POLYIE: A Dataset of Information Extraction from Polymer Material Scientific Literature

Anonymous ACL submission

Abstract

Scientific information extraction (SciIE), which aims to automatically extract information from scientific literature, is becoming more important than ever. However, there are no existing SciIE datasets for polymer materials, which is an important class of materials used ubiquitously in our daily lives. To bridge this gap, we introduce POLYIE, a new SciIE dataset for polymer materials. POLYIE is curated from 146 full-length polymer scholarly articles, which are annotated with different named entities (i.e., materials, properties, values, conditions) as well as their N-ary relations by domain experts. POLYIE presents several unique challenges due to diverse lexical formats of entities, ambiguity between entities, and variable-length relations. We evaluate state-of-the-art named entity extraction and relation extraction models on POLYIE, analyze their strengths and weaknesses, and highlight some difficult cases for these models. To the best of our knowledge, POLYIE is the first SciIE benchmark for polymer materials, and we hope it will lead to more research efforts from the community on this challenging task. Our code and data are available on: https://anonymous.4open.science/r/PolyIE.

1 Introduction

Material science literature is growing at an unprecedented rate. For example, a simple search on Google Scholar with the term "polymers" returns more than 5 million articles on polymer materials. Such literature reports valuable information on the latest advances in material science, ranging from experimental material properties to material synthesis recipes and procedures. As machine learning (ML) has achieved success in different applications of material science (Butler et al., 2018; Schmidt et al., 2019), Scientific Information Extraction (SciIE) from literature for supporting various tasks is becoming increasingly important. Automatically extracting structured information about materials from massive unstructured literature data can be invaluable to understanding material properties and synthesis, as well as building data-driven ML tools for material discovery (Court et al., 2021).

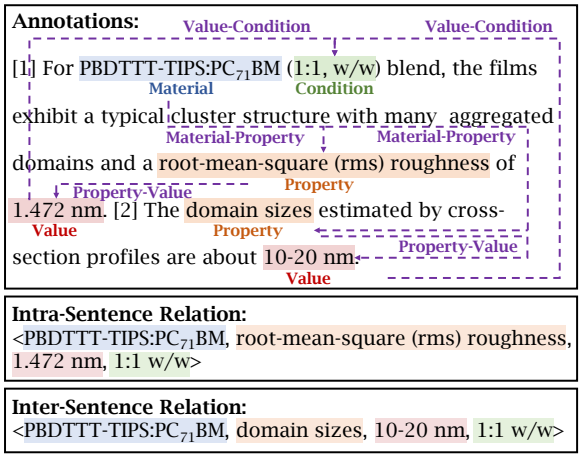


Figure 1: An example of entity and relation annotations in POLYIE from a material science paper (Shi et al., 2011), including entity mentions as well as intra-sentence and inter-sentence N-ary relations.

While SciIE has rapidly developed in domains such as biomedical science (Luan et al., 2018; Gábor et al., 2018; Jain et al., 2020; Hou et al., 2019; Jia et al., 2019), it has made limited progress in the material science domain. So far, there are only a handful of datasets for material information extraction. Some earlier works use ChemDataExtractor (Swain and Cole, 2016) to automatically generate datasets for battery materials (Huang and Cole, 2020) and temperatures (Court and Cole, 2018). More recent datasets are created manually for solid oxide fuel cells (Friedrich et al., 2020) and material science synthesis procedures (O'Gorman et al., 2021). However, none of these datasets cover polymer materials, which are an important class of organic materials that play critical and ubiquitous roles in our daily lives. Due to their versatile properties, polymer materials are being widely used in applications such as packaging, coating, energy

saving, and medical applications. As vast amounts of information on polymer development are being reported in literature data, there is a critical need for SciIE benchmarks and tools to harvest such information from the polymer literature.

To address this gap, we construct a dataset for extracting polymer property knowledge from unstructured literature data. Our dataset POLYIE is curated from 146 full-length polymer scientific articles, which are annotated by domain experts with named entities (i.e., materials, properties, values, conditions) as well as the N-ary relations among them (see Figure 1). POLYIE contains 41635 entity mentions and 4443 relations in total. It covers four different application domains of polymer materials: polymer solar cells (PSC), ring-opening polymerization (RP), polymer membranes (PM), and polymers in lithium-ion batteries (LB). This diversity of content enables the training of models with enhanced generalization capabilities. To the best of our knowledge, POLYIE is the first benchmark for SciIE from full-text polymer literature.

From the natural language processing perspective, extracting information for polymers on POLYIE introduces unique challenges for both named entity recognition and relation extraction:

Diverse Lexical Formats of Entities. Polymer-related entities often have different schemes of nomenclature, such as IUPAC names (*e.g.*, 'poly(3-hexylthiophene)'), abbreviations ('PDPPNBr'), trade names ('Styron'), common names ('ABS plastic'), and sample labels ('PE-HDPE-01'). In addition, the identification of polymers can also be achieved through the concatenation of homopolymer names with hyphens or slashes, and the inclusion of numerical values for the component ratios and molecular weights ('PVC-PS-PC-20/30/50-800000'). This diversity of nomenclature in literature poses a challenge for named entity recognition.

Variable-length and Cross-Sentence N-ary relations. Previous research on relation extraction has focused on either binary relations (Luan et al., 2018; Yao et al., 2019) or N-ary relations with a fixed number N (Jia et al., 2019; Jain et al., 2020; Zhuang et al., 2022). In contrast, many relations described in polymer literature are variable-length N-ary relations. This is because 1) the reported properties may be describing one or several materials; and 2) different properties can be measured under specific conditions. Furthermore, the elements in a relation tuple may span multiple sentences as shown in Figure 1.

We study six mainstay NER and five N-ary RE models on POLYIE in terms of their overall performance and sample efficiency. We find that the models based on domain-specific pre-trained models (e.g., MatSciBERT) yield better performance than other baselines. However, all the models struggle with accurately recognizing certain categories of named entities and inferring challenging varied-length N-ary relations. Moreover, our observations indicate that, under few-shot settings, the recently popular large language models (LLMs) demonstrate inferior performance than the other baselines on POLYIE, highlighting potential limitations in comprehending material science concepts.

Our main contributions are: (1) The first polymer information extraction dataset curated from 146 full-length articles for polymer named entity recognition and relation extraction. (2) Thorough evaluation of seven mainstream NER and five N-ary RE models on our curated dataset. (3) Analysis of the difficult cases and limitations of existing models, which we hope will enable future research on this challenging task from the NLP community.

2 Related Work

Material Science NLP Datasets. Earlier studies (Court and Cole, 2018) leverage tools such as ChemDataExtractor (Swain and Cole, 2016), ChemSpot (Rocktäschel et al., 2012), and ChemicalTagger (Hawizy et al., 2011) to perform NER annotation for dataset curation. For example, ChemDataExtractor is applied to generate datasets for Curie and Neel magnetic phase transition temperatures (Court and Cole, 2018) and magnetocaloric materials (Court et al., 2021). Besides, people also create expert-annotated datasets (Wang et al., 2021; Weston et al., 2019) for the extraction of non-value named entities (*e.g.*, material and property names) and their relationships. In recent years, there has been an uptick in efforts to include numerical values in datasets for further extraction, with several studies closely related to POLYIE: Friedrich et al. (2020) annotate a corpus of 45 open-access scholarly articles on solid oxide fuel cells, covering entity types of `materials`, `values`, and `devices`. Panapitiya et al. (2021) provide annotations of `CHEM`, `VALUE`, and `UNIT` on a set of papers on soluble materials. However, both only provide binary relations between pairs of entities, which is inadequate for describing more complex relations.

