SciNER: Extracting Named Entities From Scientific Literature *

Zhi Hong[1], Roselyne Tchoua[1], Kyle Chard[1], and Ian Foster[1,2]

[1] University of Chicago, Chicago IL 60637, USA
{hongzhi, roselyne, chard, foster}@uchicago.edu
[2] Argonne National Lab

Abstract. The automated extraction of claims from scientific papers via computer is difficult due to the ambiguity and variability inherent in natural language. Even apparently simple tasks, such as isolating reported values for physical quantities (e.g., "the melting point of X is Y") can be complicated by such factors as domain-specific conventions about how named entities (the X in the example) are referenced. Although there are domain-specific toolkits that can handle such complications in certain areas, a generalizable, adaptable model for scientific texts is still lacking. As a first step towards automating this process, we present a generalizable neural network model, SciNER, for recognizing scientific entities in free text. Based on bidirectional LSTM networks, our model combines word embeddings, subword embeddings, and external knowledge (from DBpedia) to boost its accuracy. Experiments show that our model outperforms a leading domain-specific extraction toolkit by up to 50%, as measured by F1 score, while also being easily adapted to new domains.

Keywords: Named Entity Recognition · LSTM · word embeddings.

1 Introduction

The scholarly model has long relied on publication as a means of documenting and disseminating results. As such, scientific papers often contain the ultimate source of truth about a particular scientific entity, for example how it was produced, analyzed, and processed. Unfortunately, this approach to dissemination has obvious shortcommings, most notably that data and results are inaccessible to machines due to their esoteric encoding. Further, given the enormous number of publications—estimated to be over 2.5 million every year [31] and exponentially growing [12]—it is increasingly infeasible for individual researchers to locate important data in publications. A researcher might have to read dozens if not hundreds of papers just to get a rough idea of the state-of-the-art research

* This work was supported in part by NIST contract 60NANB15D077, the Center for Hierarchical Materials Design, and DOE contract DE-AC02-06CH11357, and by computer resources provided by Jetstream [26].

in an area, and even after they have done so, there is no guarantee that they have not missed important results.

One potential solution to this publication deluge is to extract scientific facts from free text articles and then to store these facts in structured searchable databases. A researcher might then be able to issue queries such as "`SELECT (*) from polymers WHERE glass_transition_temperature >= 100`" to obtain data of interest when needed. However, while such databases can avoid redundant effort, building them in the first place requires extensive manual extraction and curation effort, that is furthermore complicated by the difficulty and uncertainty of extracting scientific facts from text [16].

Crowdsourcing provides one method for performing manual, human-oriented tasks in an efficient and cost-effective manner. [3, 4, 7]. However, the expertise required to extract scientific facts makes crowdsourcing impractical. The few existing scientific databases and repositories, such as the Japanese Polymer Data Handbook [19], are curated by domain experts and thus costly to maintain; as a result, they quickly become out of date.

Since it is infeasible to rely solely on humans to extract scientific facts from publications, automatic approaches are needed to address the increasing rate of publication. A first problem to be tackled when extracting facts from publications is the identification of scientific entities (e.g., a chemical, sample, or anatomical region): a problem that we call scientific Named Entity Recognition (NER).

Considerable progress has been made in machine learning (ML) and natural language processing (NLP) in the last decade, with state-of-the-art models outperforming humans in various tasks [9]. However, most such efforts are centered on day-to-day language corpora, such as news articles, Twitter posts, and online product reviews. Little attention has been paid to the unique challenges associated with understanding scientific texts, such as idiosyncratic writing styles, specialized article organizations, and domain-specific vocabularies that are not common in other texts. A previous study of the biomedical literature from PubMed shows that the quality of machine learning models depends on the training corpora, model architectures, and hyper-parameters used [6]. Hence, to obtain high quality scientific NER, models must be trained on corpora from the same domain.

In this paper, we present SciNER, a NER model that is specifically designed for recognizing named entities in a scientific context. We show that this model is generalizable and can be trained on and applied to different domains. Our primary contributions are:

1. Development of SciNER using bidirectional LSTM networks and conditional random fields.
2. Integration of several word embedding models and lexicons from DBpedia as an external source of knowledge to boost learning performance.
3. Evaluation of the accuracy of SciNER on two different scientific NER problems and comparison with a domain-specific, state-of-the-art toolkit.

