Integrating Materials and Manufacturing Innovation
https://doi.org/10.1007/s40192-024-00363-5

THEMATIC SECTION: HARNESSING THE POWER OF MATERIALS DATA

Tackling Structured Knowledge Extraction from Polymer Nanocomposite Literature as an NER/RE Task with seq2seq

Bingyin Hu[1] · **Anqi Lin**[1] · **L. Catherine Brinson**[1]

Received: 1 October 2023 / Accepted: 16 May 2024

Abstract
There is an urgent need for ready access to published data for advances in materials design, and natural language processing (NLP) techniques offer a promising solution for extracting relevant information from scientific publications. In this paper, we present a domain-specific approach utilizing a Transformer-based model, T5, to automate the generation of sample lists in the field of polymer nanocomposites (PNCs). Leveraging large-scale corpora, we employ advanced NLP techniques including named entity recognition and relation extraction to accurately extract sample codes, compositions, group references, and properties from PNC papers. The T5 model demonstrates competitive performance in relation extraction using a TANL framework and an EM-style input sequence. Furthermore, we explore multi-task learning and joint-entity-relation extraction to enhance efficiency and address deployment concerns. Our proposed methodology, from corpora generation to model training, showcases the potential of structured knowledge extraction from publications in PNC research and beyond.

Keywords seq2seq · NER · RE · Polymer nanocomposites · Structured knowledge extraction · Natural language processing

Introduction and Background

With the advent of the materials genome initiative (MGI) [1, 2], there is a growing vision to integrate data science and machine learning to forge new capabilities in the understanding and design of materials, with applications from health care to advanced structures to renewable energy [3]. The vast majority of the work to date has focused on the development of machine learning algorithms to leverage data from computational models and make new discoveries [4–6]. The lack of FAIR (findable, accessible, interoperable, and reusable) materials data resources means that it is difficult to utilize data generated by peers in the field, pushing researchers to rely on computational models within their own laboratory that could generate large amounts of data, which is necessary for training a decent machine learning model. To address this issue, we have been developing MaterialsMine,[1] an ontology-driven open-source FAIR data resource for polymer nanocomposites (PNC) and metamaterials [7–11]. In addition to the data generated in our own laboratories, we have been curating experimental data from the literature into MaterialsMine. Data curation is challenging in that it requires domain knowledge, is extremely time consuming and its highly repetitive nature makes it prone to human error. Even in a subdomain of materials like PNCs, there were 70 k + publications in 2022 as suggested by searching "polymer nanocomposite" on Semantic Scholar,[2] along with an enormous number of important existing publications prior to 2022, none of which enable ready access to organized, annotated data. Based on MaterialsMine statistics, one PNC paper contains around 12 samples and each sample maps to 3 reported properties, meaning around 2.5 million of data points are published for PNC a year. Thus, the limited bandwidth of the progress of manually curated data poses a significant drag on the data-driven design of materials. As an alternative, we have a vision to harness the power of AI to extract data from the vast, published, archival literature, and to make that data FAIR by incorporating it into a robust materials data framework. Access to this enormous array of published data, both experimental and computational, would transform our ability to use existing knowledge in understanding and developing design paradigms for new

✉ L. Catherine Brinson
cate.brinson@duke.edu

[1] Department of Mechanical Engineering and Materials Science, Duke University, Durham, NC 27708, USA

[1] https://materialsmine.org/nm#/

[2] https://www.semanticscholar.org/search?year%5b0%5d=2022&year%5b1%5d=2022&q=polymer%20nanocomposite&sort=relevance

Published online: 01 July 2024

materials. Natural language processing (NLP), which seeks to automate the extraction of knowledge from human-written text, offers an opportunity to make this data accessible and readily reusable by humans and machines.

In the curation workflow for experimental data on materials as in MaterialsMine, the first step is to create a "sample list", which summarizes the experimental units or samples reported in the paper, denoting their composition and other distinguishing characteristics. Human curators use sample lists as guidance for curation. There are five major steps involved in sample list generation: (1) identify experimental samples and assign them unique sample codes as indices in the sample list, (2) identify each sample composition, 3) identify properties characterized for each sample, (4) associate sample code with composition, and 5) associate sample code with properties. In some cases, authors may provide a specific sample code, such as "Ep-SiO2-01," within the publication to denote a particular experimental sample. Rule-based syntactical matching algorithms can be employed to extract these sample codes. However, for many papers within the field of PNC, this is not the case. Common representations, such as "1 wt% silica in epoxy," require a deeper semantic understanding, rendering the syntactic approach less effective. Transformer models are well suited for sample list generation for their outstanding semantic understanding ability and wide applications on various NLP tasks [12–15]. The five steps can be smoothly translated into two structured prediction tasks within NLP. The task of named entity recognition (NER), which involves identifying and categorizing word-spans in a document, is suitable for the "identify" steps. The relation extraction (RE) task that seeks to predict the relationship between entities can be applied to the "associate" steps.

In this work, we propose to use domain-specific sequence-to-sequence (seq2seq) model for structured information extraction from polymer nanocomposite publications, focusing specifically on the sample list creation objective, by formulating the problem as an NER/RE task, benchmarked against some popular public pre-trained encoder models and their domain-specific variations. Related works are reviewed in "Related work" section. Details about how the dataset for pre-training and finetuning was collected and processed, and how pre-training and downstream tasks were conducted are summarized in "Methods" section. Results and discussion can be found in "Results and discussion" section, followed by conclusions in "Conclusions" section. Though the paper focuses on PNC publications, we would like to stress that the methodology is applicable to other scientific research domains, both within and outside materials science.

Related Work

To date, a few materials researchers have begun to apply NLP techniques to the NER task, focusing on inorganic materials like metal oxides [16], zeolites [17], and nanomaterials [18]. A similar recent effort utilizes rule-based heuristics and an unsupervised Snowball algorithm for relation extraction (RE) to generate ontologies for a class of crystallographic materials [19]. However, NER and RE for inorganic, crystalline materials are relatively simple because the compact chemical formula of inorganic materials acts as unique identifiers. In contrast, organic materials, especially polymers, cannot be uniformly represented [20]. PNCs are even more complex, with the introduction of nanofillers of complex geometry and chemistry. Accordingly, authors of PNC papers refer to experimental samples with fluid language; a single sample in a paper may be referred to as "1 wt% silica in epoxy," "epoxy/1 wt% SiO_2," "epoxy-SiO_2-0.01," or "Ep-SiO_2-01" interchangeably, which makes annotating a corpus for training an NER system difficult [21].

The large pre-trained transformer-based NLP models have achieved state-of-the-art performance in various downstream tasks, including NER and RE, in recent years [22]. Most of the NLP + materials science works leverage the "pre-training then finetuning" paradigm to train their models. It has been concluded in multiple works that the transformer-based models pre-trained on domain-specific corpora outperform the ones pre-trained on generic natural language since sentences in materials science publications are extremely specialized [23, 24]. Due to the uniqueness of the PNC language and the fact that materials science corpora used in existing works are kept private due to copyright concerns, we need to create our unannotated pre-training corpus and annotated finetune corpus from PNC publications.

Encoder-only transformer models, such as BERT, are pre-trained using denoising objectives that do not require annotation [13, 14, 25]. The goal is to teach the model the language by masking, shuffling, and other methods which introduce noise to the input sentence and then asking the model to restore the original sentence. After pre-training, there is an additional finetuning step, where the model is taught to perform a task of interest. For this finetuning step, we will need human annotations to provide the ground truth to the model.