N-ary Relation Extraction. N-ary relations are size-N tuples that describe the factual relationship between N entities. In general domains, the MUC dataset (Chinchor, 1998) describes event participants in news articles. In the biomedical domain, the BioNLP Event Extraction Shared Task (Kim et al., 2009) and PubMed dataset (Jia et al., 2019) aim to extract biomedical events from biomedical text. In the machine learning domain, SciREX (Viswanathan et al., 2021; Jain et al., 2020; Zhuang et al., 2022) extracts N-ary relations in terms of `<Task, Dataset, Method, Metric>`. Different from these works' relations, the N-ary relations in POLYIE can have a varied number of named entities, which is more flexible in describing material knowledge but meanwhile introduces new challenges to RE. The closest work to POLYIE is drug-combo (Tiktinsky et al., 2022), which extracts variable-length combinations of different drugs. However, POLYIE and drug-combo are curated for two different domains, and the relations in POLYIE include numerical values.

3 The POLYIE Dataset

In this section, we describe the details of the POLYIE dataset. We first formulate the two information extraction tasks for polymer material literature in § 3.1. We then describe the data preprocessing and annotation procedures in § 3.2 and § 3.3, and finally present the statistics and characteristics of the POLYIE dataset in § 3.4.

3.1 Task Definition

POLYIE is curated for studying two key information extraction tasks on polymer literature data: (1) identifying relevant named entities, and (2) composing different entities to form N-ary relations.

Named Entity Recognition. Named Entity Recognition (NER) is the process of locating and classifying unstructured text phrases into predefined entity categories such as compound names, property names, etc. Given a sentence with n tokens $\mathbf{S} = (w_1, \cdots, w_n)$, a named entity mention is a span of tokens $\mathbf{e} = (w_i, \cdots, w_j)(0 \leq i \leq j \leq n)$ associated with an entity type. In POLYIE, we focus on NER for describing polymer material properties and include four important entity types: material name, property name, property value, and condition. An illustrative example can be found in Figure 1. Based on the BIO schema (Li et al., 2012), NER can be formulated as a sequence labeling task of assigning a sequence of labels $\mathbf{y} = (y_1, \cdots, y_n)$, each corresponding to a token in the input sentence.

Variable-Length N-ary Relation Extraction. Variable-length N-ary relation extraction (RE) refers to the process of identifying and extracting relationships between multiple entity mentions where the number of entities in the relationship can vary. Formally, given a list of k context sentences $\mathcal{C} = (S_1, \cdots, S_k)$ in one paragraph, let $\mathcal{E}$ be the set of entities appearing in $\mathcal{C}$ where each entity $e \in \mathcal{E}$ belongs to one of the four entity types described in the NER task. The relation extraction task aims to extract a set of m relations $\mathcal{R} = (r_1, \cdots, r_m)$ from $\mathcal{C}$. Each relation r_i is a tuple of entities $r_i = (e_1, \cdots, e_{N_i}), (1 \leq i \leq m)$ that describe their `<material, property, value, condition>` relations. Here, the number of entities N_i can be variable in $\mathcal{R}$ because: 1) the property value may correspond to several materials instead of one; and 2) the condition entity may be absent. Figure 1 illustrates this variable-length N-ary RE task.

3.2 Data Preparation

We curate POLYIE from 146 publicly available scientific papers, covering four different material science domains: polymer solar cells, ring-opening polymerization, polymer membranes, and lithium-ion batteries. These papers are sub-sampled from the corpus of 2.4 million material science articles described in Shetty et al. (2023). This corpus consists of papers published between 2000 to 2021 and is collected from 7 different material science publishers (Shetty and Ramprasad, 2021a,b). Keyword-based search was used to locate papers that span multiple application domains within polymers. The resulting dataset consists of 100 papers describing fullerene-acceptor polymer solar cells, 21 papers describing ring-opening polymerization, 20 describing lithium-ion batteries, and 5 describing polymer membranes. The text of these papers is parsed from the PDF of these papers using sciPDF [1] (a scientific parser based on GROBID (GRO, 2008–2023)) into utf-8 format. The incorrectly parsed units and symbols are corrected using regular expressions.

3.3 Data Annotation

The POLYIE dataset is annotated by two polymer science domain experts as well as three computer science graduate students who are trained by the

[1] https://github.com/titipata/scipdf_parser

polymer scientists. Both the NER and RE annotations are performed using the Doccano (Nakayama et al., 2018) platform, which is an open-source text annotation tool that facilitates visual annotation with a Web interface. Below, we detail the annotation schemes for the NER and RE tasks.

3.3.1 Annotating Named Entities

In POLYIE, we annotate mentions of named entities for four categories: material names, property names, property values, and conditions. Each mention is a continuous text span that specifies the actual name of an entity or its abbreviation. This is done by marking the entity mention on the Doccano platform with the corresponding entity type.

Compound Names (Material). Compound Name entities include text spans that refer to material objects. Only chemical mentions that could be associated with a chemical structure are annotated as Compound Names. They may be specified by a particular composition formula (*e.g.*, "4,9-di(2-octyldodecyl) aNDT"), a mention of chemical names (*e.g.*, "trimethyltin chloride"), or just an abbreviation (*e.g.*, "PaNDTDTFBT"). General chemical nouns (*e.g.*, "ionic liquids") are not considered.

Property Names (Property). We annotate the properties of chemical compounds as long as they can be measured qualitatively (*e.g.*, "toxicity" and "crystallinity") or quantitatively (*e.g.*, "open-circuit voltage", "decomposition temperature"). Corresponding abbreviations should also be annotated (*e.g.*, "PCE", "HOMO level").

Property Values (Value). We annotate the spans that can indicate the degree of qualitative properties (*e.g.*, "soluble to water") or describe numerical values with units for quantitative properties (*e.g.*, "$9.62 \times 10^{-5}\ \Omega^{-1}m^{-1}$", "5.14 ppm").

Conditions. In material science papers, the properties of materials can be constrained by quantitative modifiers, and we annotate them as conditions to distinguish them from normal property names and property values (*e.g.*, "room temperature", "frequency range 500 Hz – 3 MHz").

3.3.2 Variable-Length N-ary Relations

For RE, we annotate the N-ary relations between the named entities to capture their `<Material, Property, Value, Condition>` relations.

Primary Binary Relations. As Doccano and most other existing text annotation tools only support annotations for binary relations, we decompose the N-ary relation annotation task into simpler binary relation annotation and later aggregate them into full N-ary relations. We split an N-ary relation into multiple binary relations for annotation: `Material-Material` marks the relations between material names that constitute one material system; `Material-Property` identifies the relation between a material and its reported property name; `Property-Value` annotates the corresponding property name and value; and `Value-Condition` marks the property values measured under a specific condition.

