a given paragraph contains any material property measurements and subsequently extract all material properties contained in the text.[1]

In comparison with prior efforts, our work is the first to perform a comprehensive analysis of multiple NER approaches in the low-data regime of the material science domain. We additionally perform model development and evaluation on a large set of representative material properties to demonstrate which methods can perform consistently across a broad range of extraction tasks.

Data. Our data comprise sentences extracted from the full text of over 10 million papers from the PubMedCentral open access text mining data set[1] and S2ORC,[21] as described by Panapitiya.[25] From these sentences, we identify those likely to contain 11 different material properties: absorption, boiling point, capacitance, charge, conductivity, density, mass, melting point, redox potential, solubility, and viscosity. The sentences are extracted with the Python Natural Language Toolkit[4] and filtered by requiring each sentence to include the property's name, at least one digit, and specific acceptable units for that property. We then perform automated property annotation using existing information extraction tools on the full set and manual property annotation on a small subsample. The automated annotations are expected to be a noisy and imperfect data set to be used in combination with high-fidelity, gold-standard, manually annotated data. This multifidelity approach allows us to leverage a larger volume of training data while minimizing the required annotation effort. For each property, we aim to annotate three pieces of information: (1) the chemical entity/material name; (2) the property value; and (3) the property unit.

Automated Data Labeling. To generate automatically labeled data, we use two information extraction tools: Grobid-quantities[10] and ChemDataExtractor.[28] The Grobid-quantities tool employs a linear conditional random field (CRF) model to identify expressions of measurements in the text, encompassing both numerical values and their corresponding units. ChemDataExtractor uses a CRF-based entity recognizer with a dictionary-based approach to identify chemical names. With these tools, we obtain a data set of sentences automatically annotated with labels of chemical entities, measurements, and units. To increase our confidence that the tagged information is related to the desired property, we keep only those annotations where there is a single entity, value, and unit extracted from the sentence. However, we anticipate that these automatically labeled data will be noisy and contain inaccuracies. Our intention is to evaluate the ability to use such automated but noisy labels to train NER models without the need for an extensive manual labeling effort. Additionally, this quality filtering step will tend to reduce the complexity of the sentences, as it will remove cases that describe multiple measurements or contextual information. This limitation will likely impact the ability of the NER models trained on these data to perform extraction on such complex sentences. Of the properties, all except redox potential and viscosity are included in the data set derived from automated labeling. These two properties did not have sufficient data to meet the filtering criteria.

Manual Data Labeling. We select a small set of sentences for manual annotation to create a set of high-quality data sets

https://doi.org/10.1021/acs.jcim.4c01352
J. Chem. Inf. Model. 2025, 65, 4906−4917

that are intended for use as an evaluation set, as well as for few-shot support set examples. We place an emphasis on sentences with complex structure (containing, for example, clauses, comparative forms, conjunctions, etc.) and those with more than one set of measurement data to represent realistic but challenging cases for information extraction. This evaluation approach is motivated by the idea of a stress test or challenge set evaluation.[23,2,15] Given a fixed annotation budget, a test set that prefers complex sentences based on observed prior failure modes of similar models[25,8,18] will provide more discriminatory power between models compared with a randomly selected test set. Most extraction approaches will get the "easy" examples correct in the random test set, whereas this approach allows us to probe edge cases and learns the failure modes of the models.

For each property, we select 10 sentences to manually label for a total of 110 sentences. One domain expert annotated all sentences in this data set. While these comprise small test sets for each property, we leveraged our manual annotation efforts to cover a broad range of material properties rather than annotating more examples on a smaller number of properties. Using these targeted test sets, we do observe significant performance differences between different modeling approaches, indicating that these test cases provide sufficient information to judge the relative merits of the tested approaches. In addition to these small sets of manually labeled data, we also leverage an existing large set of manually labeled data for solubility, including 3041 manually annotated sentences.[25]

GPT-Based Synthetic Data. We use GPT-3.5 (model version = "gpt-35-turbo-0301") to generate additional sentences and labels to augment the automatically extracted data sets for absorption, capacitance, charge, and density. We instruct the model to adopt the persona of a chemistry expert, generating sentences similar to those found in scientific publications about a property of interest and the corresponding entity tags in the generated sentences. To guide the generation of these sentences for each property, we supply two chemical compounds sampled from the automatically labeled data set from the target property, along with example sentences and annotations from the corresponding manually labeled data set to demonstrate both the style of the sentences and the format desired in the assistant-level field. Finally, we provide 10 chemical compounds sampled from a list that we generate by aggregating across all properties from the automatically extracted data set, excluding ones from the property of interest. We found that providing example chemical names encourages the model to use more specific chemical entities in the generated sentences rather than more generic terms such as "substance". For each property, we iterate until we can create an augmented data set that contains 50% more sentences than the automatically labeled data. The full prompt template used to generate these data is in the Supporting Information.

METHODS

Material property extraction is typically framed as an NER task, where entities are detected and classified as a predefined type (e.g., material, value, unit). Traditionally, NER is formulated as a sequence labeling task, in which each token in a sequence is assigned to an entity class. The task is formally defined as follows: given an input sequence of tokens $x = \{x_t\}_{t=1}^{T}$, NER aims to assign each token x_i a label $y_i \in Y$. For property measurement extraction, we need at least three entities: material, value, and unit. Tokens that do not correspond to these three pieces of information are assigned an 'O' tag. When using NER in specialized domains, a primary challenge acquires sufficient annotated data for field-specific entity types.[17,9]

Supervised Learning. Supervised learning is the most common approach for extracting material-related information. We train models that take cased SciBERT embeddings[3] and apply a linear layer with dropout, followed by a CRF layer to predict labels at the token level. We select this architecture because it has the best overall performance for the task of labeling chemical, value, and unit entities for solubility across a variety of model architectures with different transformers and pretraining objectives.[25]

A model is trained for each property using the data set acquired with automated data labeling described above in the Automated Data Labeling section. The exception is the solubility model, which is trained on a mixture of automated data and an existing set of manually annotated data. The data set size for each property can be found in the Supporting Information. We use the best hyperparameters, selected based on their F1 scores, in predicting entity types, as described by Panapitiya,[25] to train models that take a maximum sequence length of 128. We partition the data set for each property into train, validation, and test subsets using an 80:10:10 ratio. For optimization, we use the AdamW optimizer with a learning rate of 3×10^{-5}. Stabilization of the validation curve is observed after 12 epochs of training with batch sizes of 32 for training and 8 for validation. While the models are trained on the automatically annotated data, we test each of the property-specific models on their respective manually labeled data set.

Few-Shot Learning. FSL offers a way to identify and classify entities with just a few labeled examples without the need for extensive annotations. During training in N-way, K-shot learning, the model has access to episodes of data, each of which comprise a support set $S_{\text{train}} = \{x_j^{(i)}, y^{(i)}\}_{i=1}^{N\times K}$ containing K examples for each of N classes and a query set $Q_{\text{train}} = \{x_j^{(j)}, y^{(j)}\}_{j=1}^{N\times K'}$ containing K' examples. The model is trained by using information in the support set S_{train} to predict labels of disjoint query set Q_{train} in a supervised manner. By training across a large number of such tasks, the model learns to extract the desired information using only a few examples for a new task specified at inference time.