Of all available models in the broad transformers model family, a group of BERT-based encoder models, such as BERT, RoBERTa, and DeBERTa, is often elected for structured prediction tasks [13, 26, 27]. Examples in the materials science domain include SciBERT, MatBERT, and MatSciBERT, all of which selected to pre-train

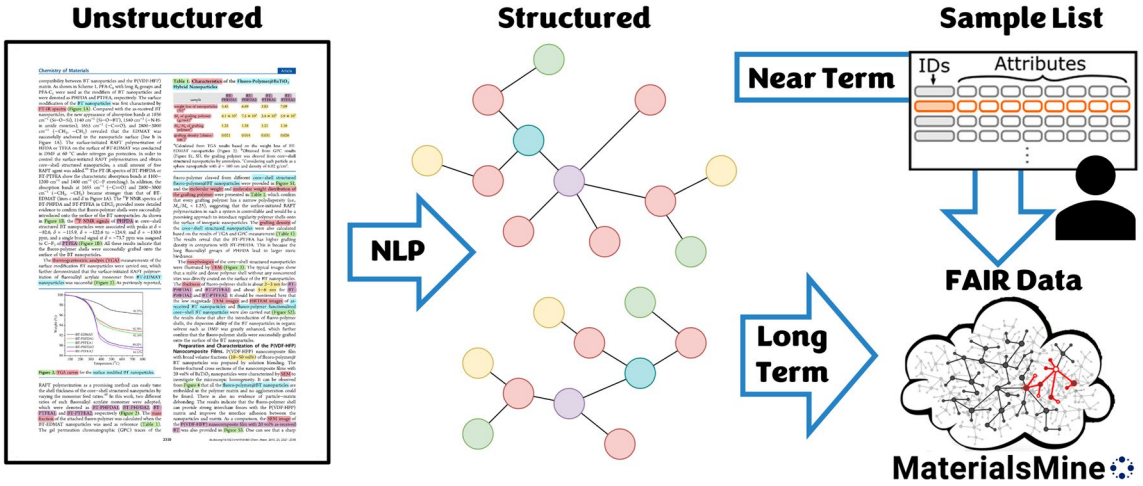

Fig. 1 Our vision of the NLP-driven curation pathway for MaterialsMine

BERT-based transformers with domain-specific corpora [28–30]. The "matching the blanks" (MTB) architecture as an extension to the BERT model is reported to perform reasonably well on RE tasks [31]. Though encoder models typically are the first options for structured prediction tasks like token classification, several works have shown that a seq2seq (or encoder-decoder) model can also perform well on such tasks by proxying structured prediction tasks as text generation tasks [32–34]. Treating structured prediction as generation allows for jointly making interrelated predictions without changing the architecture of the model. By proxying as text generation tasks with seq2seq models, we can provide different templates or task prefix to use one single model artifact for multiple tasks. Seq2seq (or encoder-decoder) models like T5 have shown their versatility on an array of NLP tasks, structured or unstructured, with one single model [14]. For example, the TANL framework was developed for an array of structured prediction tasks to be formed as a translation task between the target sequence and the input augmented natural language, building on top of seq2seq models like T5 [32].

Another big branch of transformer-based models that has become extremely impactful recently is decoder-only models, including GPT-3, GPT-4, PaLM, LLaMa, LlaMa-2 [15, 35–38]. Despite some early attempts to apply decoder-only models in materials science study [39], it has been reported that using decoder-only large language models (LLMs) like GPT on domain-specific tasks requires finetuning on domain-specific corpus [40]. LLMs usually have tens or hundreds of billions of parameters (GPT-4 is alleged to have trillions of parameters), which is too large to be fit into a 16 GB GPU like T5-base. Meanwhile, several studies report that by comparing performance on seq2seq tasks with seq2seq and decoder-only models of the same compute, i.e., they restrict the resource that a model could utilize for training to be the same, seq2seq models outperform LLMs [14, 25, 41]. Thus, while LLMs have promise which will be realized with time, research and additional compute capabilities, we have deployed T5, a seq2seq model, for the task described in this work.

As illustrated in Fig. 1, our vision is to create a semi-automated curation pathway by generating sample lists from PNC journal articles, that could gradually evolve into an automated curation pathway that could populate the MaterialsMine knowledge graph directly from the articles in future. This work serves as a first step toward the sustainable future of MaterialsMine driven by the automated curation pipeline.

Methods

Figure 2 provides the overall workflow of this work. In general, we start with data collection and cleaning, resulting in two corpora, one for pre-training and the other for finetuning, which requires annotation as well. Both T5 and BERT-based models will be finetuned for downstream tasks like NER and RE. In addition, we pre-trained our own domain-specific T5 model with the unannotated corpus. For BERT-based models, we can only finetune them as single-task NER or RE models (red dashed pathway). For T5 models, we finetuned them for all four realizations depicted in Fig. 2 (blue solid pathway). Entity pairs with relation for structured knowledge extraction from PNC papers can be obtained via either (1) two sequential calls to the two single-task models, or (2) two sequential calls to the multi-task model with different task prefixes, or (3) a single call to the joint-entity-relation extraction model. We will dive deep into each of the steps in the following sections.

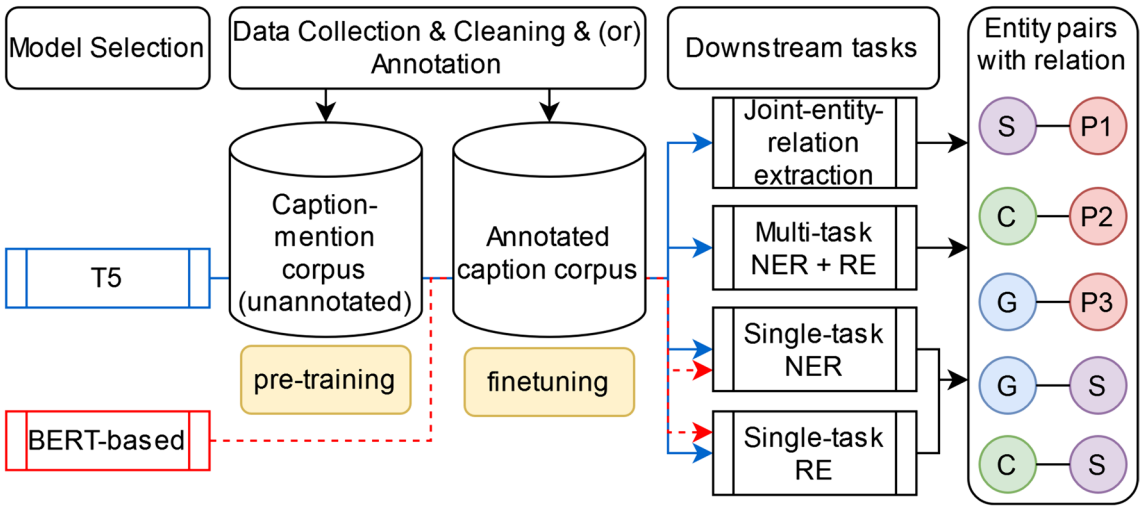

Fig. 2 General workflow of this work

Dataset Collection and Preparation

During manual sample list generation, curators are instructed to pay special attention to figures and tables since sample codes, composition, and properties are most likely reported in those components. Information can also be extracted from paragraphs where a discussion of figures or tables is presented. Learning from this practice, we built our PNC corpus using figure captions, table captions plus individual sentences from the text including figure and table referencing. All datasets use an 80:20 split for the training set and validation set.

Caption-Mention Corpus—the Pre-training Corpus

The caption-mention corpus discussed in this work consists of 1 M (1,002,904) sentences sourced from figure captions, table captions, and sentences that mention a figure or a table in the body text of 23,090 PNC papers. Figure 3 demonstrates the construction process of the caption-mention corpus.