Transforming Binary to N-ary Relations. We then transform all the binary relations with common involved entities to generate N-ary relations in the format of `<Material, (Material), Property, Value, (Condition)>`. We abandon all binary relations that cannot be combined with other binary relations, only maintaining the generated N-ary relations with $N > 2$.

3.3.3 Inter-Annotator Agreement

All documents in POLYIE are annotated by at least two annotators independently. If annotation conflicts arise across two annotators, a third annotator is then assigned to annotate the corresponding sentences independently. The final annotation is determined by majority voting.

We calculate the inter-annotator agreement in terms of Fleiss' Kappa (Fleiss, 1971). The Fleiss' Kappa for individual entity types is calculated by treating other entity types as negative samples. The results are shown in 1. The Fleiss' Kappas for `Material`, `Property`, and `Value` are all in the range of almost perfect agreement, while the corresponding value for `Condition` lies in the range of substantial agreement. For RE, we consider all annotated relations as subjects and treat categories as binary. The Fleiss' Kappa for RE is 0.67.

We also compute the average F1-score similar to Friedrich et al. (2020). The F1-score is calculated by treating one annotator as the gold standard and the other annotator as predicted. For the NER, spans and entity types have to exactly match. For RE, all entity mentions within the n-ary relation have to exactly match. The averaged F1-score for the NER and RE task is 0.89 and 0.84 respectively.

Overall	Material	Property	Value	Condition
0.86	0.88	0.82	0.88	0.71

Table 1: Fleiss' kappa for all annotators across all mentions and each entity type respectively.

	PSC	RP	LB	PM	All
documents	100	21	20	5	146
sentences	9,367	3,120	3,031	555	16,073
tokens	288,142	91,421	90,381	15,579	485,523
avg. tokens/doc.*	3,201.6	3,102.4	3,227.9	3,115.8	3,325.5
mentions	28,775	5,760	6,013	1,087	41,635
– `Material`	13,244	3,120	3,390	740	20,494
– `Property`	9,848	1,597	1,616	187	13,248
– `Value`	5,294	792	835	111	7,032
– `Condition`	364	150	167	21	702
entities	7,099	1,621	1,739	431	10,890
avg. mentions/doc.*	287.8	274.3	300.7	217.4	285.2
relations	3,084	592	615	152	4,443
– 3-ary	2,554	503	516	123	3,838
– 4-ary	388	89	99	29	605
– 5-ary	142	-	-	-	142
avg. relations/doc.*	30.8	28.2	30.8	30.4	30.4

*Avg. indicates average and doc. refers to document.

Table 2: POLYIE corpus statistics.

3.4 Data Analysis

Table 2 shows the key statistics for our corpus. POLYIE contains 41635 entity mentions and 4443 relations in all 146 fully annotated polymer material science literature. We quantitively analyze some key properties of POLYIE:

Statistics of Entities. For all the named entity mentions, the distribution of the four entity types `Material`, `Property`, `Value`, and `Condition` are 49.54%, 31.82%, 17.00%, and 1.70%, respectively. In total, those 41365 mentions describe 10890 distinct named entities for polymer materials.

Statistics of N-ary Relations. Among the 4443 relations on POLYIE, 86.38% are 3-ary; 13.62% are 4-ary; and 3.20% are 5-ary. Meanwhile, 26.65% of the relations are cross-sentence relations, while the rest are intra-sentence relations.

4 Modeling

In this section, we describe how we model the named entity recognition and N-ary relation extraction tasks on POLYIE.

Named Entity Recognition. We model the NER task as a sequence labeling problem and learn a neural sequence tagger, as shown in Figure 2. We study both the bi-directional LSTM-CRF (**BiLSTM-CRF**) (Ma and Hovy, 2016) model and BERT-based (Devlin et al., 2019) NER models for neural sequence tagging. We also study the performance of GPT-3.5 and GPT-4 on NER.

In BiLSTM-CRF, the input text is passed through an embedding layer to obtain token representations. These representations are then fed into a BiLSTM layer (Lample et al., 2016) to capture contextual information. The output of the BiLSTM layer is finally sent to a subsequence Conditional Random Field (CRF) layer (Lafferty et al., 2001) for sequence labeling. For the pre-trained language models (PLM), we study both **BERT**$_{\text{base}}$ (Devlin et al., 2019) and **RoBERTa** (Liu et al., 2019) for NER. We also include three domain-specific BERT models: **SciBERT** (Beltagy et al., 2019), **BioBERT** (Lee et al., 2020), and **MatSciBERT** (Gupta et al., 2022). All the NER models in the BERT family are fine-tuned for sequence labeling, by stacking a linear layer that maps the contextual token representations into the label space. In addition, we also evaluate LLMs' abilities in marking material science concepts. Following the existing work (Tang et al., 2023), We directly prompt **GPT-3.5-turbo** and **GPT-4** with few-shot exemplars to use special marks "@@" to annotate the boundaries and types of the named entities. Detailed explanations and examples of prompts are included in App. C.

Relation Extraction. For relation extraction, we evaluate the performances of the rule-based method, PLM-based models, and graph-based models. For the rule-based method, we leverage the assumption, **Proximity-Rule**, that relations are more likely to be formed with most proximitive entities. As illustrated in Figure 2, PLM-based models (such as **BERT-RE**) leverage the strong representation power of pre-trained language models on entities and employ simple aggregation techniques, such as concatenation and summation, to compose relation embeddings for further prediction. Example models in this category are state-of-the-art models **PURE** (Zhong and Chen, 2021), which inserts a special "entity marker" token around the entities in a candidate relation; and its variant **PURE-SUM** (Tiktinsky et al., 2022), which uses embedding summation for variable-length N-ary RE. We also study graph-based methods for N-ary RE, **DyGIE++** (Luan et al., 2019), which constructs a dynamic span graph from the input text, with entities as nodes and relations as edges to reason over multi-hop relations. For models based on LLMs like **GPT-3.5-turbo** and **GPT-4**, we randomly choose a subset of examples from the training set to serve as few-shot instances. These are then directly sent to the models as prompts to facilitate the relation extraction.

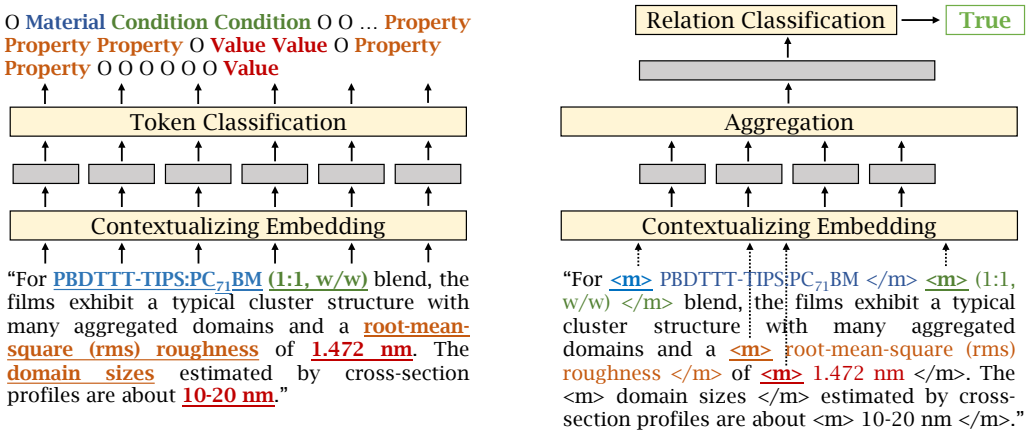


Figure 2: Model architecture for Named Entity Recognition (left) and N-ary Relation Extraction (right).