We apply the NNShot algorithm for FSL, which uses token-level similarity scores for predictions.[31] NNShot first calculates the squared Euclidean distance of the L2-normalized embeddings between a query sample token and every token in the support set. Following this, NNShot assigns the tag from the support set token that is most similar to the corresponding query sample token. During training, the model learns an effective embedding space for performing this similarity-based tagging.

Since entities of type material, value, and unit are needed, we have a three-way classification task for each property. To construct each episode in the training set, we randomly sampled two properties from the pool of nine and included three example sentences for each selected property. This results in a six-way, three-shot setup. For example, a single episode might include three solubility sentences and three capacitance sentences in the support set, with the task to annotate both solubility and capacitance in the query set sentences. By combining multiple properties in each episode, we can introduce more variety into the tasks performed during training and improve the model's ability to adapt to new annotation tasks. The validation set follows the same structure, where each episode also includes two sampled properties. The automatically generated data used to

https://doi.org/10.1021/acs.jcim.4c01352
J. Chem. Inf. Model. 2025, 65, 4906−4917

Table 1. Summary of Model Performances by Tag with Error Variability over Multiple Properties and 10 Inference Runs[a]

tag	model	precision	recall	F1
Material	GPT-4o: In-context	**0.731 ± 0.183**	0.751 ± 0.157	**0.739 ± 0.166**
	GPT-4o: Zero-shot	0.686 ± 0.222	0.736 ± 0.198	0.704 ± 0.120
	GPT-3.5: In-context	0.698 ± 0.218	0.735 ± 0.144	0.711 ± 0.179
	GPT-3.5: Zero-shot	0.628 ± 0.221	0.670 ± 0.239	0.639 ± 0.213
	LLaMa-3.3: In-context	0.712 ± 0.165	**0.767 ± 0.146**	0.731 ± 0.140
	LLaMa-3.3: Zero-shot	0.643 ± 0.189	0.763 ± 0.177	0.687 ± 0.168
	Supervised	0.455 ± 0.320	0.304 ± 0.202	0.352 ± 0.228
	FSL: NNShot	0.310 ± 0.220	0.349 ± 0.200	0.318 ± 0.204
Units	GPT-4o: In-context	**0.942 ± 0.062**	0.925 ± 0.096	**0.932 ± 0.074**
	GPT-4o: Zero-shot	0.921 ± 0.100	0.903 ± 0.109	0.909 ± 0.097
	GPT-3.5: In-context	0.929 ± 0.091	**0.938 ± 0.074**	0.931 ± 0.069
	GPT-3.5: Zero-shot	0.836 ± 0.124	0.850 ± 0.119	0.840 ± 0.112
	LLaMa-3.3: In-context	0.890 ± 0.144	0.924 ± 0.127	0.899 ± 0.123
	LLaMa-3.3: Zero-shot	0.845 ± 0.152	0.911 ± 0.138	0.871 ± 0.134
	Supervised	0.726 ± 0.278	0.808 ± 0.301	0.757 ± 0.276
	FSL: NNShot	0.702 ± 0.157	0.810 ± 0.150	0.749 ± 0.147
Value	GPT-4o: In-context	**0.878 ± 0.093**	**0.871 ± 0.102**	**0.873 ± 0.091**
	GPT-4o: Zero-shot	0.711 ± 0.151	0.741 ± 0.143	0.723 ± 0.143
	GPT-3.5: In-context	0.725 ± 0.150	0.764 ± 0.146	0.742 ± 0.143
	GPT-3.5: Zero-shot	0.625 ± 0.156	0.670 ± 0.140	0.644 ± 0.144
	LLaMa-3.3: In-context	0.657 ± 0.155	0.738 ± 0.152	0.689 ± 0.144
	LLaMa-3.3: Zero-shot	0.518 ± 0.189	0.637 ± 0.178	0.564 ± 0.176
	Supervised	0.559 ± 0.300	0.630 ± 0.307	0.559 ± 0.270
	FSL: NNShot	0.561 ± 0.141	0.709 ± 0.160	0.622 ± 0.141

[a]Best results are shown in bold.

train and validate the FSL model are described in the Automated Data Labeling section.

To evaluate the model's ability to handle complex sentences across a variety of properties, we use the manually annotated data as the test set. For each property, we make episodic data for one property at a time to test the model by sampling three disjoint sets of three sentences to include in $S_{support}$; for each set, we take the remaining complementary set of seven sentences to form S_{query}. This results in three episodes of data. We repeated this five times to generate 15 episodes of data for each property. During inference, the model only has access to information in $S_{support}$ to predict labels of the tokens in S_{query}.

We train at most 9600 episodes in batches of 16 and implement an early stopping mechanism that terminates training if no improvement in validation F1 is observed after 25 consecutive iterations. In this model, we use cased SciBERT as the pretrained encoder.[3] The weights are optimized with AdamW with a learning rate of 1×10^{-4} and a cross-entropy loss function.

Large Language Models. We formulate NER as a generation task and assess the ability of several different LLMs to extract tags associated with properties under zero-shot and in-context learning settings. We compare two LLMs from the GPT series developed by OpenAI, GPT-3.5 (model version = "gpt-35-turbo-0301"), and GPT-4o[2] (model version = "gpt-4o-2024−05−13") and one model from the LLaMa series, Llama-3.3−70B-Instruct[3], developed by Meta. For all three LLMs, we use a structured template inspired by Guo.[13] In both the zero-shot and in-context settings, the model is provided with a prompt containing three distinct components: a general template, a task-specific NER template, and the question. The general template instructs the model to adopt the persona of an expert chemist, extracting measurements about a specific property. We then request that the model extract entities from a supplied sentence, following a strict format. Lastly, the model is given in the sentence. In the in-context learning setting, the model is first shown a varying set of three samples of input/output pairs of sentence entities from the manually labeled data set before it receives the specific sentence to be processed. In contrast to the supervised and few-shot methods, the LLMs do not provide token-level tags but instead produce a list of the chemicals, values, and units present in the input sentence. The prompt template used both settings, as shown in the Supporting Information.

Evaluation. We evaluate each model based on precision and recall of the extracted entities. To calculate these metrics, we implement strict word-level matching (as opposed to token-level) due to the text output of GPT not being at the token level. This also ensures that each model's performance reflects the accuracy of identifying complete entities, which is critical in developing high-quality databases for material development. For the supervised and FSL methods, we decode the output and combine consecutive subword tokens with the same predicted label to get the words. To ensure an accurate comparison between predicted entities and those labeled manually while preserving the original interpretation of the words, white spaces are removed from both sets of words. Before comparing material and unit entities, the words are lowercased; for value entities, commas are removed. Precision is then calculated as the percentage of extracted entities that are found in the ground-truth annotated entities, while recall measures the percentage of ground-truth entities that are present in the extractions. For each model, we repeat the extraction process 10 times and compute

https://doi.org/10.1021/acs.jcim.4c01352
J. Chem. Inf. Model. 2025, 65, 4906−4917