A Scopus API query was utilized to obtain 99,985 DOI's with keyword filtering of "polymer + composite".[3] The obtained DOI's are further filtered by keywords ("poly" or "rubber") and "composite" in the abstract. DOI's of book chapters are removed from the collection. The list of DOI's is then grouped by the publishers, resulting in 18,210 DOI's from Elsevier, 4,880 DOI's from other publishers. The Elsevier corpus is obtained via the Elsevier API,[4] which returns XML's. The rest are obtained via an HTML scraper developed in-house. The markup language files are then parsed with a modified HtmlReader of the ChemDataExtractor package [42]. For each DOI, we store the abstract, the full text structured with top-level headers and content, figure captions, and table captions, all of which are normalized with the python unicodedata package.[5] We then extract sentences that mention a figure or a table from the full-text content. Finally, we use ChemDataExtractor to perform sentence segmentation on all the figure captions, table captions, and sentences that mention a figure or a table, to build our caption-mention corpus. Sentences with a length between 10 and 256 after tokenization are kept in the pre-training corpus.

Annotated Caption Corpus—the Finetune Corpus

The annotated caption corpus discussed in this work consists of 1896 captions collected from 214 PNC papers manually curated into the MaterialsMine data resource. Users can visit https://materialsmine.org for curated data. The doccano annotation platform is used for NER tagging and RE tagging [43]. This manual task is accomplished by two human curators, one who leads the annotation task and the other who verifies the annotation.

For NER tagging, we propose four classes of named entities for sample list generation purposes: *sample code* (S), *composition* (C), *group reference* (G), and *property* (P). For a span to be labeled with S, it must either be able to point any materials scientist to a unique experimental unit without reading through the full paper or used explicitly as a sample code in the paper. Though polymer/filler names are usually

[3] https://dev.elsevier.com/documentation/ScopusSearchAPI.wadl

[4] https://dev.elsevier.com/documentation/FullTextRetrievalAPI.wadl

[5] https://docs.python.org/3/library/unicodedata.html

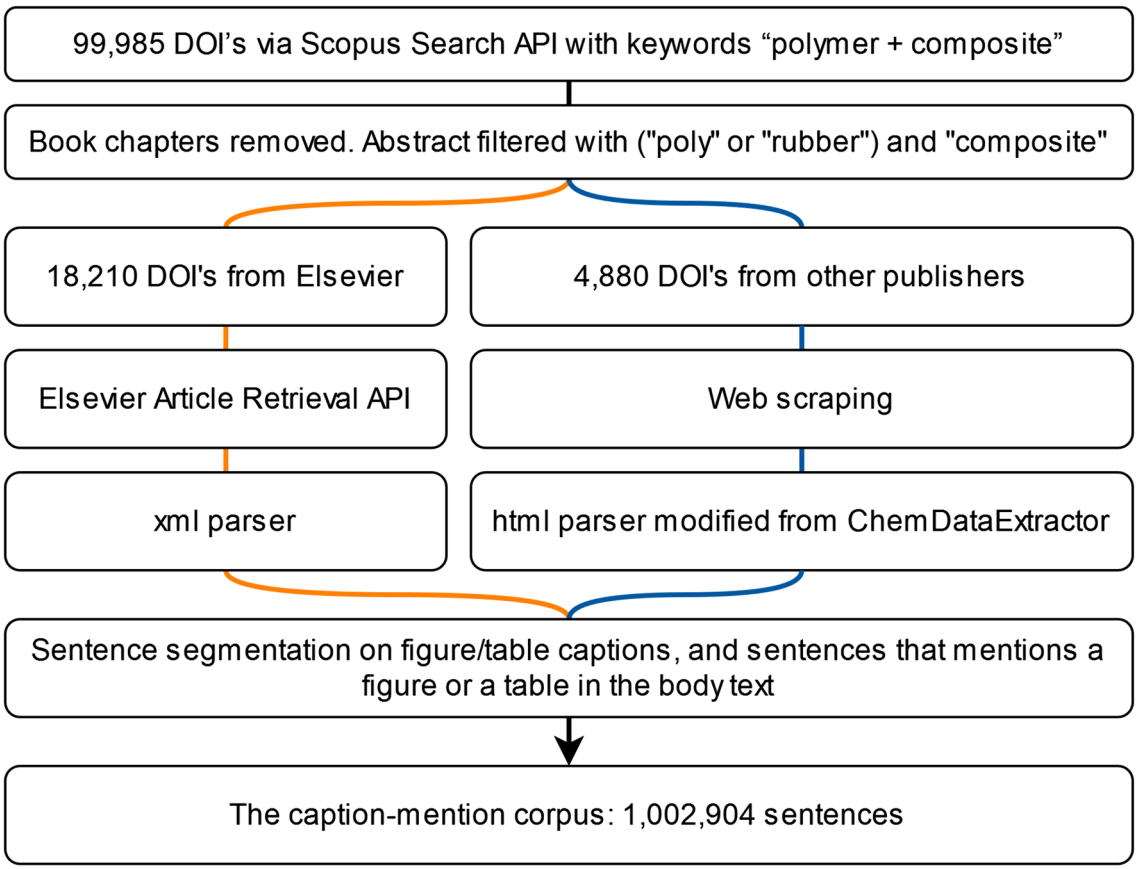

Fig. 3 Summary of the construction of the caption-mention corpus

included in the sample code, this work focuses on the sample code extraction. Spans that indicate nanofiller loadings like a mass fraction or volume fraction will be labeled with C. Similarly, spans that describe the property of interest in the figure or table are labeled with P, which can be any measurable that is characterized in materials science research. While being able to extract the actual value of a property is the North Star, this work focuses on extracting the property name. In PNC papers, it is common to compare properties within a group of PNCs with different nanofiller loadings in a figure or a table. Instead of listing all sample codes in the captions, authors usually use spans like "epoxy nanocomposites" or "silane-modified samples" as a group reference to multiple samples. Such spans are labeled with G. It is worth mentioning that C spans overlap with S spans on rare occasions. For example, "epoxy/1 wt% SiO_2" is a S span while "1 wt%" is a C span. Since most of the BERT-based NER models do not support overlapping named entities, we will remove overlapped C span in this case for simplicity in downstream tasks.

For RE tagging, we propose three relation classes: *isPropertyOf*, *isCompositionOf*, and *isMemberOf*. *isPropertyOf* can be applied to the (P, S) pair, (P, C) pair, and (P, G) pair, indicating a P span is reported for the other entity in the pair. *isCompositionOf* is straightforward as it can only be applied to the (C, S) pair. It is common that a caption contains multiple C tags and S tags, making the *isCompositionOf* class necessary. *isMemberOf* can be applied to the (S, G) pair and (C, G) pair, bridging the group reference to a sample or a smaller group of samples with identical nanofiller loadings. For detailed annotation guidelines, please refer to the online supplementary material.

Figure 4 is an example of a figure caption annotated for NER and RE in the doccano platform[26]. The resulting corpus has 2028 entities with the S label, 491 entities with the C label, 1606 entities with the G label, 2465 entities with the P label, 4262 entity pairs labeled with *isPropertyOf*, 633 entity pairs labeled with *isCompositionOf*, and 872 entity pairs labeled with *isMemberOf*.

Having introduced the named entities and relation classes, we can better understand the similarities between a sample list and the entity-relation-entity triples generated by the model proposed in this work as illustrated in Fig. 5.