5 Experiments

5.1 Experimental Setup

Evaluation Protocol. We split the dataset into 123 training articles, 27 validation articles, and 27 test articles following a 70%/15%/15% ratio. The three sets do not have overlapping scientific documents. For NER, we report the entity-level precision, recall, and F-1 scores of each baseline for different entity categories, as well as the corresponding micro-average of these metrics. For RE, we report the precision, recall, and F-1 score.

Hyperparameters. For BiLSTM-CRF, we use one layer of BiLSTM layer with 256-dimensional hidden states and 128 embedding dimensionality. For the BERT-family NER models, we stack a linear layer with a hidden size of 128 on the BERT architecture for token classification. For all the NER and RE models, we use early stopping on the dev set for regularization. See App. B for details.

5.2 Main Results

Entity Mention Extraction. Table 3 shows the performance of different methods for the NER task on POLYIE. From the results, we make the following observations: (1) BERT-based models significantly outperform BiLSTM-CRF model with a 14.8% gain in micro average F1-score. This is because BERT-based models have been pre-trained on a large corpus of data, allowing them to possess more semantic knowledge than BiLSTM-CRF and to better understand the context. (2) Domain-specific BERT models achieve slightly better performance than the vanilla BERT due to the encoding of domain-specific knowledge. MatSciBERT, which is fine-tuned on a corpus of materials science articles, shows the best performance on almost all metrics. (3) Upon comparing the performance of different entity types, we find that it is challenging for all models to discriminate `Condition` entities from the other categories. We hypothesize that this is because `Conditions` are relatively rare in the training data, and the `Condition` entities could resemble property value entities.

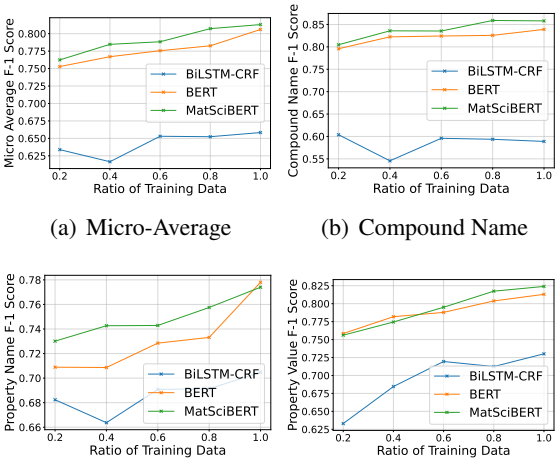
Relation Extraction. Table 4 shows the performance of different methods for the RE task on POLYIE, and we make the following observations: (1) Among all the models evaluated, the PURE-SUM model with MatSciBERT as the encoder achieves the highest F-1 score, indicating that MatSciBERT can better understand the context, and the summation operation is an appropriate aggregation method for variable-length N-ary relation extraction. (2) The rule-based approach exhibits inferior performance in comparison to most deep learning models, indicating that there are many cases that do not conform to the proximity rule, such as cross-sentence relations and parallel relations. (3) Interestingly, the BERT-RE model shows even worse performance than the rule-based method. Compared to PURE-based models, BERT-RE directly averages the embeddings of all tokens related to the relation. As tokens with similar types have similar representations, and N-ary relations are composed of certain entity-type elements, the averaging operation results in similar relation representations, ultimately leading to poor model performance. (4) As DyGIE++ is a model specifically designed for binary relation extraction, it can only determine the presence of N-ary relations by assessing the connectivity of arbitrary pairs of elements in the relationship. It thus has stricter judging criteria than the other methods, making its precision

Model	Material	Property	Value	Condition	Micro Average
BiLSTM-CRF	58.9 (68.4/51.7)	70.5 (75.4/66.2)	73.0 (74.6/71.5)	13.1 (36.4/8.0)	65.8 (72.4/60.4)
$\text{BERT}_{\text{base}}$	83.9 (84.0/83.8)	77.8 (81.1/74.7)	81.3 (83.9/79.0)	13.8 (16.2/12.0)	80.6 (82.4/78.8)
$\text{RoBERTa}_{\text{base}}$	85.4 (86.4/84.4)	76.2 (77.4/75.2)	81.8 (83.3/80.3)	12.5 (16.7/10.0)	80.7 (82.0/79.4)
SciBERT	85.6 (87.1/84.1)	74.6 (77.2/72.3)	81.9 (84.6/79.4)	11.3 (19.0/8.0)	80.3 (82.7/78.1)
BioBERT	85.1 (84.5/85.7)	76.9 (79.3/74.6)	82.6 (82.6/82.5)	15.2 (16.6/14.0)	81.0 (81.7/80.3)
MatSciBERT	85.8 (84.4/87.3)	77.4 (78.2/76.5)	82.4 (81.9/82.8)	11.4 (13.2/10.0)	81.3 (81.1/81.7)
GPT-3.5-Turbo	63.7 (61.4/67.2)	49.4 (47.5/52.5)	59.5 (86.6/45.9)	2.2 (17.5/1.3)	56.4 (58.8/54.1)
GPT-4	64.7 (57.6/75.2)	61.6 (52.2/76.6)	74.2 (67.1/84.2)	5.7 (8.5/4.8)	64.5 (56.5/75.1)

Table 3: Main NER results on the test dataset, presented as "F-1 Score (Precision/Recall)" in %. We offer scores under different metrics for each entity category and the overall micro-average performance.

Model	Precision	Recall	F-1 Score
Proximity-Rule	26.49	30.83	28.50
BERT-RE	12.06	40.28	18.57
DyGIE++	67.53	50.28	57.64
PURE	60.27	54.04	56.98
PURE-SUM (SciBERT)	42.86	82.50	56.41
PURE-SUM (MatSciBERT)	51.91	83.06	63.89
GPT-3.5-Turbo	16.37	34.27	21.73
GPT-4	37.82	54.16	44.06

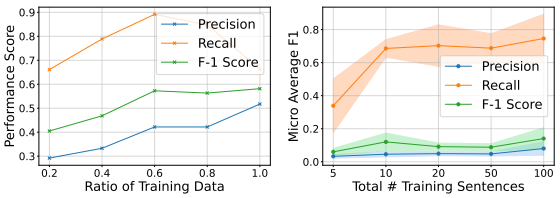
Table 4: Main RE results on the test dataset, presented as Precision, Recall, and F-1 Scores in %.



(a) Micro-Average (b) Compound Name

(c) Property Name (d) Property Value

Figure 3: Effect of training data size on NER task.



(a) Limit Data (b) Few-Shot Setting

Figure 4: Performances of PURE-SUM on RE task with limited training data and few-shot setting.

higher at the cost of lower recall.