The paper is organized as follows. In Section 2 we discuss the specific problem we aim to solve and introduce the architecture of our proposed model. In Section 3 we describe the word embedding and lexicon features used in our model. In Section 4 we evaluate the accuracy of SciNER on two different scientific named entity recognition problems. Finally, we explore related work in Section 5 and summarize our approach in Section 6.

2 The SciNER Model

We focus on the task of identifying scientific named entities in scientific publications. In this section we first define our extraction problem and then outline the Bidirectional LSTM and Lexicon-infused LSTM models used in SciNER.

2.1 Problem Definition

Given a publication, comprised of sections containing natural language text, our task is to identify scientific named entities of interest. For example, in materials science publications about polymers, we want to identify polymer names, such as "*polystyrene*" in the following:

> "We measured the viscosity of unentangled, short-chain **polystyrene** films on silicon at different temperatures and found that ..." [33]

In social science publications, the task is slightly different. Here researchers explore hypotheses and make assertions based on analysis of known datasets but the dataset are often not cited like other artifacts. The authors do not always use the full names when referencing datasets in the natural language text of the paper. When given the following paragraph from a social science paper, we want to extract the boldfaced words.

> "By analyzing data from 3279 individuals who participated in the **Longitudinal Study of American Youth**, this study examines ..." [25]

While these two examples are from different domains, their named entity extraction tasks are similar: in each case, we want a model that, without altering its structure, can adapt to the task of identifying a certain class of scientific named entity (polymers and social science datasets, respectively) in scientific text.

2.2 The Basic Model: Bidirectional LSTM with CRF

The Long-Short Term Memory (LSTM) network has shown promise for various natural language processing tasks. LSTMs, like the human mind, can retain knowledge of previous tokens (i.e., words or punctuation marks) and use them to better understand the meaning of the next token in its context. SciNER aims

to build upon this prior work by adapting LSTM-based approaches to the specific challenges associated with scientific NER.

Fig. 1 shows an overview of the structure of a basic LSTM network model. One major advantage that LSTM has over other traditional methods is that it does not require any specific (and often manually selected) features. A common approach for applying LSTMs to NER tasks is to assign labels that indicate the entities to be extracted. For example when using beginning-inside-outside (BIO) labeling [22], a label "B," "I," or "O" is assigned to each token in the training corpus. "B" is assigned to the first word in a named entity or a single-word named entity, "I" marks a subsequent word in a multi-word entity, and all other (non-named entity) words are given the label "O."

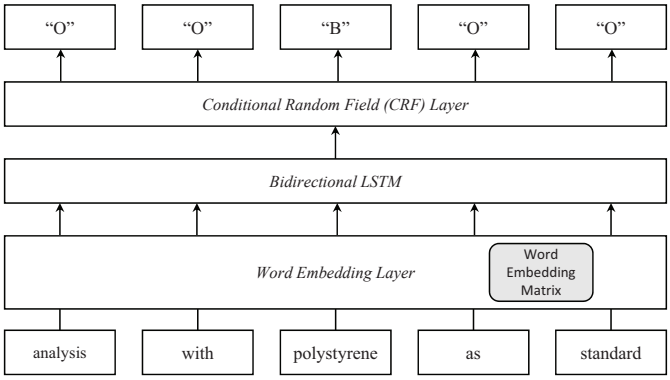

Fig. 1. Overall structure of the proposed neural network model.

For NER tasks, LSTM reads the input in one pass and assigns a BIO label to each word. However, in some cases, it is hard to tell whether a word is part of a named entity by looking at only the words *preceding* it in the sentence. For example, upon seeing the word "New" in a sentence, it is difficult to determine whether it should be given the label "B" or "O." However, if the next word is "York," then we can determine that it is likely a named entity. Bidirectional LSTM (Bi-LSTM) is used in natural language processing to address this need to exploit information about the words that come before *and* after a given word.

The Bi-LSTM network predicts a label for every word. This, however, means that the network has no awareness of the validity of the label sequence that it generates. Thus, it may output sequences (e.g., "OIO") that are invalid under the BIO labeling scheme. To penalize such invalid label sequences, we add a Conditional Random Field (CRF) layer on top of the Bi-LSTM network.