Note that the gray dashed area is the final step to generate a graph which is equivalent to a row of data in the sample

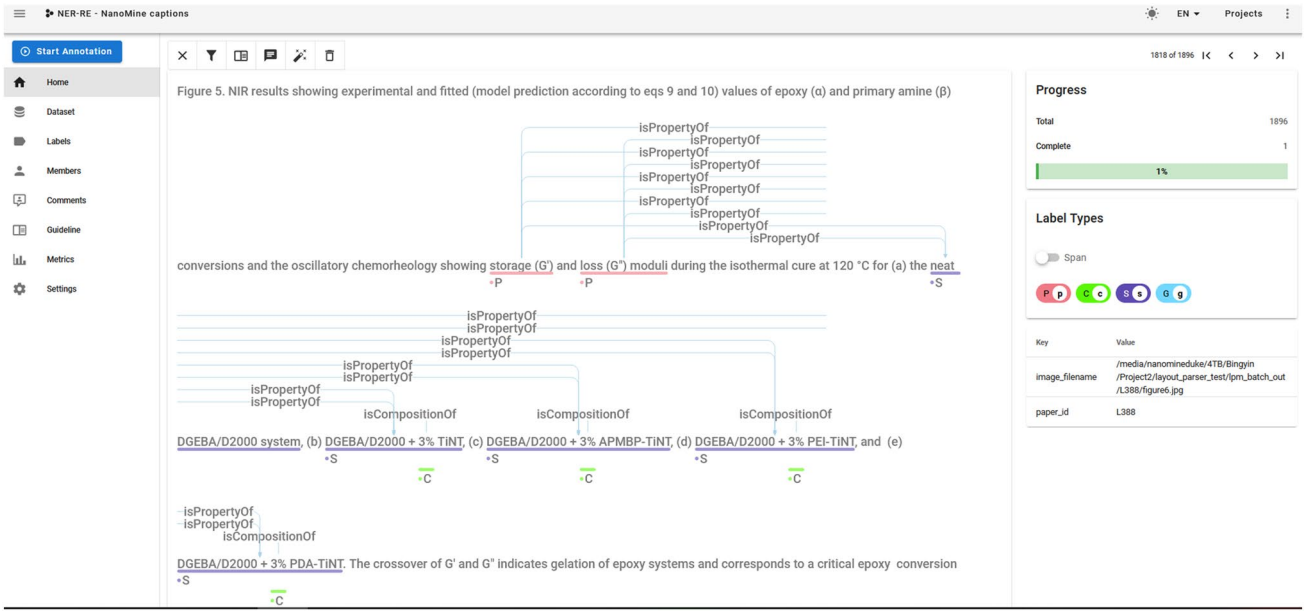

Fig. 4 A screenshot of the caption corpus annotation with doccano

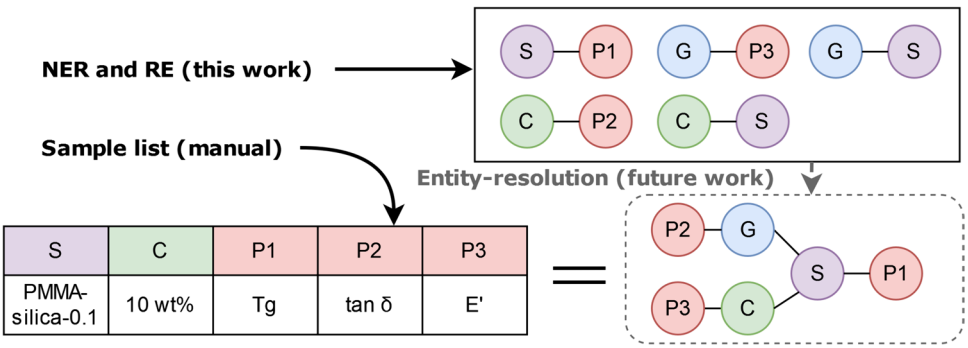

Fig. 5 Comparison between the manually curated sample list versus the entity-relation-entity triple

list. The final step involves an extra layer of entity resolution to find identical nodes semantically or physically for joining.

Datasets

For the NER task, an 80:20 split was adopted, resulting in 3045 sentences for training and 762 sentences for testing.

For the RE task, similarly, an 80:20 split resulted in 9328 sentences for training and 2332 sentences for testing. An additional "*Other*" label was added for 5893 entity pairs with no relation. For example, a caption can include descriptions of multiple sub-figures. An entity pair with one entity describing sub-figure (a) and the other entity describing sub-figure (d) is likely to be considered as "*Other*". Note that we did not label "*Other*" entity pair during human annotation. Instead, we automatically assign the "*Other*" label for entity pairs without a relation label in the same sentence. Since no entity pairs with the same NE labels were annotated with a relation, like P–P and S–S, we did not include those in generating the sentences for the RE task. For each entity pair with different NE labels, like P–S and C–S, we generate a sentence with either entity markers (EM) or the augmented natural language pre-processing for TANL. For the EM-style pre-processing, each NE of the entity pair was wrapped around with entity markers "<e1>", "</e1>", "<e2>", and "</e2>". For the TANL style pre-processing, please refer to the description of the relation classification task in their original paper [32].

Due to the limited size of annotated data, no dev set was spared from the training dataset. Models were finetuned on the test set, meaning our results represent an upper bound. More insights of the finetune corpus are available in the online supplementary material, with distributions of polymer matrices, nanofillers, and properties.

Pre-training of Domain-Specific T5

The T5-base model is pre-trained on the domain-specific unannotated caption-mention corpus. We used the same denoising pre-training objective as reported in the T5 paper that replaces dropped-out spans with sentinel tokens with a 15% corruption rate and an average of 3 tokens per corrupted span. A SentencePiece tokenizer is used here to break sentences into words and sub-words, referred to as tokens, and then it converts the textual input into numerical representations via a vocabulary look-up [44]. Our models are implemented with HuggingFace [45].

To fit in a 16 GB GPU, a batch size of 16 and a gradient accumulation step of 8 were selected, resulting in 128 total train batch size. Based on our experience of finetuning a T5 model and the pre-training configs reported in the original T5 paper [14], we evaluated 5 different combinations of optimizers and peak learning rates for pre-training: Case (1) AdamW optimizer with a peak learning rate of 5e-4, Case (2) AdamW optimizer with a peak learning rate of 5e-5, Case (3) AdaFactor optimizer with the AdaFactor scheduler that adjusts learning rate internally, Case (4) AdaFactor optimizer with an external peak learning rate of 1e-3, and Case (5) AdaFactor optimizer with an external constant learning rate of 1e-3. AdamW cases used a weight decay of 1e-3. Each model was scheduled to warmup for 5000 steps. A linear scheduler was utilized unless otherwise specified. Models were evaluated every 2500 steps. The maximum length of the input sequence is limited to 256. The best pre-trained model was trained on an NVIDIA Quadro P5000 GPU with 16 GB GPU RAM for 6 days. Pre-training codes are available at our GitHub repository.[6]

Downstream Tasks

NER

For the NER task, the BILOU tagging scheme (see SI) was adopted for pre-processing the labels. The input and label encodings generated by the tokenizers are truncated or padded to a fixed length of 200. For baselines, we assessed encoder models like DeBERTa-base, MatBERT, and MatSciBERT, and seq2seq models like TANL for NER with T5 as the starting point, and two other formulations of the target sequence for T5 to treat NER as a text generation task. In the first formulation, the T5 model predicts a sequence of label tokens, denoted as $T5_{\text{label seq}}$. The second option is to predict an interleaved style of word token and label token, denoted as $T5_{\text{interleave}}$. An example of the two formulations is as follows.

Input: Fig. 3. Tg of PMMA-silica-0.1.
Output (label sequence): "<O><O><O><U-P><O><B-S><I-S><I-S><I-S><L-S><O>"
Output (interleave): Fig<O>3<O>.<O>Tg<U-P>of<O>PMMA<B-S>-<I-S>silica<I-S>-<I-S>0.1<L-S>.<O>

Apart from the baselines, we also assessed three seq2seq formulations, namely TANL, $T5_{\text{label seq}}$, and $T5_{\text{interleave}}$, with our domain-specific T5 model for the NER task.