Analysis on LLMs. LLMs such as GPT-3.5-turbo and GPT-4 exhibit worse performance compared to most baseline models on both NER and RE tasks. This discrepancy is likely due to the small proportion of polymer material science content in their pre-training corpus. When these models are directly prompted with few-shot examples, as opposed to being fine-tuned with training data, they receive less domain-specific information. This limitation hinders their ability to effectively understand and process concepts related to polymer material science. Potential updates on LLMs, like external tools (e.g., knowledge retriever) (Shi et al., 2023; Zhuang et al., 2023) or collaborations between LLMs and smaller pre-trained language models (Yu et al., 2023; Xu et al., 2023), may further boost the performance via injecting more domain-specific knowledge. Due to the poor performance obtained under the few-shot prompting setting and the high cost when fine-tuning LLMs, we recommend fine-tuning smaller domain-specific pre-trained language models, like MatSciBERT in Table 3 and PURE-SUM (MatSciBERT) in Table 4, to extract polymer material science entities and relations.

5.3 Impact of Data Size

We evaluate the NER model performance as a function of the amount of training data in Figure 3. Compared to BERT-based models, the performance of the BiLSTM-CRF model is consistently inferior, with only slight changes with varying sizes of training data. This trend demonstrates the superiority of language model pre-training stage, which allows BERT-family NER models to encode relevant knowledge for the downstream task. Comparing different BERT models, MatSciBERT consistently outperforms vanilla BERT by a slight margin, which reflects the benefit of developing domain-

Noise Types	Input Text
Interweaving Relations	The corresponding HOMO and **LUMO** energy levels for **PIDTT-TzTz** and PIDTT-TzTz-TT are (-5.24, **-3.21**) and (-5.34, -3.03) eV, respectively.
Partially Correct Relations	For example, OFETs made using a **porphyrin-diacetylene** polymer give **mobilities** of **1×10^{-7}cm^{2}V^{-1}s^{-1}** at room temperature and $2 \times 10^{-6} cm^2 V^{-1} s^{-1}$ at 175°C.
Inverted Sentences	For polymer PDTG-DPP, the thermal stability is even better than the Sibridged analogue, PDTS-DPP, and the T_d = 409°C of PDTG-IID is the same as the Si-bridged analogue, PDTS-IID.

Table 5: Examples incorrectly predicted by MatSciBERT. Entities highlighted in green indicate the gold N-ary relation in the input text. Predicted relations made by the model are shown in bold fonts. Red fonts represent the location of errors.

specific pre-trained language models.

Figure 4(a) shows the performance of the best RE model PURE-SUM as training data size varies. With more training data, the model's performance generally increases in all the metrics. However, after training on 60% data, the recall starts to decrease, while the other metrics still slightly increase. This is because the imbalance between positive and negative cases starts to influence the training, where models are more likely to predict relations as negative, making the false negative cases increase and the recall decrease. Additional details about the impact of data size on RE tasks can be seen in Appendix D.

5.4 Error Analysis

We analyze the key error types of the $\text{BERT}_{\text{base}}$ NER model by drawing its confusion matrix on the test set, as shown in Figure 5. The confusion matrix shows that the majority of entities are correctly predicted as their gold label, with the exception of `Condition` entities. The limited number of training samples containing `Condition` entities makes it difficult for the model to distinguish them from other irrelevant entities (labeled "O"). Additionally, the resemblance between `Condition` and `Property Value` entities often results in incorrect predictions between them.

For RE, Table 5 illustrates the major error types made by the PURE-SUM model, including: (1) Interweaving or parallel relations in the text present a significant challenge for models in understanding the alignment between multiple sets of entities; (2) The task of flexible-length N-ary relation extraction is challenging, and errors often occur when encountering relations that cover more entities (e.g., determining whether to include the `Condition` in the prediction); (3) The last type of error frequently arises when the sentence organization is atypical,

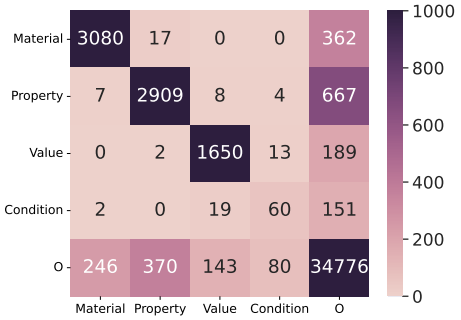


Figure 5: The confusion matrix of BERT on NER task.

including sentences written in the passive voice.

6 Conclusion

We have curated a new dataset PolyIE for named entity recognition and N-ary relation extraction from polymer scientific literature. PolyIE covers thousands of `<Material, Property, Value, Condition>` relations curated from 146 full polymer articles. We have evaluated mainstay NER and RE models on PolyIE and analyzed their performance and error cases. In addition, we have also tested the performance of the strongest LLMs, GPT-3.5 and GPT-4, on PolyIE. We found that even state-of-the-art models, either domain-specific pre-trained language models or most advanced LLMs, can struggle with hard NER and RE cases. Through error analysis, we found that such difficulties arise from the diverse lexical formats and ambiguity of polymer named entities and also variable-length and cross-sentence N-ary relations. Our work contributes the first polymer scientific information extraction dataset as well as insights into this dataset. We hope PolyIE will serve as a useful resource that will and attract more research efforts from the NLP community to push the boundary of this task.

Limitations

One limitation of POLYIE is that we have annotated only the text modality of the polymer literature corpus. While tables and figures are not included in POLYIE, they are two important modalities that contain a considerable amount of information about polymer properties. It will be interesting to explore annotation schemes that can extend POLYIE to include tables and figures and enable multi-modal information extraction jointly from text, tables, and figures. In addition, POLYIE currently covers four application subdomains for polymer materials. In the future, POLYIE can benefit from including more sub-domains for polymers, as well as scientific publications for other organic materials. Such extensions will not only make POLYIE more comprehensive for studying polymer information extraction, but also allow it to be used to study cross-domain transfer of different information extraction models.

References

2008–2023. Grobid. https://github.com/kermitt2/grobid.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.

Keith T Butler, Daniel W Davies, Hugh Cartwright, Olexandr Isayev, and Aron Walsh. 2018. Machine learning for molecular and materials science. *Nature*, 559(7715):547–555.

Nancy A Chinchor. 1998. Overview of muc-7/met-2. Technical report.

Callum J Court and Jacqueline M Cole. 2018. Auto-generated materials database of curie and néel temperatures via semi-supervised relationship extraction. *Scientific data*, 5(1):1–12.

Callum J Court, Apoorv Jain, and Jacqueline M Cole. 2021. Inverse design of materials that exhibit the magnetocaloric effect by text-mining of the scientific literature and generative deep learning. *Chemistry of Materials*, 33(18):7217–7231.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.

Annemarie Friedrich, Heike Adel, Federico Tomazic, Johannes Hingerl, Renou Benteau, Anika Marusczyk, and Lukas Lange. 2020. The SOFC-exp corpus and neural approaches to information extraction in the materials science domain. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1255–1268, Online. Association for Computational Linguistics.

Kata Gábor, Davide Buscaldi, Anne-Kathrin Schumann, Behrang QasemiZadeh, Haifa Zargayouna, and Thierry Charnois. 2018. Semeval-2018 task 7: Semantic relation extraction and classification in scientific papers. In *SemEval*, pages 679–688.