2.3 A Lexicon-infused Bi-LSTM Network

The Bi-LSTM model can learn only about the order of the words. For example, seeing "*poly(vinyl methyl ether)*" in the training data would not indicate that the

unseen text "*poly(ethylene glycol)*" is also a polymer name. However, knowledge that both *vinyl methyl ether* and *ethylene glycol* are chemical compounds and they both follow the pattern "*poly([chemical compounds])*" could be used to determine that both are in fact polymer names.

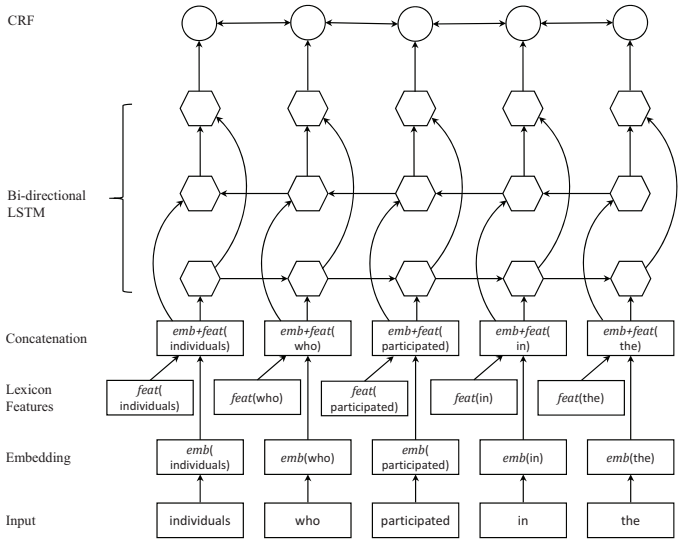

Fig. 2. The model with external lexicon knowledge.

To further improve the accuracy of our model, we introduce an external source of knowledge, by mapping words to classes obtained from DBpedia. Simply put, DBpedia is a structured version of Wikipedia, which consists of 4.2 million entries that are categorized in 774 distinct classes such as people, organization, location, and chemical compound [1]. Previous studies have explored the use of DBpedia classes for standard NER tasks such as CoNLL [6]; however, to the best of our knowledge such approaches have not been applied to scientific NER problems. Encoding lexicon features and feeding them to the LSTM gives the network more opportunities to capture the internal structure of named entities. Thus the network may be able to recognize "*poly(ethylene glycol)*" as a polymer name even when it has not been seen by the network before.

To encode the knowledge from the external lexicon, we use the same BIO labels as described above, as shown in Table 1. When overlaps occur between classes, as in this example (or within a single class, as in the case of "US Bureau of Labor Statistics" and "Bureau of Labor Statistics," both present in the Organization lexicon), we choose the longest match. We add a concatenation layer to combine these the word embeddings with the lexicon features and feed the concatenated vector to the LSTM, as illustrated in Fig. 2.

Table 1. An example showing how external knowledge from DBpedia is encoded for the Lexicon-infused Bi-LSTM Network. The first line is a sample sentence from a paper. The next two lines shows how BIO labels are assigned to the words that match entries in the Location or Organization categories in DBpedia.

	The	U.S.	Bureau	of	Labor	Statistics	Industry	Injury	and	Illness	Data	reveals	that	···
LOC	O	**B**	O	O	O	O	O	O	O	O	O	O	O	···
ORG	O	**B**	**I**	**I**	**I**	**I**	O	O	O	O	O	O	O	···

3 Features

SciNER's LSTM models can consume input features beyond the word labels described above. To capture the context and meaning of scientific text we add various word embedding models and lexicon features as input to the models.

3.1 Word Embeddings

The LSTM models require that input sentences are represented as numerical data (i.e., vectors). The simplest way to convert words to vectors is one-hot encoding, but it is not ideal because significant syntactic and semantic properties of the word would be lost. One approach for capturing these properties is through word embeddings. In the remainder of this subsection, we introduce and compare several different word embedding models. In Section 4 we explore the performance of SciNER when using each of these word embedding models.