Models were evaluated on micro-averaged precision, recall, and F1 score for the NER task. Each model was finetuned until the F1 score stops increasing with 5 random seeds unless otherwise specified.

RE

For the RE task, baselines include the "matching the blank" (MTB) architecture on top of the BERT, MatBERT, and MatSciBERT model with entity marker (EM) as a state-of-the-art architecture for RE task among the encoder models, and the TANL model built on top of T5 with the augmented natural language for the relation classification task. We assessed our domain-specific T5 model on the RE task with two proposed approaches: (1) use the TANL framework but with our domain-specific T5 model, and (2) an EM-style input sequence and relation triple style output sequence. For the EM-style finetuning, similar to the NER task, we added entity markers and relation labels wrapped in "<" and ">" as additional special tokens to the T5 tokenizer. An example target sequence will be "<isPropertyOf><e2><e1>", meaning entity 2 is property of entity 1. Input sequences were truncated or padded to a fixed length of 200.

Micro-averaged F1 score was used as the metric for model evaluation. Each model was finetuned until the F1 score stops increasing with 5 random seeds unless otherwise specified.

Since our goal is to create a pipeline for sample list generation in MaterialsMine, using individual single-task models for NER and RE separately might bring deployment concerns. The charm of the seq2seq model lies in its multi-tasking ability. There are two options for us to use one seq2seq model for both tasks, namely a multi-task seq2seq model, and a joint-entity-relation extraction model.

Multi-task NER + RE

A multi-task TANL on top of our pre-trained T5 model was trained in a multi-task setting with a separated NER dataset and RE dataset. A task prefix, like "NM_NER:" and "NM_REL:" was added to each sentence as a prefix. The micro-averaged F1 score of this model on the NER task and the RE task will be compared with single-task models as well.

[6] https://github.com/bingyinh/NLP_PNC_sample_list

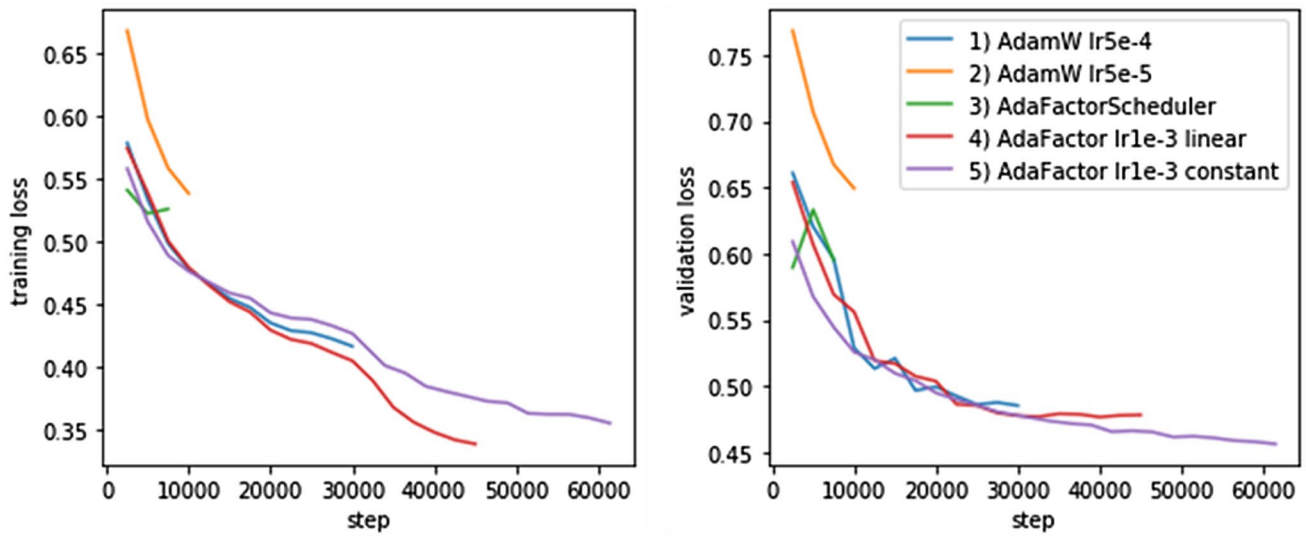

Fig. 6 Pre-training history of domain-specific T5 models with caption-mention corpus with early stopping enabled

Joint-Entity-Relation Extraction

A TANL model starting with our pre-trained T5 model was trained for a joint-entity-relation extraction task as well. After being translated into the augmented natural language, an example input sentence will become:

> **Input:** Fig. 3. Tg of PMMA-silica-0.1.
> **Output:** Fig. 3. [Tg | *P* | *isPropertyOf* = PMMA-silica-0.1] of [PMMA-silica-0.1 | *S*].

Note that the evaluation of the RE task in the joint-entity-relation extraction task is contingent on the performance of the NER task. For example, if none of the named entities can be detected in the sequence, no relation will be detected as well in a joint-entity-relation extraction setting, while in the standalone or multi-task settings, RE inference has labeled named entities in the input sequence. Therefore, it will be unfair to compare the performance on the RE task of the joint-entity-relation extraction directly to the other two settings. We include the performance metrics here just for reference. It is worth noting that the joint-entity-relation extraction is the more realistic setting since NE labels are not ordinarily available during inference.

Hyperparameters

For T5 models, we tested a wide range of learning rates from [5e-5, 1e-4, 2e-4, 3e-4, 4e-4, 5e-4, 1e-3], weight decay from [1e-4, 1e-3, 1e-2, 1e-1], batch size from [8, 16], number of beams from [5, 10]. We did not set a cap on the training epochs for any of the models being assessed. The stopping criteria are purely based on the F1 score.

Results and Discussion

Pre-training of Domain-specific T5

The pre-training history is provided in Fig. 6.

The five cases were experimented on one after another. Cases (2) and (3) perform significantly worse than case (1) at around 10,000 steps. Thus, they were terminated early. Case (4) showed a decreasing trend at 30,000 steps, so it was trained for another 30,000 steps with the scheduler started afresh until the validation loss converged. Case (5), which performs the best on the validation set, was also kept training until 60,000+ steps. Case (5), AdaFactor with constant external learning rate at 1e-3, as reported in the T5 paper for finetuning, obtained the lowest validation loss at 0.457.

Downstream Tasks

NER

Table 1 summarizes the micro-averaged precision, recall, and F1 scores of the assessed models on the NER task. Note that the multi-task TANL model and the joint-entity-relation extraction TANL model are included in the table along with other single-task NER models.