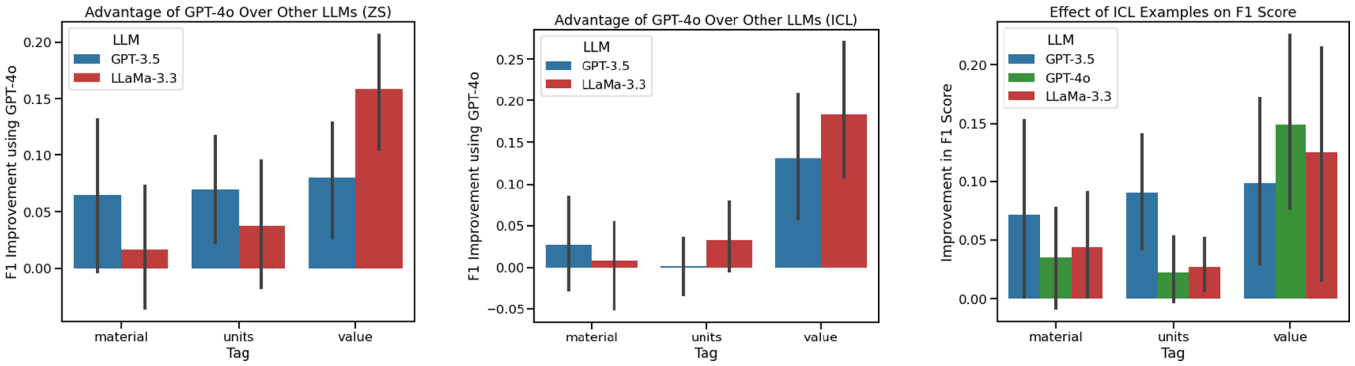


Figure 1. Performance comparisons of different LLM models and prompting methods for the three tag types averaged over all properties. (Left) Average improvement in the F1 score when using GPT-4o rather than GPT-3.5 and LLaMa-3.3 in the zero-shot context. (Middle) Average improvement in the F1 score when using GPT-4o rather than GPT-3.5 and LLaMa-3.3 in the in-context learning setting. (Right) Average improvement in F1 when using in-context learning examples compared with the zero-shot setting. The error bars represent the 95% confidence interval of the distribution across properties.

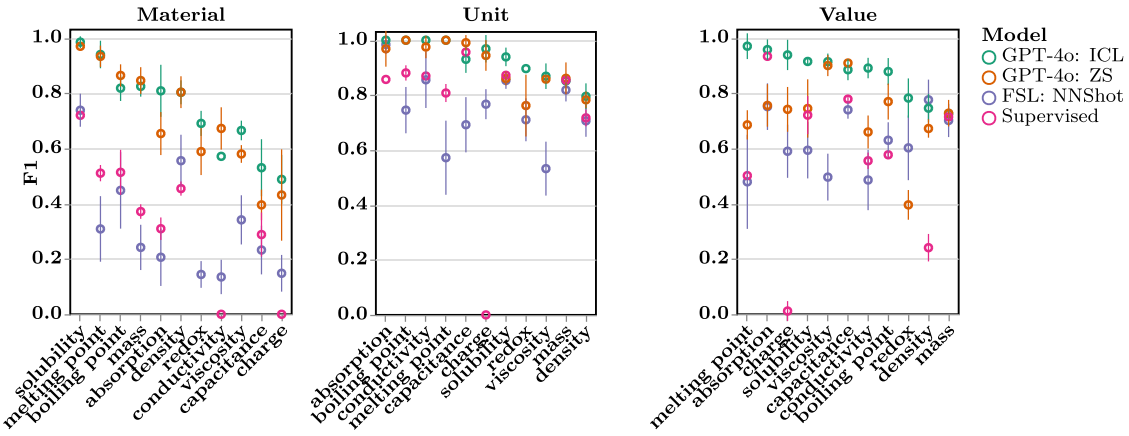


Figure 2. From left to right, comparison of model performance for material, unit, and value entity extraction across 11 properties on annotated data, measured by the F1 score. Nb.: *x*-axis order, sorted by the descending maximum F1 score.

both the mean and standard deviation of the precision, recall, and F1 metrics.

RESULTS

The evaluation metrics for all models without GPT-based data augmentation for the three tag types averaged across all 11 properties are listed in Table 1. First, we compare the performance of the different LLM models and prompting approaches. We find that GPT-4o with ICL has the best overall performance in terms of F1 score. We visualize the advantage of this LLM versus those of the other LLMs, as shown in Figure 1. We find that in the zero-shot setting, GPT-4o has a consistent advantage over GPT-3.5 for all tags, while it has a large advantage over LLaMa-3.3 in terms of value extraction but a smaller advantage for material and unit extraction. When in-context learning examples are provided, GPT-3.5 and LLaMa-3.3 are almost able to match GPT-4o for material and unit extraction; however, GPT-4o has a significant advantage over both for value extraction. Finally, we find that all three LLM models benefit from using in-context examples in the prompt, especially for the value tag extraction.

Both the supervised and FSL approaches lag significantly behind the LLMs for all tag types, with the material tag being a particular weakness. Supervised training outperforms the FSL approach for the material and unit tags, while the FSL performance exceeds that of supervised learning for the value tag.

We next examined the performance of these models at the property level, as shown in Figure 2, to explore potential disparities in model performance masked by aggregated metrics. We focus on comparing the best-performing LLM, GPT-4o, with the other two approaches. We observe a property-level dependence in model performance that is most pronounced for material mentions and least pronounced for unit mentions. This suggests that model performance on the material tag is highly sensitive to property, whereas the unit tag is less so. Whether in the zero-shot or in-context learning setting, GPT-4o still generally outperforms the supervised and FSL approaches across the three entity types. The in-context examples provided to GPT provide a boost to performance over the zero-shot setting across the majority of properties, especially for the value tag and for the material tag on more challenging properties.

While both the supervised and FSL approaches lag behind the LLMs in performance, it is interesting to note that the supervised methods typically outperform the FSL models with median F1 improvements of 14.6, 2.2, and 4.7% for material, value, and unit tags. When making predictions, the supervised methods have only been trained on noisy automated data annotations. In contrast, while the FSL methods are trained on the same noisy data set, they have access to the high-quality manual annotations in the support set when making predictions on new sentences.

https://doi.org/10.1021/acs.jcim.4c01352
J. Chem. Inf. Model. 2025, 65, 4906−4917

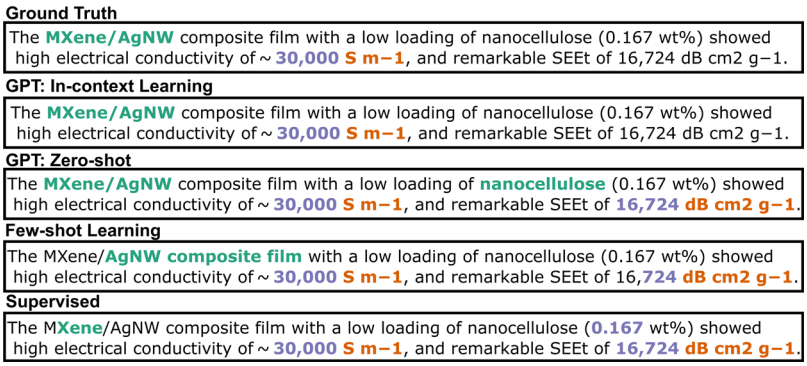


Figure 3. Example of a sentence containing a conductivity measurement with material, value, and unit tags extracted by GPT-based, FSL, and supervised approaches.

This shows that the volume of training data available to supervised methods still outweighs the benefit of the small quantity of high-quality annotated data available only to the FSL model.