Tanishq Gupta, Mohd Zaki, NM Krishnan, et al. 2022. Matscibert: A materials domain language model for text mining and information extraction. *npj Computational Materials*, 8(1):1–11.

Lezan Hawizy, David M Jessop, Nico Adams, and Peter Murray-Rust. 2011. Chemicaltagger: A tool for semantic text-mining in chemistry. *Journal of cheminformatics*, 3(1):1–13.

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

Shu Huang and Jacqueline M Cole. 2020. A database of battery materials auto-generated using chemdataextractor. *Scientific Data*, 7(1):1–13.

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

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

Jin-Dong Kim, Tomoko Ohta, Sampo Pyysalo, Yoshinobu Kano, and Jun'ichi Tsujii. 2009. Overview of BioNLP'09 shared task on event extraction. In *Proceedings of the BioNLP 2009 Workshop Companion Volume for Shared Task*, pages 1–9, Boulder, Colorado. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

John D Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California. Association for Computational Linguistics.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Qi Li, Haibo Li, Heng Ji, Wen Wang, Jing Zheng, and Fei Huang. 2012. Joint bilingual name tagging for parallel corpora. In *21st ACM International Conference on Information and Knowledge Management, CIKM 2012*, pages 1727–1731.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3219–3232, Brussels, Belgium. Association for Computational Linguistics.

Yi Luan, Dave Wadden, Luheng He, Amy Shah, Mari Ostendorf, and Hannaneh Hajishirzi. 2019. A general framework for information extraction using dynamic span graphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3036–3046, Minneapolis, Minnesota. Association for Computational Linguistics.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074, Berlin, Germany. Association for Computational Linguistics.

Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. 2021. On the stability of fine-tuning {bert}: Misconceptions, explanations, and strong baselines. In *International Conference on Learning Representations*.

Hiroki Nakayama, Takahiro Kubo, Junya Kamura, Yasufumi Taniguchi, and Xu Liang. 2018. doccano: Text annotation tool for human. Software available from https://github.com/doccano/doccano.

Tim O'Gorman, Zach Jensen, Sheshera Mysore, Kevin Huang, Rubayyat Mahbub, Elsa Olivetti, and Andrew McCallum. 2021. MS-mentions: Consistently annotating entity mentions in materials science procedural text. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1337–1352, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Gihan Panapitiya, Fred Parks, Jonathan Sepulveda, and Emily Saldanha. 2021. Extracting material property measurement data from scientific articles. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5393–5402, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Tim Rocktäschel, Michael Weidlich, and Ulf Leser. 2012. Chemspot: a hybrid system for chemical named entity recognition. *Bioinformatics*, 28(12):1633–1640.

Jonathan Schmidt, Mário RG Marques, Silvana Botti, and Miguel AL Marques. 2019. Recent advances and applications of machine learning in solid-state materials science. *npj Computational Materials*, 5(1):1–36.

Pranav Shetty, Arunkumar Chitteth Rajan, Chris Kuenneth, Sonakshi Gupta, Lakshmi Prerana Panchumarti, Lauren Holm, Chao Zhang, and Rampi Ramprasad. 2023. A general-purpose material property data extraction pipeline from large polymer corpora using natural language processing. *npj Computational Materials*, 9(1):52.

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

Pranav Shetty and Rampi Ramprasad. 2021b. Machine-guided polymer knowledge extraction using natural language processing: The example of named entity normalization. *Journal of Chemical Information and Modeling*, 61(11):5377–5385.

Qinqin Shi, Haijun Fan, Yao Liu, Wenping Hu, Yongfang Li, and Xiaowei Zhan. 2011. A copolymer of benzodithiophene with tips side chains for enhanced photovoltaic performance. *Macromolecules*, 44(23):9173–9179.

Wenqi Shi, Yuchen Zhuang, Yuanda Zhu, Henry Iwinski, Michael Wattenbarger, and May Dongmei Wang. 2023. Retrieval-augmented large language models for adolescent idiopathic scoliosis patients in shared decision-making. In *Proceedings of the 14th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, pages 1–10.

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

Ruixiang Tang, Xiaotian Han, Xiaoqian Jiang, and Xia Hu. 2023. Does synthetic data generation of llms help clinical text mining? *arXiv preprint arXiv:2303.04360*.

Aryeh Tiktinsky, Vijay Viswanathan, Danna Niezni, Dana Meron Azagury, Yosi Shamay, Hillel Taub-Tabib, Tom Hope, and Yoav Goldberg. 2022. A dataset for n-ary relation extraction of drug combinations. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3190–3203, Seattle, United States. Association for Computational Linguistics.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Vijay Viswanathan, Graham Neubig, and Pengfei Liu. 2021. CitationIE: Leveraging the citation graph for scientific information extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 719–731, Online. Association for Computational Linguistics.

Xuan Wang, Vivian Hu, Xiangchen Song, Shweta Garg, Jinfeng Xiao, and Jiawei Han. 2021. ChemNER: Fine-grained chemistry named entity recognition with ontology-guided distant supervision. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5227–5240, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Leigh Weston, Vahe Tshitoyan, John Dagdelen, Olga Kononova, Amalie Trewartha, Kristin A Persson, Gerbrand Ceder, and Anubhav Jain. 2019. Named entity recognition and normalization applied to large-scale information extraction from the materials science literature. *Journal of chemical information and modeling*, 59(9):3692–3702.

Ran Xu, Hejie Cui, Yue Yu, Xuan Kan, Wenqi Shi, Yuchen Zhuang, Wei Jin, Joyce Ho, and Carl Yang. 2023. Knowledge-infused prompting: Assessing and advancing clinical text data generation with large language models. *arXiv preprint arXiv:2311.00287*.

Yuan Yao, Deming Ye, Peng Li, Xu Han, Yankai Lin, Zhenghao Liu, Zhiyuan Liu, Lixin Huang, Jie Zhou, and Maosong Sun. 2019. DocRED: A large-scale document-level relation extraction dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 764–777, Florence, Italy. Association for Computational Linguistics.

Yue Yu, Yuchen Zhuang, Jieyu Zhang, Yu Meng, Alexander Ratner, Ranjay Krishna, Jiaming Shen, and Chao Zhang. 2023. Large language model as attributed training data generator: A tale of diversity and bias. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

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

Yuchen Zhuang, Yinghao Li, Jerry Junyang Cheung, Yue Yu, Yingjun Mou, Xiang Chen, Le Song, and Chao Zhang. 2022. Resel: N-ary relation extraction from scientific text and tables by learning to retrieve and select. *arXiv preprint arXiv:2210.14427*.

Yuchen Zhuang, Yue Yu, Kuan Wang, Haotian Sun, and Chao Zhang. 2023. Toolqa: A dataset for llm question answering with external tools. *arXiv preprint arXiv:2306.13304*.

A T-SNE Visualization of Entity Embeddings

Figure 6 shows t-SNE (Van der Maaten and Hinton, 2008) visualization of entity embeddings generated by $\text{BERT}_{\text{base}}$, SciBERT and MatSciBERT. Compared with all the visualization of different entity embeddings, we can observe that pre-training on a more similar domain of corpus to fine-tuning corpus will make model generate high-quality embeddings. From the figures, we can easily observe that MatSciBERT embeddings of the same entity type are more clustered than those of $\text{BERT}_{\text{base}}$, which is also consistent with what we observe from the quantitive results.