As expected, encoder models perform well on the NER task. MatBERT, which was pre-trained on a corpus consisting of 2 M full-text materials science journal articles, performs the best in the NER task if we ignore the joint-entity-relation TANL model. The DeBERTa model, as an advanced BERT model with disentangled attention, outperforms the domain-specific MatSciBERT model despite not being pre-trained on a domain-specific corpus. Interestingly,

Table 1 Performance on the NER task collected from 5 random runs

	Precision	Recall	F1
DeBERTa-base	79.8 ± 0.7	82.6 ± 0.4	81.2 ± 0.4
MatBERT	83.3 ± 0.8	83.2 ± 1.9	<u>83.0 ± 1.3</u>
MatSciBERT	78.0 ± 0.5	83.8 ± 0.5	80.8 ± 0.5
TANL	79.9 ± 0.8	80.2 ± 0.8	80.1 ± 0.6
$T5_{label\ seq}$*	79.6	79.4	79.5
$T5_{interleave}$*	73.0	77.2	75.0
With domain-specific T5-base			
domain-specific $T5_{label\ seq}$*	80.9	82.3	81.6
domain-specific $T5_{interleave}$*	57.0	80.2	66.6
TANL + domain-specific T5	80.6 ± 0.8	80.2 ± 0.3	80.4 ± 0.4
$TANL_{multi\text{-}task}$ + domain-specific T5	81.5 ± 0.6	81.4 ± 0.5	81.4 ± 0.6
$TANL_{joint\text{-}entity\text{-}relation}$ + domain-specific T5	85.4 ± 0.6	82.3 ± 0.6	**83.8 ± 0.5**

Bold value indicates best performing model, while underline value indicates the second best performing model

*Models were trained without 5 random runs

all three TANL with domain-specific T5 models, including $TANL_{multi\text{-}task}$ and $TANL_{joint\text{-}entity\text{-}relation}$, obtain better F1 scores than the single-task TANL. This finding suggests that learning for the RE task can be beneficial to the NER task. Our domain-specific T5-base model helps the label sequence formulation increase its F1 score from 79.5 to 81.6, which is still impressive given that our caption-mention corpus is more than 100 times smaller than the MatBERT pre-training corpus since we only used caption-related sentences excerpted from a total of 23 k papers, and that it outperforms the MatSciBERT model, which was pre-trained on ~ 150 k full-text materials science journal articles. Surprisingly, the $T5_{interleave}$ model suffers a significant performance drop with the domain-specific T5.

RE

The micro-averaged F1 scores after evaluating multiple RE models on our annotated caption corpus can be found in Table 2. Again, the multi-task TANL model and the joint-entity-relation extraction TANL model are included in the table as well.

Since only 3 relation classes were annotated in our fine-tune corpus, the performance of all models listed in Table 2, except for $TANL_{joint\text{-}entity\text{-}relation}$, is strong, while a clear gap exists between the 3 BERT-based encoder models and the T5-based seq2seq models. The best micro-averaged F1 score of 96.9 was reached by the $TANL_{multi\text{-}task}$ model. As we mentioned before, it is unfair to compare the performance of the $TANL_{joint\text{-}entity\text{-}relation}$ model on the RE task directly with the other models because the other models predict on true NE labeled input sequence while the joint model does not. On the other hand, around 30% of the sequences generated by $TANL_{joint\text{-}entity\text{-}relation}$ model cannot match the input tokens exactly, which is called a "wrong construction" in the TANL framework. According to the input and output sequence examples we provided in the Methods section, the augmented natural language allocates a longer span to RE expressions than NER expressions. A failed reconstruction thus impacts the RE task more than the NER task. The aforementioned two reasons lead to a high NER score but a low RE score for the $TANL_{joint\text{-}entity\text{-}relation}$ model.

Discussion

Figure 7 summarizes the three potential approaches to implement the encoder or seq2seq models for NER and RE tasks.

Approach 1 uses one single-task model for NER and another for RE. In that case, based on performance shown in Table 1 and Table 2, one would choose the MatBERT

Table 2 Performance on the RE task collected from 5 random runs

	F1
MTB + $BERT_{EM}$	91.9 ± 0.2
MTB + $MatBERT_{EM}$	93.6 ± 0.6
MTB + $MatSciBERT_{EM}$	94.8 ± 0.2
TANL	95.6 ± 0.6
With domain-specific T5-base	
domain-specific $T5_{EM}$*	<u>95.8</u>
TANL + domain-specific T5	95.5 ± 0.4
$TANL_{multi\text{-}task}$ + domain-specific T5	**96.9 ± 0.2**
$TANL_{joint\text{-}entity\text{-}relation}$ + domain-specific T5	72.0 ± 0.9

Bold value indicates best performing model, while underline value indicates the second best performing model

*Models were trained without 5 random runs

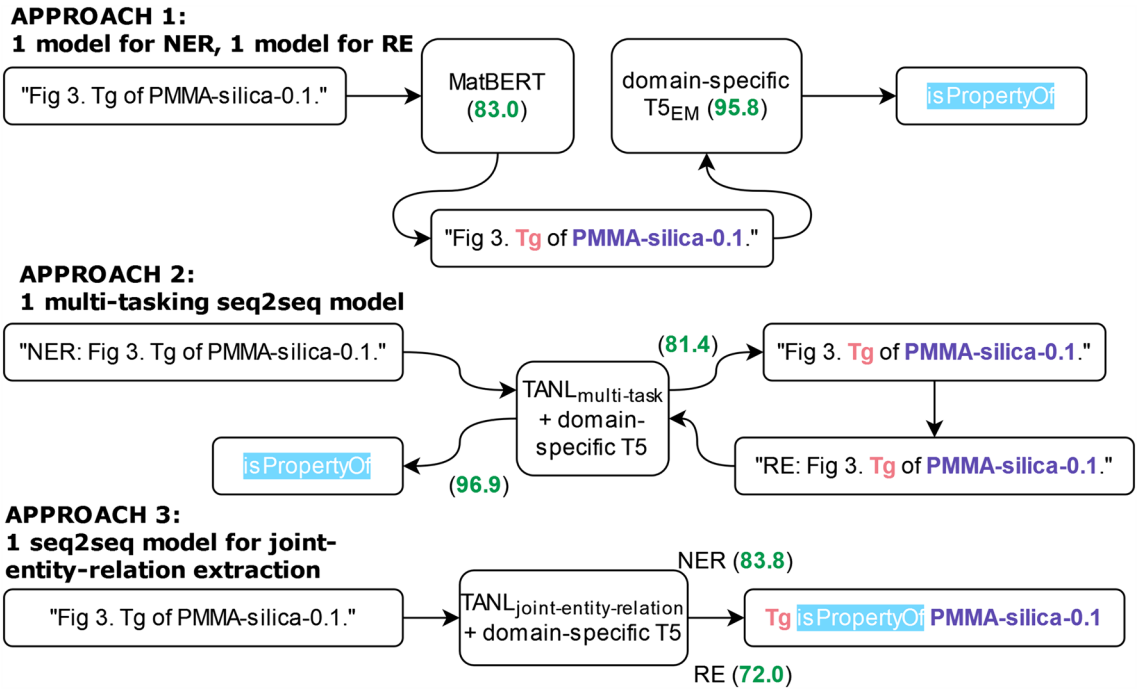

Fig. 7 Proposed pipelines for the application of NER and RE models in MaterialsMine data curation. F1 scores shown in parentheses in green

for NER and the domain-specific $T5_{EM}$ for RE. Approach 2 and Approach 3 only require one model. The difference lies in that Approach 2 uses a multi-tasking model, which will be called twice for each pass, and Approach 3 uses a model finetuned for joint-entity-relation extraction task that predicts NEs and relationships simultaneously in a single pass. Again, the RE F1 score here for Approach 3 has a different setting from other RE F1 scores. As discussed in the previous section, there is a 30% "wrong construction" issue in TANL. Another reason for the significantly low RE score is that the relation extraction part of a joint-entity-relation extraction task in TANL does not use input sentences with correctly labeled NE as other single-task RE or multi-task RE models do. To roughly equilibrate the impact of the incorrectly labeled NEs in the input, a score of 85.9 can be obtained by dividing 72.0 with 83.8, which is still low but in line with other models. Application-wise, the RE score in a joint-entity-relation extraction task is closer to the real use case where the performance on the RE task is impacted by the NER task.

When considering Approach 1 and Approach 2, if the primary concern is performance, Approach 1 would be preferable due to its higher NER score, as it is inferred from Approach 3 that the NER score has a significant impact on the RE score in a practical applications. However, from a production standpoint, a pipeline incorporating a single multi-task model offers several advantages over one with two single-task models, including efficient resource utilization, reduced operational costs, and simplified code base.