Randomly Initialized Trainable Word Embeddings As shown in Fig. 1, the input layer is connected to the Bi-LSTM layer via the Embedding layer. By default, the weights of this layer, i.e., the word embedding matrix, is randomly initialized and is trained along with the whole neural network. No special algorithm is applied in this case, the word embedding matrix is treated in the same way as any other trainable parameters in the network.

Continuous Bag-Of-Words Model (CBOW) CBOW [17] is a popular method for training word embedding models. The core idea is that the semantic and syntactic information of a word can be determined (or represented) based on the context in which the word appears. Hence comes the idea of a fixed-sized window around the center word. Reusing the same sentence as in Fig. 1 ("... analysis with polystyrene as standard ..."), If the center word is "polystyrene" and the window size is 2, then "analysis," "with," "as," and "standard" are all in the window and are considered as context for the word "polystyrene." The context words are treated as a bag of words so the order does not matter. When given any one of these four context words, the CBOW model could predict the word "polystyrene."

Skip-gram Model Skip-gram [18] is another widely used method for training word embedding models. The major difference between CBOW and Skipgram is that CBOW predicts the center word based on the surrounding words within the context window, while Skip-gram does it the opposite way. When trained on the sample input "...analysis with polystyrene as standard ...", the Skip-gram model can predict any of its context words, "analysis", "with", "as", "standard" based on the center word "polystyrene."

Empirical data have suggested that CBOW is more efficient computationally, whereas Skip-gram works better when the training corpus is relatively small [11].

As pre-trained word embedding models cannot accurately represent the meaning of words in scientific contexts. We train word embeddings using CBOW or Skip-gram on the texts from the target domain, and then use them as the fixed (i.e., untrainable) word embedding matrix for the embedding layer in the neural network (grey box in Fig. 1).

FastText Word Embedding Model The FastText model from Facebook [2] provides yet another architecture for creating word embeddings. Aside from representing a word based on its context, FastText also makes use of character n-grams. Words are mapped to character n-grams, which are then embedded in vectors. The n-gram vectors will make up a part of the embeddings for the words that do not appear frequently enough and thus do not have sufficient context in the training corpus. The addition of n-gram embeddings also greatly helps when the target word is not in the vocabulary of the pre-trained word embeddings. There is little that classic methods such as CBOW or Skip-gram can do when faced with an unknown word. They may either give it a random vector or the average of all the other vectors in the vocabulary, but, unsurprisingly, such a vector does not reflect the actual meaning of the out-of-vocabulary word. FastText, meanwhile, can capture the meaning of an unknown word better by making a word vector out of its character n-grams.

3.2 Lexicon Features

The current ontology of DBpedia has over 4.2 million entries in 774 classes. Matching all of them to the training and testing corpus is a computationally intensive task. Appending the one-hot encodings of the BIO labels for all the classes to a word vector will result in an extra $774 \times 3 = 2322$ dimensions, and for each word vector most of these dimensions will be zero. In other words, the concatenated vector will be very sparse and inefficient to compute.

To avoid diluting the dense word vector, we use only the few DBpedia classes that are relevant to the NER task at hand. For example, to identify polymers we use "Chemical Elements", "Chemical Compounds", and "Chemical Substance." When identifying social science dataset names we use "Location" and "Organization" as they are likely to be more suitable.

4 Experimental Analysis

We evaluate SciNER by applying it to two distinct scientific NER tasks: recognizing polymer names in materials science publications, and identifying dataset names in social science publications. In this section, we describe the process of obtaining and cleaning the publications, as well as how the labels were generated with minimal manual effort. We then explore the affect of using different combinations of features with SciNER to compare their influence on the extraction results. Finally, we compare SciNER against other methods for identifying scientific named entities.

4.1 Datasets

We use two distinct free-text scientific datasets to evaluate the accuracy of our models: materials science publications that contain polymer names and social science publications that contain references to datasets.

Our first dataset is a collection of 100 materials science publications from the journal *Macromolecules*. We chose this journal because we have established an agreement with its publisher, the American Chemical Society, that allows us to access the full text publications. Our second dataset comprises 6368 social science papers. We chose these papers because they are indexed by the Inter-university Consortium for Political and Social Research (ICPSR), which provides manually annotated relationships between datasets and papers in the field of social science. ICPSR has indexed over 72 000 papers. We selected a set of 6368 papers hosted by Elsevier for which we can easily download, via an API, the full text in JSON format.