B Implementation Details

All the NER and RE models are trained with the Adam optimizer (Kingma and Ba, 2014), with different learning rate: The BiLSTM-CRF model is trained with a learning rate of 0.005 and batch size of 64; While fine-tuning the BERT-family NER models, we select the learning rate of $3e-4$; For relation extraction, instances with lengths exceeding 300 are broken into several shorter segments, without cutting off relations, and the models are trained with a learning rate of $2e-4$ and a batch size of 8. All experiments are conducted on *CPU*: Intel(R) Core(TM) i7-5930K CPU @ 3.50GHz and *GPU*: NVIDIA GeForce RTX A5000 GPUs using python 3.8 and Pytorch 1.10.

C Implementation Details of LLMs

We conduct experiments on Azure OpenAI platform, with GPT-3.5-turbo and GPT-4 in 0613 version. We set the temperature as 0 to obtain a stable and faithful evaluation of the LLMs' results. Following the existing work (Tang et al., 2023), we have 4 components in our NER prompt: general instruction, annotation guideline, output indicator, and few-shot exemplars. (1) The general instruction part specifies the objective of the LLM to mark the polymer material science entities or relations. (2) The annotation guideline is to provide additional explanation and guidelines for the LLM to follow when annotating different types of entities and relations. (3) The output indicator specifies the output format of the LLM. (4) The few-shot exemplars allow LLM to form a more cohesive understanding of previous instructions.

The NER and RE prompts are presented below.

Listing 1: NER prompt.

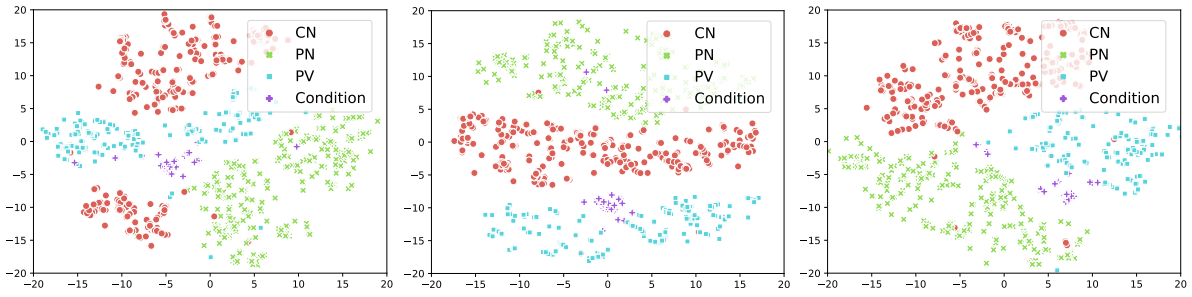
```
As a proficient linguist, your objective is to identify and label specific entities within a provided paragraph. These entities include chemical names (CN), property names (PN), property values (PV), and conditions (Condition). Chemical names, polymer material names and their abstractions are entities. Polymer material names might contain multiple chemical names within it, label them as a single entity. Abstractions of property names are also considered entities. Property values contain both the number and the unit. To represent recognized named entities in the output text, enclose them within special symbols '@', followed by their respective types '(CN)', '(PN)', or '(PV)' before the ending '@'. The remaining text should remain unchanged.
```

Listing 2: RE prompt.

```
As a skilled linguist, your mission is to analyze a provided paragraph that contains four distinct types of entities: Chemical Names (CN), Property Names (PN), Property Values (PV), and Conditions (Condition). Each of these entities is enclosed within "@" symbols, with their entity type specified in brackets before the closing "@". Your objective is to identify and extract relationships among these entities, and then present them in one of two possible formats: (Chemical Names, Property Names, Property Values, Condition) or (Chemical Names, Property Names, Property Values). Please only establish relationships using the provided entities, and only provide a list of the extracted relations. Below are some examples:
```

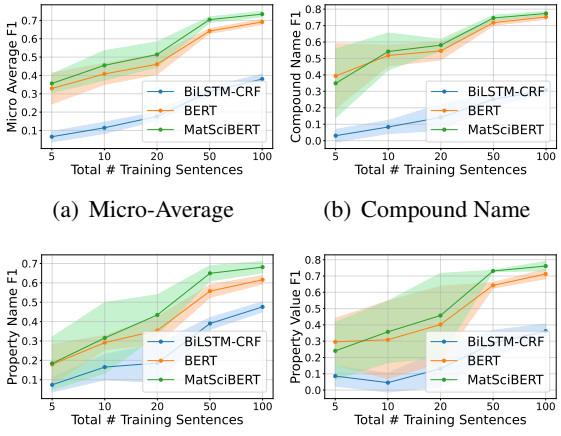
D Few-Shot Learning

Figure 7 shows the performance of different NER models under few-shot settings. We can see BERT-based NER models consistently outperform BiLSTM-CRF models by large margins. However, the variances of such BERT-based NER models are also much larger. This is likely due to the different quality and representativeness of the training samples and the capacity of pre-trained language models. The MatSciBERT model, for instance, has already captured a significant amount of domain knowledge during pre-training. When it is fed with critical cases during the fine-tuning stage, it can quickly adapt such knowledge to fine-tuning, resulting in high-quality decision boundaries on



(a) Original $\text{BERT}_{\text{base}}$ (b) SciBERT (c) MatSciBERT

Figure 6: t-SNE visualization of entity embeddings generated by BERT, SciBERT, and MatSciBERT.



(a) Micro-Average (b) Compound Name

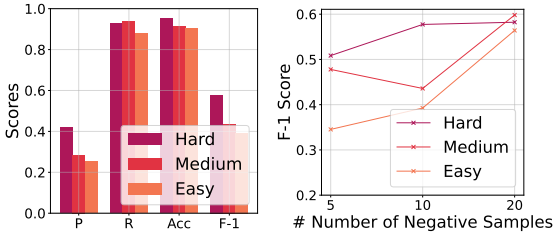
(c) Property Name (d) Property Value

Figure 7: Effect of the few-shot learning on NER task.

the corpus. However, if the training samples are of poor quality and not representative, the model's performance can be limited. Such instability of BERT-based fine-tuning is also observed on GLUE (Mosbach et al., 2021).

E Impact of Negative Sampling in Training

As the RE models are trained with negative sampling, we investigate the impact of negative samples during the training process. We study three ways to create negative samples from existing relations, by corrupting entities with other irrelevant entities of the same type in the context sentences. (1) Easy: all possible random corruptions; (2) Medium: single or double element corruption; and (3) Hard: only single-element corruption. Figure 8(a) shows the results when training with different negative sampling policies, with a fixed $k = 10$. We find that the hard negative sampling strategy achieves superior performance, suggesting that using hard negative cases can help the model learn better decision boundaries. In Figure 8(b), we also evaluate the model performances when varying the number of negative samples k from 5 to 20. The trend shows that $k = 20$ achieves the best performances with all different negative sampling strategies.



(a) Quality (b) Amount

Figure 8: Effect of the quality and amount of negative samples during training in N-ary relation extraction.