In addition, we propose that the multi-task model has the potential to do better on the NER task. The fact that the joint-entity-relation extraction model performs the best on the NER task suggests that the performance difference between MatBERT and $TANL_{multi\text{-}task}$ + domain-specific T5 is not about the model structure, but about the size of pre-training corpus. As mentioned, the MatBERT model was pre-trained from the BERT-uncased-base model with 2 M full-text materials science publications. In contrast, our pre-training corpus consists of captions and caption-related sentences extracted from 23 k polymer nanocomposite papers, a corpus more than 100 k times smaller. Limiting the pre-training corpus to only caption-related sentences might also limit the semantic understanding of T5. Thus, future work includes extending our pre-training corpus to include full-text PNC

Table 3 NE-class-level performance of the $TANL_{multi\text{-}task}$ + Pre-trained T5-base model

Named entity class	Precision	Recall	F1
(S)ample code	84.6 ± 1.6	84.9 ± 1.1	84.8 ± 1.2
(C)omposition	70.6 ± 1.9	72.4 ± 1.1	71.4 ± 0.5
(P)roperty	85.0 ± 0.7	84.4 ± 0.7	84.7 ± 0.7
(G)roup reference	73.6 ± 1.1	73.2 ± 1.5	73.4 ± 0.5

papers and pre-training the T5 model on the extended corpus.

Overall, the multi-task approach is the optimal solution for sample list generation in our use case of the MaterialsMine platform. Taking a closer look at the NER performance on individual NE classes, Table 3 summarizes the precision, recall, and F1 score of each named entity (NE) class over 5 random runs.

Impressively, the sample code and property named entities can be detected with an F1 score of 84.8 and 84.7. As shown in Fig. 5, the final graph that is equivalent to a row in the sample list will be triples joining on the sample code. Thus, performing well on the sample code class is critical. The composition class, in a few cases, can be detected via regular expressions (regex) as well, which might be a potential augmenting solution to improve the performance. The group reference is the most complicated and natural-language-like class in this work which confuses human curators from time to time. It is not surprising that the performance of our multi-task model is less competitive in predicting the G class.

One limitation of our work is the inability to directly extract polymer and filler names as separate named entities, primarily due to constraints in human annotation resources. However, we propose leveraging rule-based algorithms to segment the sample code into distinct entities, and then facilitating polymer/filler name detection through two potential approaches. The first approach is to call the ChemProps API that our group developed for polymer and keyword name standardization [11]. The service can detect variations of 129 popular polymer names and 54 common filler names. The second approach is to generate embeddings for segmented entities with an encoder or Word2Vec model like the work by Shetty and Ramprasad [46]. Subsequently, utilizing K Nearest Neighbor with cosine distance within a pool of embeddings for popular polymer names with a cutoff threshold to filter out non polymer name entities.

The ultimate goal of our framework is to curate PNC data in a fully automated way. However, two significant challenges persist. The first challenge relates to the necessity of an entity resolution layer to merge entity pairs into a graph, despite the fluid language used by authors, as presented in Fig. 5. The second challenge is about numerical value detection and allocation, a task that is indeed achievable but demands additional NER/RE labels, more data, and extra human annotation efforts. Like many other NLP challenges within scientific domains, the primary obstacle remains the scarcity of human annotation resources, given its requisite domain expertise, rendering crowd-sourcing ineffective.

Conclusions

In this work, we presented the methods we used to collect a domain-specific unannotated corpus for pre-training and a domain-specific annotated corpus for finetuning an array of BERT-based models and seq2seq models for NER and RE tasks on captions excerpted from PNC publications. A domain-specific T5-base model was pre-trained using 1 M caption-related sentences collected from 23 k PNC articles. A finetune corpus containing 1,896 figure captions from PNC papers was annotated with named entities from 4 classes and relations from 3 classes.

The NER task results showed that a large pre-training corpus is critical to boost the performance as MatBERT outperforms other single-task models. The caption-mention corpus also helps improve the performance of our T5 model with a label sequence formulation, which performs better than the domain-specific MatSciBERT model despite a significantly smaller pre-training corpus.

For the RE task, our T5 models, one utilizing the TANL framework and another using an EM-style input sequence with relation triple output sequence, demonstrated competitive performance in terms of micro-averaged F1 score.

To enhance efficiency and address deployment concerns, multi-task learning and joint-entity-relation extraction were explored. The multi-task TANL model, trained on separate NER and RE datasets, achieved promising results in both tasks. The joint-entity-relation extraction task model has a satisfactory NER F1 score and a low RE F1 score, due to the inherent complexity and interdependence of jointly predicting both NER and RE in a single model. It also suggests that the NER task, as the upstream, plays a crucial role in enabling accurate RE in practical use.

Overall, our study showcases the potential of using a domain-specific T5 model for automating the process of sample list generation for accelerating data curation of experimental data from published materials papers. The proposed methodology was demonstrated for a specific use case of PNCs and enables efficient data extraction on targeted information for experimental samples from this specific materials domain. This method will facilitate manual curation, leading to faster ingestion of data into materials specific data repositories. As this method is expanded, additional entities can be added to the automated extraction tasks. The methodology can also be applied to other scientific domains, within and outside materials science, for efficient structured data extraction from publications. Automated curation to provide fully annotated data into materials repositories from the historical materials literature will enable new materials discoveries and advances.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s40192-024-00363-5.

Acknowledgments We would like to thank Dr. Sam Wiseman for the constructive advice on the conceptualization of this work and the comments on the manuscript. We would like to thank Defne Circi for providing comments on the manuscript. The authors gratefully acknowledge support of the NSF CSSI program (OAC-1835677).

Code Availability Pre-trained and finetuned models are available on HuggingFace for public access. Pre-trained domain-specific T5 model: bingyinh/pretrained_t5_polymer_composite_caption. TANL + domain-specific T5 model for single-task NER: bingyinh/TANL-based_MaterialsMine_NER. TANL + domain-specific T5 model for single-task RE: bingyinh/TANL-based_MaterialsMine_RE. $\text{TANL}_{\text{multi-task}}$ + domain-specific T5 model: bingyinh/TANL-based_MaterialsMine_NER_RE_Multitask. $\text{TANL}_{\text{joint-entity-relation}}$ + domain-specific T5 model: bingyinh/TANL-based_MaterialsMine_joint_entity_relation.