We also highlight that one of the best-performing supervised models is for the solubility data, which has a mix of both noisy automated annotations and high-quality manual annotations in its training data. It achieves the best performance of any supervised model on the material task by a large margin and is the third best on the unit and value tasks. The comparison of solubility with the other properties provides an indication of how the supervised model's performance is affected by the use of noisy, automated data labels versus high-quality manual annotations. Interestingly, we find that GPT-4o with in-context learning still significantly outperforms the supervised model for solubility, despite the use of direct supervision with high-quality training examples. This indicates that the use of extensive manual labeling efforts may not be worthwhile, relative to workflows that leverage LLM-based extraction.

We find a strong correlation across models between the property-level performance for the material tag, indicating that the different models find the same properties to be difficult, whereas there is less correlation for the unit and value tags, indicating that the different models rank the properties differently in terms of difficulty. The average Spearman correlation between all pairs of models for the material tag is 79%, while the average correlations for the unit and value tags are only 44 and 22%, respectively.

Qualitative Error Analysis. Given the observed performance variability in the material tag, we examine the predictions each model makes on this tag to understand the underlying causes of errors and identify patterns in misclassification. This analysis aims to pinpoint specific characteristics or contexts within the data that challenge the models, thereby providing insights into how the model performance can be further optimized for more accurate and reliable material tagging. We show one example of the predicted tags for several models in Figure 3.

We group the factors affecting model performance into two main groups: those impacting recall and those impacting precision. On the recall front, common issues hindering material extraction often stem from the presence of multiple chemicals due to multiple property measurements (e.g., solubility measurements for many compounds) or the mention of a generic material such as "nanocomposite films" or a "substance".

In examining precision-related issues, the models face challenges in accurately identifying material entities due to multiple factors. These include the complexity of chemical nomenclature, the presence of multiple chemical mentions, and a preference for compound-like terms. For instance, with regard to chemical nomenclature, models often struggle when a chemical entity is long, contains special characters, or includes a common term (e.g., 5-sulfosycilic acid dihydrate, Mo_2C@NPC/NPRGO, Si wafer). In these cases, a substring of the chemical is selected (e.g., 5-sulfoscylic, NPC, and Si).

When multiple chemical mentions occur, with some providing contextual information for a material's measurement, models often include the contextual information. For example, in a statement like "NiO, CoO_3, and $NiCo_2O_4$ were synthesized, with $NiCo_2O_4$ yielding the best specific capacitance of 684 F g-1", the models will often extract NiO and CoO_3 in addition to $NiCo_2O_4$. This tendency leads to a decrease in the measured precision due to the strict framework employed that limits the models' ability to include contextual information. While this restriction enhances focus on essential information, it also highlights a challenge in extracting complex scientific data, where context can provide additional information.

Additionally, the models exhibit a preference for more specific chemical mentions or terms that are chemically similar, even when these terms do not correspond to the actual material related to a property measurement or are not true chemicals. This tendency is closely related to the presence of contextual information and affects recall. As an example, in a sentence about the melting point of polyethylene obtained using CO_1−CO_5 precatalysts, the models may extract CO_1 and/or CO_5, or no material. Similarly, when a sentence is about weighing an amount of a "sample" into an alumina crucible, the models tend to extract alumina. Furthermore, when terms that resemble chemical notation appear, such as in a sentence containing a capacitance measurement in the context of an equation characterizing a dynamic potential with terms such as I_{app}, I_{leak}, or one about the current density, j_0, of a material, the models identify terms such as I_{leak} and j_0 as materials.

These patterns are present in material entity extraction by supervised, FSL, and LLM-based approaches. However, we note that in the case of the LLMs, they are less prevalent and typically further ameliorated in the in-context learning setting, where some examples are provided. The LLMs more adeptly identify chemicals even when they are general or complex and avoid terms that appear like chemicals.

It is also interesting to note that despite being trained on data containing at most one material, value, and unit tag per sentence, we observe that supervised models do attempt to extract multiple measurements from the more complex sentences in the

https://doi.org/10.1021/acs.jcim.4c01352
J. Chem. Inf. Model. 2025, 65, 4906−4917

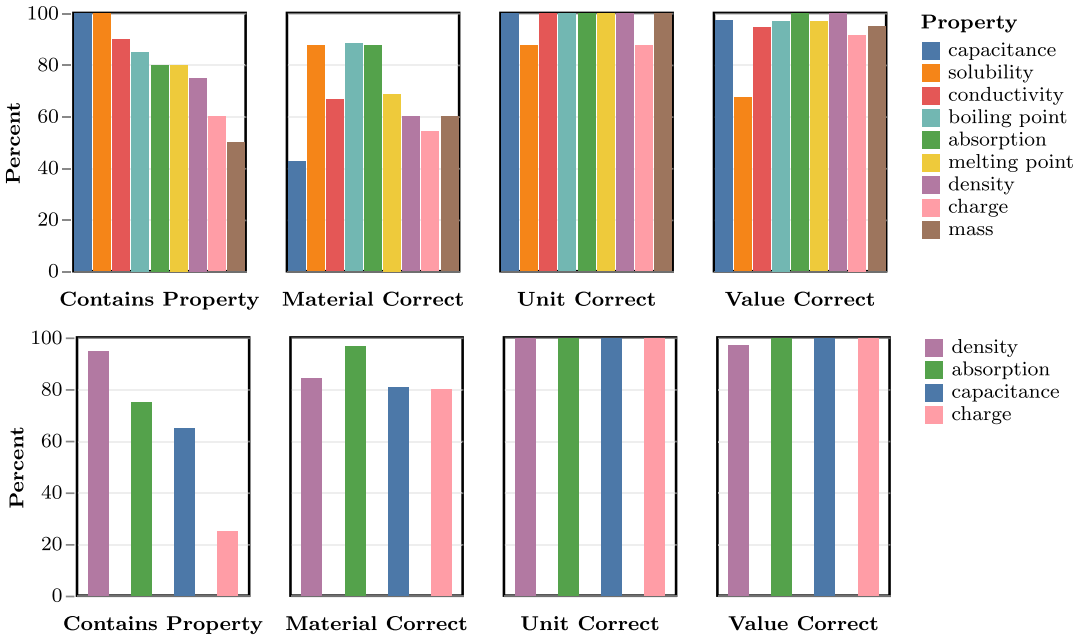


Figure 4. Data quality in sampled sentences. Plots depict measurement presence and subsequent correctness across three tag types for (top) automatically labeled and (bottom) GPT-generated data sets.

test sets. The supervised approach extracts more than one material 31% of the time and more than one value 54% of the time. However, the accuracy of these multiple extractions remains low relative to the performance observed for the LLM-based methods.

Impact of Materials' Data Quality. Since the models exhibit property-level dependence and the quality of data used to train models plays a significant role in performance, we conduct an evaluation of data quality. For each property in our automatically annotated and GPT-generated data sets, we sample 20 annotations and employ a Boolean method to tabulate whether each actually contains a measurement of the target property. If so, we subsequently use a ternary system to evaluate the correctness of the annotations of each tag type, where 0 indicates an incorrect annotation, 1 signifies a correct annotation, and 0.5 represents a partially correct annotation.