4.2 Data Preparation

In order to feed our input datasets to the LSTM models we must first process the raw input publications into labeled collections of words.

Common representation The polymer and ICPSR datasets are represented in raw HTML or JSON formats which contain redundant information such as HTML tags, document object identifiers (DOIs) and publisher copyright statements. To remove these artifacts and create a clean format for processing, we parsed each file into a tree structure and removed any non-text related nodes in the tree. The resulting format includes only the raw text from the publication.

Tokenization There are two steps in the tokenization process: sentence tokenization and word tokenization. First we split each paper into sentences so that each training sample consists of a sentence, not a whole passage. We do so by applying the `tokenize_sents()` function from the Python Natural Language

Toolkit (NLTK). The reason why this is required is two-fold. The first is to assemble a large enough number of training examples, the second is to ensure that each training example has a reasonable length for the LSTM network to learn.

The second step is word tokenization, which converts each sentence from a string to a list of words and punctuations (tokens), so that it can be labeled in the next step.

Token Labeling In order to be processed by our LSTM models each token in the training set must be labeled (using BIO labeling).

For the social science dataset, we have access to an ontology of named entities from ICPSR. For the materials science dataset there is no such ontology of named entities. Here we rely on domain experts to create one. However, instead of asking them to label every word in the 100 papers, we asked them to produce a list of unique polymer names from the corpus. Each paper was reviewed by two expert reviewers to label polymer names, and when disagreement arose a third more senior domain expert made the final decision. The result is a list of 495 polymer names identified by experts in the 100 papers [29].

Then, for both datasets, we then applied an automated script to search for known named entities in the unlabeled texts, and assign a label to each token according to the BIO scheme described in Section 2.2. To reduce the number of negative examples and create a balanced training set, we removed sentences that do not have any named entities.

Lexicon Features Labeling We use the latest release of DBpedia [8] to associate class labels. As described in Section 3.2, we manually selected which classes to include for each NER task. For the social science dataset, we selected entries belonging to "Location" or "Organization." For the polymer dataset, we selected entries belonging to "Chemical Element" and "Chemical Compound" classes. We associated BIO labels automatically for each class following the same procedure as described above and encoded as additional features using one-hot encoding. The lexicon features are concatenated to the word embeddings of each word.

Splitting the Datasets The labeled examples are then split into training, validation, and test sets.

For the 6368 social science papers, we use a 64–16–20% split, yielding 14 945, 3737, and 4699 sentences in the training, validation, and test sets, respectively.

Splitting the polymer science dataset is trickier because unlike the social science papers, multiple polymers often appear in one sentence. If we divide the sentences randomly, we may end up with many polymer names occurring in both the training and the testing set, in which case our model might learn specific polymer names rather than general concepts. To mitigate this problem, we randomly select half of the unique polymers mentioned in the 100 papers and use the sentences that mention any of those polymers for training and validation,

Table 2. Experimental results when applying SciNER using different word embedding models and lexicon features to the materials science dataset. The table also includes results from the baseline ChemDataExtractor (CDE) for comparison.

#	Model and Features	With Lexicon Features			Without Lexicon Features		
		Precision	Recall	F1	Precision	Recall	F1
1	SciNER w/o pre-trained word embeddings	–	–	–	93.0%	78.2%	0.850
2	SciNER with CBOW word embeddings	84.6%	71.9%	0.777	82.1%	70.9%	0.761
3	SciNER with Skip-gram word embeddings	92.3%	81.6%	0.866	85.0%	75.4%	0.799
4	SciNER with FastText word embeddings	89.6%	92.3%	**0.909**	82.3%	80.6%	0.814
5	CDE (NLP module only)	–	–	–	54.3%	58.3%	0.562
6	CDE (NLP+regex+dictionary)	–	–	–	65.1%	58.7%	0.617

while the rest makes up the test set. We split the first group 80–20, yielding 3676 sentences for training and 919 sentences for validation, and leaving 2497 sentences for testing. As a single sentence can contain more than one polymer name, polymer names can still co-occur in the training and test sets. In practice, we find that only 18.8% of polymers co-occur in this way, which we view as acceptable.