F Annotation Guidance

In this section, we will introduce the annotation guidance. There are 4 types of entities that should be annotated: Chemical Compound, Property Name, Property Value, and Condition.

F.1 Chemical Compound

• Only chemical nouns that can be associated with a specific structure should be labeled as Chemical Compounds: *e.g.*, "4,9-di(2-octyldodecyl) aNDT", "trimethyltin chloride";

• Abbreviation of the chemical nouns should also be labeled as Chemical Compounds as long as it can be associated with a specific structure: *e.g.*, "PaNDTDTFBT";

• General chemical concepts (non-structural or non-specific chemical nouns), adjectives, verbs, and other terms that can not be associated directly with a chemical structure should not be annotated: *e.g.*, "polymer", "conjugated polymers" should not be annotated;

• Spans: Spans of Chemical Compounds should not contain leading or trailing spaces. If the abbreviation of Chemical Compound appears inside brackets, the brackets should not be included in the annotation.

F.2 Property Name

• Properties of chemical compounds should be annotated as long as they can be measured qualitatively (such as toxicity and crystallinity) or quantitatively (with a unit and a value). Property Names that occur without a corresponding value should also be annotated: *e.g.*, "Hole mobility", "Open-circuit voltage", "decomposition temperature", "conductivity", "toxicity";
• Abbreviations of Property Names should be annotated: "PCE", "HOMO level", "LUMO level";
• Laboratory methods should not be annotated as Property Names: "Titration", "Cyclic voltammetry" should not be annotated as Property Names;
• Spans: Spans of Property Names should not contain leading or trailing spaces.

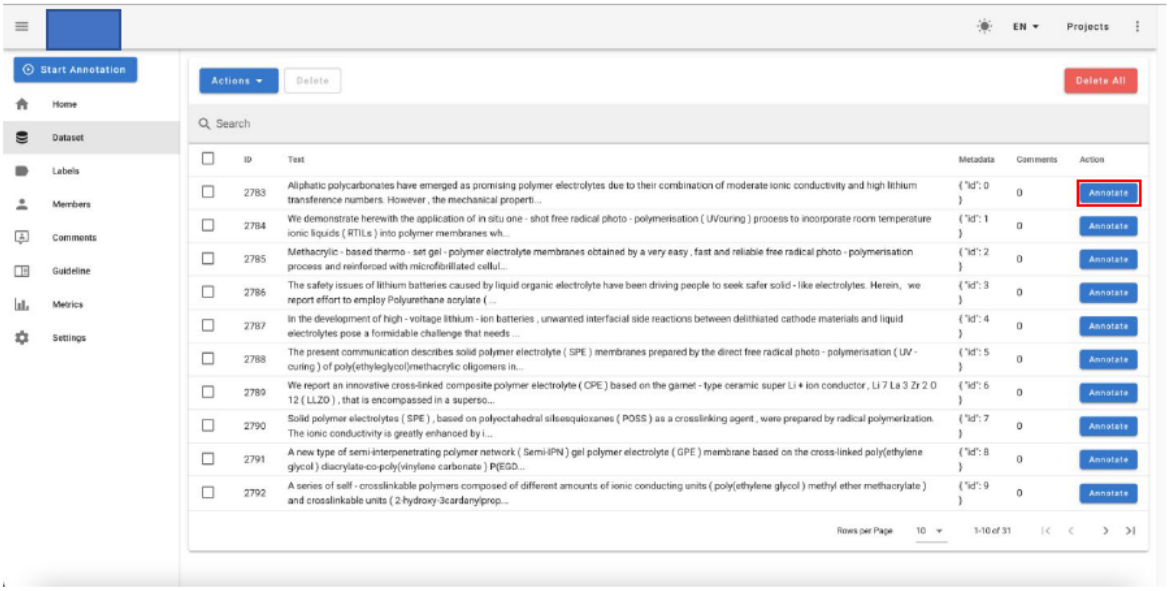
F.3 Property Value

• Both quantitative and qualitative Property Values should be annotated;
• Do not annotate overly vague adjectives;
• Spans of Property Values should not contain leading or trailing spaces. Property Value and its units should be contained as a single span. Ranges of Property Value should be contained as a single span.

F.4 Condition

• Only quantitative modifiers that constrain the numerical Property Value should be annotated as Conditions;
• Spans of Conditions should not contain leading or trailing spaces.

The screenshots of the official annotation guidance shared with all the annotators are listed in Figure 9 and Figure 10.



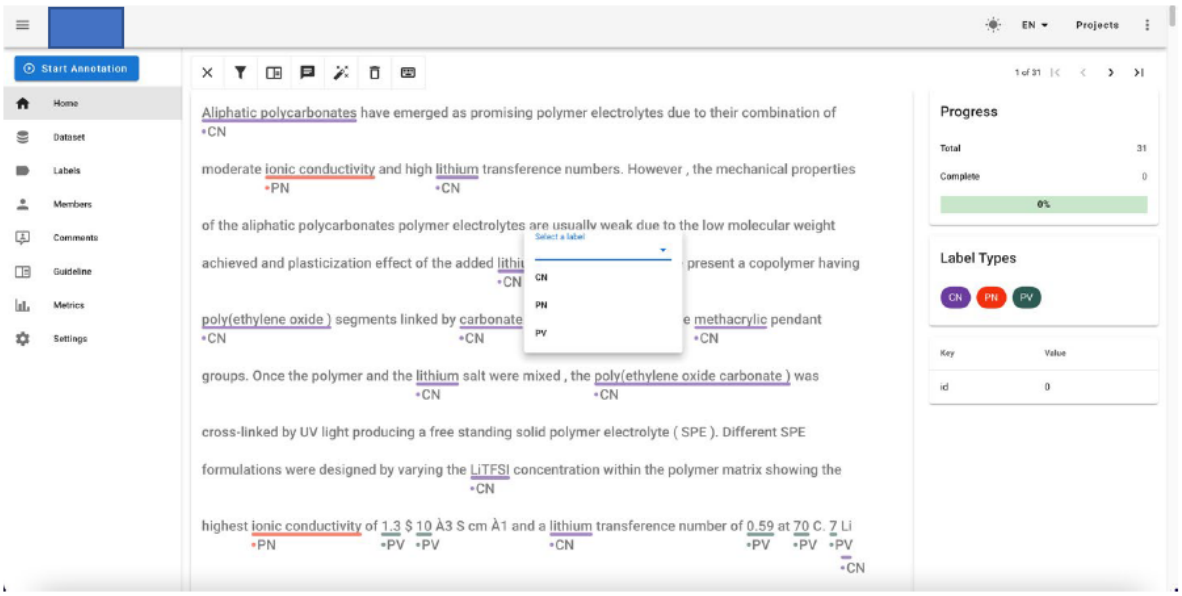
To start annotating documents, simply click on the "Annotate" Button on the right of screen.

Labels

Figure 9: Overview of documents to annotate on the annotation platform.

Annotate Data

For entity annotations, you first need to select the range of words from the corpus and then select the corresponding entity label for it:



To remove annotations, you can click on an existing annotation and click the little cross highlighted below:

Figure 10: Instructions on assigning pre-defined labels to named entities.