The result of this analysis is shown in Figure 4. The top series of plots shows the percentage of the sampled automatically annotated sentences with a desired property measurement and the subsequent set containing a correctly identified tag of each type. Overall, the automatically labeled sentences for capacitance and solubility are of high quality, with a large percentage of sentences containing associated property measurements (i.e., a unit and value). In comparison, charge and mass are of lower quality, with a smaller percentage of sentences containing a measurement. We find that across all properties, nearly all sentences that contain a property measurement have correctly identified the value and units. However, the task of identifying the correct material is not only more challenging, as reflected by a lower percentage, but also exhibits greater variability. This annotation quality issue likely impacts the observed reduced performance in extracting the correct material information.

The bottom series of plots shows the same information for the sampled GPT-generated sentences. While the percentage of sentences containing an associated property measurement has a wider spread, a similar pattern exists; the material tag is more challenging to create and identify correctly, whereas for the sentences that contain a property, GPT accurately generates and identifies unit and value tags. However, incorporating GPT-generated sentences to augment training data does improve model performance by providing more property measurements, although the inconsistency in accurately identifying materials may explain the lower-than-expected enhancement (see the Role of Data Augmentation in Supervised Learning section below).

When we observe the relationship between model performance on the material tag and the quality of the material tag annotations in the automated data, we find that all extraction methods exhibit a positive correlation, with a correlation of around 0.40 for GPT-3.5, around 0.52 for GPT-4o, around 0.68 for LLaMa-3.3, and around 0.55 for FSL and supervised learning. Given that the LLM models do not use the automated data, the correlation for these models likely reflects the fact that the manual annotations fail on the sentences that represent a harder extraction task, and therefore, the automatic tagging methods and the LLM models both struggle with the same properties. The correlations for the FSL and supervised methods may reflect how the low quality of the annotations inhibits the ability of these methods to learn correct tagging methods.

Given the observed correlation levels between data quality and model performance, there are likely additional influencing factors that drive model performance variation across properties. A significant one could be the inherent complexity of the material tag itself. As one example, conductivity has reasonable data quality but a supervised F1 score of 0 on the material tagging task. When we look at the manually annotated materials for this property, we find that all of them contain common terms or special characters or reference a class of materials that are typically challenging for extraction, as discussed above. For the charge, which also has an F1 of 0 with a supervised model, such challenging features are present in 60% of sentences. Meanwhile, charge only contains 215 sentences in its data set, whereas conductivity has a more substantial representation with 1283 sentences. This suggests that the volume of sentences in the supervised data set, which varies considerably across properties—as few as 215 sentences for charge and as many as 5158

https://doi.org/10.1021/acs.jcim.4c01352
J. Chem. Inf. Model. 2025, 65, 4906−4917

sentences for mass—combined with the quality of these sentences (Figure 4), likely affect the model's performance.

Another factor for the supervised models might be whether the test sentences contain material names that were observed in the training data for a given property. We find that the proportion of the test material names that are included in the training data ranges from 8.3 to 42% across the different properties, with an average of 22%. However, we found only a weak Pearson's correlation of 0.239 between this overlap percentage and the material extraction F1 score of that property, indicating that this is not a strong driver of the supervised model's performance. Charge has a low overlap of materials between the training and test data (8.3%), while conductivity has a typical overlap (22.2%).

The FSL models for charge and conductivity also have poor performance; however, the F1 scores for the material tag do exceed 0 despite having access to the same charge and conductivity data as the supervised model. Its ability to improve on the supervised method, despite the complexity and limited quantity of the data, could be attributed to the episodic nature of the model architecture, enabling it to learn generalizable patterns and to be less susceptible to the issues of data sparsity and quality that seem to significantly impact the supervised model. It also likely benefits from the ability to observe the high-quality, manually annotated example sentences in the support set during inference, which the supervised model lacks during both training and inference.

Impact of Units and Value Complexity. Next, we analyze the characteristics of the unit and value tags to understand how they contribute to the differences observed when extracting these mentions from text on a property-level basis. Considering the high quality of these tag labels, we next evaluated the complexity of the tags for each property. For units, we take the ratio of unique units after white space removal to the number of sentences for each property in each data set. As shown in Figure 5, the unit complexity in the automatically labeled data is

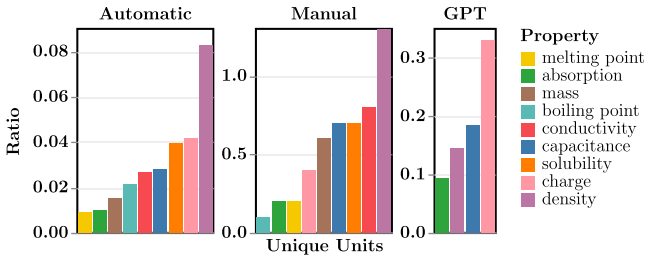


Figure 5. Unit complexity by property measured by the ratio of unique units to the total number of sentences (note that *y*-axis scales are different).

significantly lower than the manually labeled data. This difference can be attributed to the automatic data being derived from ChemDataExtractor and Grobid, coupled with an additional filtering step to a set of allowed units. The GPT-generated data exhibit unit complexity that is on the same order of magnitude as that of the manually labeled data and introduces additional units (Figure 7).

In our analysis of the value tag (Figure 6), we aim to identify the different numerical formats used to express the value. We obtain a set of unique value representations for each data set by replacing consecutive digits with placeholders, removing commas and spaces for consistency (e.g., '18,129.24' becomes 'D.D'). This approach ensures a consistent format for numerical

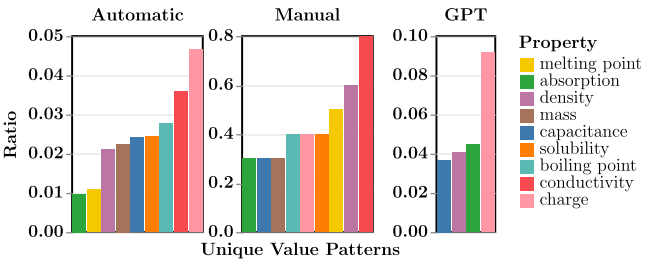


Figure 6. Value complexity by property measured by the ratio of unique value patterns to the total number of sentences (note that the *y*-axis scales are different).

data, facilitating easier comparison and analysis of distinct value patterns. As shown in Figure 6, similar to the trend for the unit tag, we find that the ratio of unique values to sentences is lowest for the automatically labeled data set and highest for the manually annotated one, with the GPT-generated data set falling between the two. We also find that sentences generated by GPT introduce new value patterns into the data that are not present in the automatically labeled data (Figure 7).

To explore the impact that the unit and value complexity have on the model performance, we compute the Spearman correlation between the property-level F1 score for each model (Figure 2) and the unit or value complexity for the same property for both the automatic and the manually annotated data (Figures 5 and 6). The results are listed in Figure 8. We find that performance is typically negatively correlated with complexity, indicating that the increased variability in the units and the value formats does, in fact, increase the difficulty of the extraction task.

Overall, the automatic-value complexity shows less evidence of negative correlations, and in fact, the LLaMa models have positive correlations. These positive correlations are driven by the two properties with the lowest automated value complexity, absorption, and melting point, which are well predicted by GPT but poorly predicted by LLaMa. In addition to having low automated value complexity, the manually labeled data for these two properties have a number of values being expressed as a range (e.g., "a melting point of 132−135 C") or incorporating notation such as "∼" to indicate approximation (e.g., "a melting point of ∼115 C"). The GPT models extract the full range as the value ("132−135") and include the surrounding notation ("∼115"), which matches the ground-truth annotation, while LLaMa tends to extract two separate values (["132", "135"]) and omit contextual notations ("115"), which is penalized.