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. National Science and Technology Council (2011) Materials genome initiative for global competitiveness. https://www.mgi.gov/sites/default/files/documents/materials_genome_initiative-final.pdf
2. National Science and Technology Council (2021) Materials genome initiative strategic plan. https://www.mgi.gov/sites/default/files/documents/MGI-2021-Strategic-Plan.pdf
3. Morgan D, Jacobs R (2020) Opportunities and challenges for machine learning in materials science. Annu Rev Mater Res 50:71–103. https://doi.org/10.1146/annurev-matsci-070218-010015
4. Himanen L, Geurts A, Foster AS, Rinke P (2019) Data-driven materials science: status, challenges, and perspectives. Adv Sci. https://doi.org/10.1002/advs.201900808
5. Schleder GR, Padilha ACM, Acosta CM et al (2019) From DFT to machine learning: recent approaches to materials science—a review. J Phys Mater 2:032001. https://doi.org/10.1088/2515-7639/ab084b
6. Choudhury A (2021) The Role of machine learning algorithms in materials science: a state of art review on industry 4.0. Arch Comput Methods Eng 28:3361–3381. https://doi.org/10.1007/s11831-020-09503-4
7. Zhao H, Li X, Zhang Y et al (2016) Perspective: NanoMine: a material genome approach for polymer nanocomposites analysis and design. APL Mater 4:053204. https://doi.org/10.1063/1.4943679
8. Zhao H, Wang Y, Lin A et al (2018) NanoMine schema: an extensible data representation for polymer nanocomposites. APL Mater 6:111108. https://doi.org/10.1063/1.5046839
9. Brinson LC, Deagen M, Chen W et al (2020) Polymer nanocomposite data: curation, frameworks, access, and potential for discovery and design. ACS Macro Lett 9:1086–1094. https://doi.org/10.1021/ACSMACROLETT.0C00264/ASSET/IMAGES/LARGE/MZ0C00264_0006.JPEG
10. Deagen ME, McCusker JP, Fateye T et al (2022) FAIR and interactive data graphics from a scientific knowledge graph. Sci Data 91(9):1–11. https://doi.org/10.1038/s41597-022-01352-z
11. Hu B, Lin A, Brinson LC (2021) ChemProps: a RESTful API enabled database for composite polymer name standardization. J Cheminform. https://doi.org/10.1186/s13321-021-00502-6
12. Vaswani A, Shazeer N, Parmar N, et al (2017) Attention is all you need. https://doi.org/10.48550/arXiv.1706.03762
13. Devlin J, Chang M-W, Lee K, Toutanova K (2019) BERT: Pre-training of deep bidirectional transformers for language understanding. https://doi.org/10.18653/v1/N19-1423
14. Raffel C, Shazeer N, Roberts A, et al (2020) Exploring the limits of transfer learning with a unified text-to-text transformer. https://dl.acm.org/doi/abs/10.5555/3455716.3455856
15. OpenAI (2023) GPT-4 technical report. https://doi.org/10.48550/arXiv.2303.08774
16. Weston L, Tshitoyan V, Dagdelen J et al (2019) Named entity recognition and normalization applied to large-scale information extraction from the materials science literature. J Chem Inf Model 59:3692–3702. https://doi.org/10.1021/acs.jcim.9b00470
17. Jensen Z, Kim E, Kwon S et al (2019) A machine learning approach to zeolite synthesis enabled by automatic literature data extraction. ACS Cent Sci. https://doi.org/10.1021/acscentsci.9b00193
18. Hiszpanski AM, Gallagher B, Chellappan K et al (2020) Nanomaterial synthesis insights from machine learning of scientific articles by extracting, structuring, and visualizing knowledge. J Chem Inf Model. https://doi.org/10.1021/acs.jcim.0c00199
19. Agichtein E, Gravano L, Snowball: extracting relations from large plain-text collections. https://dl.acm.org/doi/10.1145/336597.336644
20. Shetty P, Rajan AC, Kuenneth C et al (2023) A general-purpose material property data extraction pipeline from large polymer corpora using natural language processing. npj Comput Mater 9:1–12. https://doi.org/10.1038/s41524-023-01003-w
21. Tchoua RB, Ajith A, Hong Z, et al (2019) Creating training data for scientific named entity recognition with minimal human effort. In: Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics).
22. Min B, Ross H, Sulem E et al (2021) Recent advances in natural language processing via large pre-trained language models: a survey. ACM Comput Surv 56(2):30. https://doi.org/10.1145/3605943
23. Olivetti EA, Cole JM, Kim E et al (2020) Data-driven materials research enabled by natural language processing and information extraction. Appl Phys Rev 7:041317. https://doi.org/10.1063/5.0021106
24. Kononova O, He T, Huo H et al (2021) Opportunities and challenges of text mining in materials research. iScience 24:102155. https://doi.org/10.1016/j.isci.2021.102155
25. Tay Y, Dehghani M, Tran VQ, et al (2022) UL2: unifying language learning paradigms. https://doi.org/10.48550/arXiv.2205.05131
26. Liu Y, Ott M, Goyal N, et al (2019) RoBERTa: a robustly optimized BERT pretraining approach. https://doi.org/10.48550/arxiv.1907.11692
27. He P, Liu X, Gao J, Chen W (2020) DeBERTa: decoding-enhanced BERT with disentangled attention. https://doi.org/10.48550/arXiv.2006.03654
28. Beltagy I, Lo K, Cohan A (2019) SCIBERT: a pretrained language model for scientific text. In: EMNLP-IJCNLP 2019 - 2019 conference on empirical methods in natural language processing and 9th international joint conference on natural language processing, proceedings of the conference.
29. Trewartha A, Walker N, Huo H et al (2022) Quantifying the advantage of domain-specific pre-training on named entity recognition tasks in materials science. Patterns. https://doi.org/10.1016/j.patter.2022.100488
30. Gupta T, Zaki M, Krishnan NMA, Mausam (2022) MatSciBERT: a materials domain language model for text mining and information extraction. npj Comput Mater. https://doi.org/10.1038/s41524-022-00784-w

31. Soares LB, FitzGerald N, Ling J, Kwiatkowski T (2020) Matching the blanks: distributional similarity for relation learning. In: ACL 2019 - 57th annual meeting of the association for computational linguistics, proceedings of the conference.
32. Paolini G, Athiwaratkun B, Krone J, et al (2021) Structured prediction as translation between augmented natural languages. https://doi.org/10.48550/arxiv.2101.05779
33. Min B, Ross H, Sulem E et al (2023) Recent advances in natural language processing via large pre-trained language models: a survey. ACM Comput Surv. https://doi.org/10.1145/3605943
34. Lu Y, Liu Q, Dai D, et al (2022) Unified structure generation for universal information extraction. https://doi.org/10.18653/v1/2022.acl-long.395
35. Brown TB, Mann B, Ryder N, et al (2020) Language models are few-shot learners. https://doi.org/10.48550/arXiv.2005.14165
36. Chowdhery A, Narang S, Devlin J, et al (2022) PaLM: scaling language modeling with pathways. https://dl.acm.org/doi/10.5555/3648699.3648939
37. Touvron H, Lavril T, Izacard G, et al (2023) LLaMA: open and efficient foundation language models. https://doi.org/10.48550/arXiv.2302.13971
38. Touvron H, Martin L, Stone K, Llama 2: open foundation and fine-tuned chat models. https://doi.org/10.48550/arXiv.2307.09288
39. Jablonka KM, Ai Q, Al-Feghali A, et al (2023) 14 Examples of how LLMs can transform materials science and chemistry: a reflection on a large language model hackathon. https://doi.org/10.1039/D3DD00113J
40. Pal S, Bhattacharya M, Lee S-S, Chakraborty C (2023) A domain-specific next-generation large language model (LLM) or ChatGPT is required for biomedical engineering and research. Ann Biomed Eng. https://doi.org/10.1007/s10439-023-03306-x
41. Fu Z, Lam W, Yu Q, et al (2023) Decoder-only or encoder-decoder? Interpreting language model as a regularized encoder-decoder. https://doi.org/10.48550/ARXIV.2304.04052
42. Swain MC, Cole JM (2016) ChemDataExtractor: a toolkit for automated extraction of chemical information from the scientific literature. J Chem Inf Model 56:1894–1904. https://doi.org/10.1021/acs.jcim.6b00207
43. Nakayama H, Kubo T, Kamura J, et al (2018) Doccano: text annotation tool for human. https://github.com/doccano/doccano
44. Kudo T, Richardson J (2018) SentencePiece: a simple and language independent subword tokenizer and detokenizer for neural text processing. https://doi.org/10.18653/v1/D18-2012
45. Wolf T, Debut L, Sanh V, et al (2020) Transformers: state-of-the-art natural language processing. In: Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations. Association for Computational Linguistics, Online, pp 38–45
46. Shetty P, Ramprasad R (2021) Automated knowledge extraction from polymer literature using natural language processing. iScience. https://doi.org/10.1016/j.isci.2020.101922