Role of Data Augmentation in Supervised Learning. We target four mid- to low-performing properties—density, absorption, capacitance, and charge—and train supervised models on two additional data sets: one augmented with GPT-generated data and the other exclusively composed of this data. This approach aims to improve model performance by leveraging swiftly generated GPT data, offering an efficient way to enrich and diversify the automatic data sets, especially for properties where the data are scarce or noisy. Training on GPT data alone enables the assessment of the synthetic data on model performance. We evaluate these models on both the test split of automatically labeled data generated during model training and the high-quality, manually annotated data to assess the disparity between standard test conditions and realistic, challenging cases for information extraction.

Overall, our results shown in Figure 9 incorporate evaluations on the test split of automatically labeled data, slightly offset to

https://doi.org/10.1021/acs.jcim.4c01352
J. Chem. Inf. Model. 2025, 65, 4906−4917

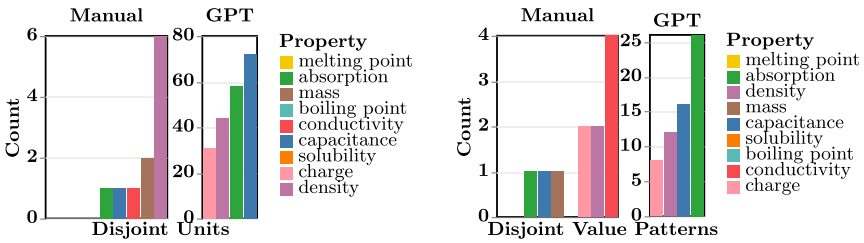


Figure 7. Number of units and value patterns observed in the manually annotated and GPT-generated data that are disjoint (absent) from the automatic data.

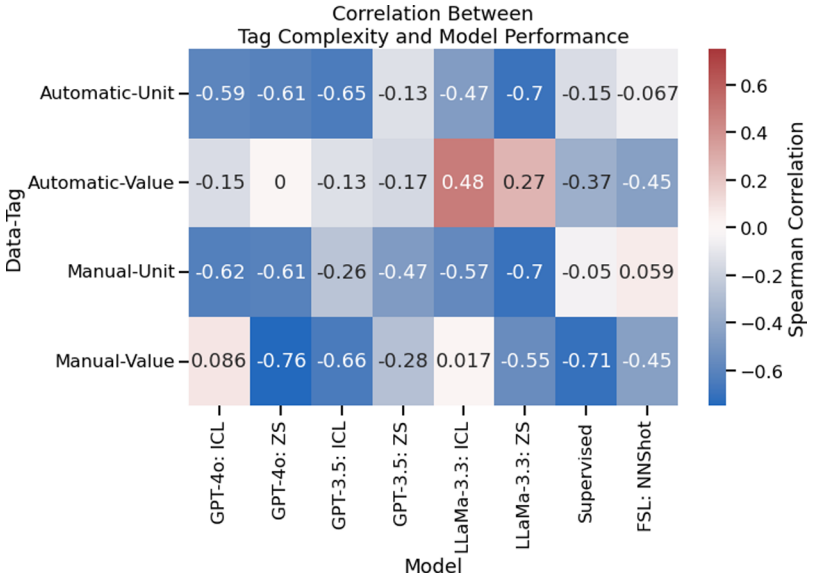


Figure 8. Spearman correlation between the property-level model performance and the complexity of the unit and value tags for each property in the automatic and manually labeled data.

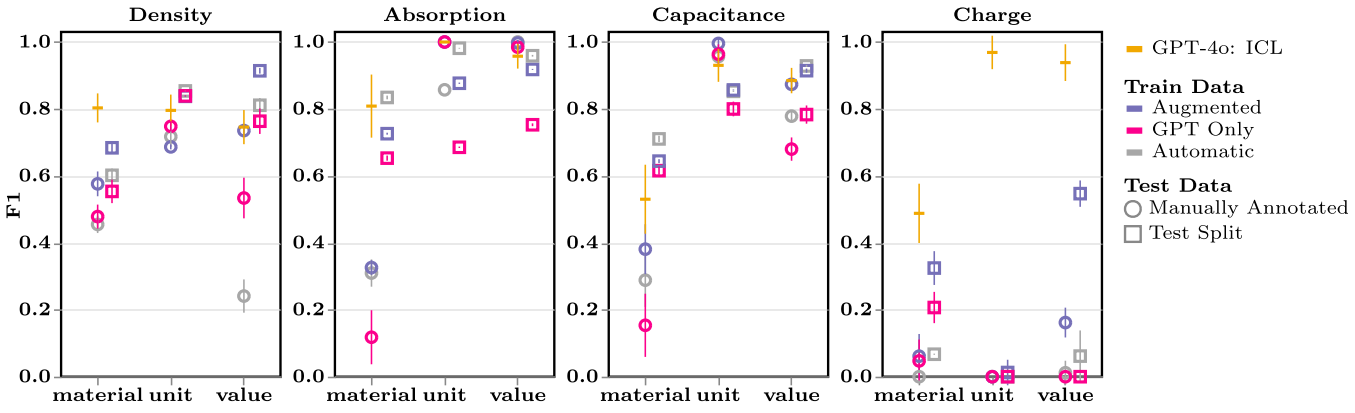


Figure 9. Performance comparison of the supervised model trained on automatic, augmented, and GPT-only data sets.

the right for clarity, alongside the overall best-performing GPT-4o model with in-context learning. This figure indicates that while models trained on augmented data sets typically outperform those trained solely on the automatically labeled or GPT-generated data, they typically do not reach the efficacy levels of the LLM-based extraction methods. For the material and value tags, models leveraging augmented data sets achieve the highest F1 scores. This can be attributed to the higher quality of the GPT-generated data for material entities (Figure 4), which provide not only more accurate but also additional examples of materials. Despite the benefits of data augmentation for this tag, the model trained solely on augmented data does not match the performance of models using a mix of real and augmented data, which could be due to a combination of volume, diversity, and complexity factors. After augmentation, the supervised model notably improved in extracting value entities for charge. This could be due to the high quality of value entities generated by GPT (Figure 4), and as indicated in Figure 7, additional value representations are not present in the automatically labeled data set. For density, there is also a marked improvement in the supervised model's performance, which indicates that augmentation introduces more varied examples, better reflecting the higher complexity of the value tag in the manually labeled data. Interestingly, for the unit tag, models

https://doi.org/10.1021/acs.jcim.4c01352
J. Chem. Inf. Model. 2025, 65, 4906−4917

trained exclusively on the GPT-generated sentences typically surpass those trained on augmented data sets. This can be linked in part to the high quality of the GPT-generated data, which includes a broader variety of units as captured in Figure 7. This observation, especially when taking charge into account, suggests that after a threshold of training data has been met, data quality and accuracy for the value tag are key.

Models trained on purely synthetic data appear promising under test conditions but underperform those with complex, manually annotated data. This performance drop may stem from the limited size of the GPT-generated data, which is only half the size of the original, limiting the model's learning ability despite stable validation. This also indicates that the unique complexities and nuances present in the manually labeled data are not entirely replicated in the GPT-generated data set, suggesting the need to better align synthetic sentences with scientific literature.

Other Considerations. In addition to model extraction accuracy, there are other considerations, which could impact the selection of which method to apply for a given use case. The use of the GPT models through the API service requires a fee to pay for usage on a per-token basis. The prompts developed for this paper required approximately 286 input tokens and 36 output tokens per query sentence on average, resulting in a cost of approximately $1.08 per 1000 sentences using GPT-4o pricing as of January 2025. If this level of cost becomes an issue for processing massive paper databases, it may be more economical to explore smaller models, which can be run locally. Additionally, the use of open-source models, such as the LLaMa series, has significant benefits over proprietary models like the GPT series in terms of reproducibility and accessibility as the user can be confident that the models will not be affected by changes to the model that are out of the user's control. Given that LLaMa-3.3 performs competitively with GPT-4o on the material and unit extraction tasks, efforts to develop pipelines that improve the value extraction performance for this model would be fruitful for developing self-deployed and reproducible workflows for material property extraction tasks.

CONCLUSIONS

We present a comparative study of traditional supervised, FSL-based, and contemporary LLM-driven approaches to material property NER under the scenario of limited amounts of manually annotated data. We evaluate their effectiveness in retrieving information from scientific texts with the aim of harnessing this information to construct databases that can enable data-centric approaches for materials research. Based on our results, we identify LLM-based approaches as the most effective across a range of material properties and tag types, especially when a few high-quality, manually annotated examples are provided, with GPT-4o being the best-performing LLM evaluated. We also demonstrated that GPT can be used to augment noisy data sets through the introduction of both additional and more varied examples of entities for supervised model training with improved overall performance. However, the direct use of LLMs in NER typically outperforms this data augmentation approach. These results provide key insights for scientists aiming to perform targeted material property extraction tasks from scientific literature with limited available annotation capabilities. The results can also inform information extraction in other scientific and technical domains, where measurement and numerical information is contained within natural language documents.

Additionally, our results identify several key areas for future research efforts that are needed to provide more robust performance of these extraction methods in the low-data regime. In particular, we find the extraction of the chemical entities or material names to be significantly more challenging than the extraction of the units or values of the measurement. This pattern held across all of the methods that were evaluated. We identified that this challenge may be partly due to the inaccuracies of the material tag in the automatically generated training data, which motivates the development of improved methods for generating artificial data examples or improved methods for mitigating label noise within the training data.

Looking ahead, the minimal annotations needed to achieve the performance of LLM-based methods indicate that efforts should prioritize refining these approaches for greater effectiveness in extracting material property measurements. This need for enhanced model training is indicated particularly by the variability of using GPT in identifying material entities. To address this challenge, low-rank adaptation could be employed to combine the already strong performance of LLM models with existing high-quality data sets that contain domain-specific annotations for materials, values, and units. Low-rank adaptation, through parameter selective fine-tuning, offers a focused, efficient way to improve the LLM's ability to interpret and extract property measurements, tailoring the LLM to the nuanced requirements of materials science information extraction tasks.

Additionally, the models developed in this paper would benefit from the development of human-in-the-loop pipelines, through which human experts could evaluate and correct the output of the models. Given that the models do not yet achieve sufficient accuracy to be employed in a fully automated fashion, they can be used in a workflow that helps human experts more rapidly search for and extract relevant information from the literature. This process would not only ensure data quality and accuracy but also provide validated training examples to improve the models.

ASSOCIATED CONTENT

Data Availability Statement

The data used in this work was sourced from the publicly available PubMedCentral open access text mining data set[4] and S2ORC.[21] The scripts for processing the data as done in this paper can be found here: https://github.com/pnnl/mat-prop-nlp-data.

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jcim.4c01352.

Full text of queries used for LLM prompting (Listings S1 and S2) and number of sentences included within the automatically labeled data sets for each property (Table S1) (PDF)

AUTHOR INFORMATION

Corresponding Authors

Jessica Kong − *Pacific Northwest National Laboratory, Richland, Washington 99354, United States;* orcid.org/0000-0003-3009-6288; Email: jessica.kong@databricks.com

Emily Saldanha − *Pacific Northwest National Laboratory, Richland, Washington 99354, United States;* orcid.org/0000-0001-8621-674X; Email: emily.saldanha@pnnl.gov

https://doi.org/10.1021/acs.jcim.4c01352
J. Chem. Inf. Model. 2025, 65, 4906−4917

Author

Gihan Panapitiya − *Pacific Northwest National Laboratory, Richland, Washington 99354, United States;* orcid.org/0000-0002-3310-7600

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jcim.4c01352

Author Contributions

‡Work completed while at PNNL.

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

This work was supported by the Energy Storage Materials Initiative (ESMI), which is a Laboratory Directed Research and Development Project at Pacific Northwest National Laboratory (PNNL). PNNL is a multiprogram national laboratory operated for the U.S. Department of Energy (DOE) by Battelle Memorial Institute under Contract no. DE- AC05- 76RL01830.

■ ADDITIONAL NOTES

[1]https://www.ncbi.nlm.nih.gov/pmc/tools/openftlist/.
[2]https://openai.com/index/hello-gpt-4o/.
[3]https://www.llama.com/.
[4]https://www.ncbi.nlm.nih.gov/pmc/tools/openftlist/.

■ REFERENCES

(1) Ajith, A. *MatPropXtractor: Generate to Extract*. In: 2023.

(2) Aspillaga, C.; Carvallo, A.; Araujo, V. Stress Test Evaluation of Transformer-based Models in Natural Language Understanding Tasks. eng. In: *Proceedings of the Twelfth Language Resources and Evaluation Conference*. Ed. by Calzolari, N.; European Language Resources Association: Marseille, France, 2020; 1882−1894. https://aclanthology.org/2020.lrec-1.232/.

(3) Beltagy, I.; Lo, K.; Cohan, A. SciBERT: A Pretrained Language Model for Scientific Text. *arXiv: 1903.10676 [cs]*. 2019. . http://arxiv.org/abs/1903.10676 (visited on 12/19/2023). preprint.

(4) Bird, S.; Klein, E.; Loper, E. *Natural language processing with Python: analyzing text with the natural language toolkit*; O'Reilly Media, Inc., 2009.

(5) Boiko, D. A.; MacKnight, R.; Gomes, G. Emergent Autonomous Scientific Research Capabilities of Large Language Models. *arXiv: 2304.05332 [physics]*. 2023. . http://arxiv.org/abs/2304.05332 (visited on 01/03/2024). preprint.

(6) Brown, T. Language Models Are Few-Shot Learners. In: *Advances in Neural Information Processing Systems*; Curran Associates, Inc., 2020, 33; 1877−1901. https://papers.nips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf (visited on 01/06/2024).

(7) Chen, J. Learning In-context Learning for Named Entity Recognition. In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Vol. 1: Long Papers)*. Ed. by Rogers, A.; Boyd-Graber, J.; Okazaki, N.; Association for Computational Linguistics: Toronto, Canada, 2023; 13661−13675. https://aclanthology.org/2023.acl-long.764.

(8) Cheung, J. POLYIE: A Dataset of Information Extraction from Polymer Material Scientific Literature. In: *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Vol. 1: Long Papers)*. Ed. by Duh, K.; Gomez, H.; Bethard, S.; Association for Computational Linguistics: Mexico City, Mexico, 2024; 2370−2385. https://aclanthology.org/2024.naacl-long.131/.

(9) Ding, N. Few-NERD: A Few-shot Named Entity Recognition Dataset. In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Vol. 1: Long Papers). ACL-IJCNLP 2021*. Ed. by Zong, C.; Association for Computational Linguistics, 2021; 3198−3213. https://aclanthology.org/2021.acl-long.248 (visited on 12/04/2023).

(10) Foppiano, L. *Lfoppiano/Grobid-Quantities*. 2023. https://github.com/lfoppiano/grobid-quantities (visited on 12/14/2023).

(11) Foppiano, L.; Dieb, S.; Suzukia, A.; de Castrob, P. B.; Iwasakib, S.; Uzuki, A.; Echevarria, M. G. E.; Meng, Y.; Terashima, K.; Romary, L.; Takano, Y.; Ishii, M. SuperMat: Construction of a Linked Annotated Dataset from Superconductors-Related Publications. *Sci. Technol. Adv. Mater.: Methods* **2021**, *1* (1), 34−44.

(12) Guo, J.; Santiago Ibanez-Lopez, A.; Gao, H.; Quach, V.; Coley, C. W.; Jensen, K. F.; Barzilay, R. Automated Chemical Reaction Extraction from Scientific Literature. *J. Chem. Inf. Model.* **2022**, *62* (9), 2035−2045.

(13) Guo, T.et al.What Can Large Language Models Do in Chemistry? A Comprehensive Benchmark on Eight Tasks. *arXiv: 2305.18365 [cs]*. 2023. http://arxiv.org/abs/2305.18365 (visited on 12/14/2023). preprint.

(14) Hiszpanski, A. M.; Gallagher, B.; Chellappan, K.; Li, P.; Liu, S.; Kim, H.; Han, J.; Kailkhura, B.; Buttler, D. J.; Han, T. Y.-J. Nanomaterial Synthesis Insights from Machine Learning of Scientific Articles by Extracting, Structuring, and Visualizing Knowledge. *J. Chem. Inf. Model.* **2020**, *60* (6), 2876−2887.

(15) Isabelle, P.; Cherry, C.; Foster, G. A Challenge Set Approach to Evaluating Machine Translation. In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Ed. by Palmer, M.; Hwa, R.; Riedel, S.; Association for Computational Linguistics: Copenhagen, Denmark, 2017; 2486−2496. https://aclanthology.org/D17-1263/.

(16) Islamaj, R.; Leaman, R.; Kim, S.; et al. NLM-Chem, a New Resource for Chemical Entity Recognition in PubMed Full Text Literature. *Sci. Data* **2021**, *8* (1), 91.

(17) Katz, U. et al.NERetrieve: Dataset for Next Generation Named Entity Recognition and Retrieval. *arXiv: 2310.14282 [cs]*. 2023. . http://arxiv.org/abs/2310.14282 (visited on 12/14/2023). preprint.

(18) Kononova, O.; He, T.; Huo, H.; Trewartha, A.; Olivetti, E. A.; Ceder, G. Opportunities and Challenges of Text Mining in Materials Research. *iScience* **2021**, *24* (3), No. 102155.

(19) Kononova, O.; et al. Text-Mined Dataset of Inorganic Materials Synthesis Recipes. *Sci. Data* **2019**, *6* (1), 203.

(20) Krallinger, M.; et al. The CHEMDNER Corpus of Chemicals and Drugs and Its Annotation Principles. *J. Cheminform.* **2015**, *7* (1), S2.

(21) Lo, K. S2ORC: The Semantic Scholar Open Research Corpus. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. ACL 2020*. Ed. by Jurafsky, D.; Association for Computational Linguistics, 2020; 4969−4983. https://aclanthology.org/2020.acl-main.447 (visited on 12/14/2023).

(22) Mysore, S. The Materials Science Procedural Text Corpus: Annotating Materials Synthesis Procedures with Shallow Semantic Structures. In: *Proceedings of the 13th Linguistic Annotation Workshop. LAW 2019*. Ed. by Friedrich, A.; Zeyrek, D.; Hoek, J.; Association for Computational Linguistics: Florence, Italy, 2019; 56−64. https://aclanthology.org/W19-4007 (visited on 12/19/2023).

(23) Naik, A. Stress Test Evaluation for Natural Language Inference. In: *Proceedings of the 27th International Conference on Computational Linguistics*. Ed. by Bender, E. M.; Derczynski, L.; Isabelle, P.; Association for Computational Linguistics: Santa Fe, New Mexico, USA, 2018; 2340−2353. https://aclanthology.org/C18-1198/.

(24) Olivetti, E. A.; et al. Data-Driven Materials Research Enabled by Natural Language Processing and Information Extraction. *Appl. Phys. Rev.* **2020**, *7* (4), No. 041317.

(25) Panapitiya, G. Extracting Material Property Measurement Data from Scientific Articles. In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. EMNLP 2021*. Ed. by Moens, M.-F.; Association for Computational Linguistics: Punta Cana, Dominican Republic, 2021; 5393−5402. https://aclanthology.org/2021.emnlp-main.438 (visited on 12/04/2023).

(26) Peng, Y.; Yan, S.; Zhiyong, L. Transfer Learning in Biomedical Natural Language Processing: An Evaluation of BERT and ELMo on Ten Benchmarking Datasets. In: *Proceedings of the 18th BioNLP Workshop and Shared Task. BioNLP 2019*. Ed. by Demner-Fushman, D.; Association for Computational Linguistics: Florence, Italy, 2019; 58−65. https://aclanthology.org/W19-5006 (visited on 01/15/2024).
(27) Shetty, P.; et al. A General-Purpose Material Property Data Extraction Pipeline from Large Polymer Corpora Using Natural Language Processing. *npj Comput. Mater.* **2023**, *9* (1), 52.
(28) Swain, M. C.; Cole, J. M. ChemDataExtractor: A Toolkit for Automated Extraction of Chemical Information from the Scientific Literature. *J. Chem. Inf. Model.* **2016**, *56* (10), 1894−1904.
(29) Thawani, A. Representing Numbers in NLP: a Survey and a Vision. In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Ed. by Toutanova, K.; Association for Computational Linguistics, 2021; 644−656. https://aclanthology.org/2021.naacl-main.53.
(30) Wang, S. GPT-NER: Named Entity Recognition via Large Language Models. *arXiv preprint arXiv:2304.10428*. 2023
(31) Yang, Y.; Katiyar, A. Simple and Effective Few-Shot Named Entity Recognition with Structured Nearest Neighbor Learning. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). EMNLP 2020*. Ed. by Webber, B.; Association for Computational Linguistics, 2020; 6365−6375. https://aclanthology.org/2020.emnlp-main.516 (visited on 12/14/2023).
(32) Zheng, Z.; et al. ChatGPT Chemistry Assistant for Text Mining and the Prediction of MOF Synthesis. *J. Am. Chem. Soc.* **2023**, *145* (32), 18048−18062.

https://doi.org/10.1021/acs.jcim.4c01352
J. Chem. Inf. Model. 2025, 65, 4906−4917