4.3 Experimental Results

We now explore the accuracy of the SciNER LSTM models using different word embedding models and lexicon features. We apply SciNER to both the materials science and social science datasets to demonstrate its effectiveness and generalizability. For the polymer name recognition task, we compare our results with a state-of-the-art domain-specific toolkit, ChemDataExtractor (CDE) [28]. For the social science dataset, in which we aim to identify dataset names, we could not identify a readily available toolkit that performs a similar tasks, so we compare our results with a basic KNN classifier.

Experiments on the materials science dataset Table 2 compares the precision, recall, and F1 scores of our LSTM model, when fed with different word embedding and lexicon features, to those achieved by CDE. As shown in the table, Tests 1–4 evaluate the effect of different word embedding models on the performance of SciNER with and without lexicon features. In Test 1, the word embedding matrix in the Embedding layer is randomly initialized as described in Section 3.1. In Tests 2–4, word embeddings are trained on the same materials science corpus before being fed to the model as the fixed weights in the Embedding layer. In Test 2, word embeddings are trained using the CBOW model (Section 3.1). Note that it produces the lowest F1 score among the first four tests, which is not surprising considering that the words used in academic papers usually follow a long tail distribution, and CBOW is not good at handling infrequent words. The model in Test 3 is fed with word vectors trained using the Skip-gram model, which is designed to better encode rare words, resulting in a

Table 3. Experimental results for social science corpus

#	Model	Precision	Recall	F1 Score
7	SciNER w/ FastText embeddings & lexicon features	82.5%	87.0%	0.847
8	KNN classifier enhanced by rules [29]	60.0%	58.7%	0.592

10% improvement in F1 score compared to CBOW. The fourth test uses word embedding generated by FastText, which encodes character n-grams in addition to contextual information. It produces the best results, achieving an F1 score of 0.909.

To explore the benefits of the lexicon features we also ran Tests 2–4 without the lexicon features. The table shows that the lexicon features improves the F1 score by 11% in the best case (row 4).

Tests 5–6 show the results achieved by CDE, which is the state-of-the-art model for recognizing chemical entities [28]. When only using CDE's NLP module, we get an F1 score of 0.562. When using the entire CDE pipeline, which relies on regular expression rules and dictionary, the F1 score increases to 0.617. In either case, SciNER's F1 score exceeds CDE by approximately 50%.

Experiments on the social science corpus For the social science dataset we apply only FastText word embeddings and lexicon features, as the previous experiment demonstrated that this configuration performed best of the configurations studied. Table 3 compares the precision, recall, and F1 scores of SciNER to our previous work, in which we used a KNN classifier and many manually created rules [29]. Even in this quite different environment, SciNER achieves an F1 score of 0.847, significantly outperforming our rule-based approach that achieves an F1 score of 0.592. This result highlights the value of SciNER, as the dataset names included in social science publications are significantly different from the polymer names included in materials science publications. Each domain uses a different set of frequently used words and domain-specific jargon. Another less obvious, but more challenging, difference is that dataset names are usually much longer than polymer names. Dataset names with more than ten words are not uncommon.

5 Related Work

Researchers have explored myriad approaches to scientific NER. Most approaches rely on crowdsourcing [27, 32] or rule-based systems [24]. For example, AQL is a declarative rule language used in IBM's SystemT [15]. With AQL, users can define a set of rules, which SystemT then uses to optimize and build an efficient query plan. SystemT can support complex expressions, but like all rule-based systems, still requires manual effort to define rules, and thus its accuracy is highly dependent on the proper construction of rules.

In other cases, extraction systems are dependent on rich domain-specific ontologies via which named entities can be matched directly with terms in the ontology [13, 20, 23]. High NER accuracy has been achieved in biomedicine [5, 10], due to the availability of structured databases (e.g., Uniprot and PDB) and well-defined, unique identifiers and names (e.g., gene/protein names, diseases, organisms) that can be easily identified in free text (e.g., the string "PDB:1BFM" denotes the 1BFM protein in the PDB database, in this case a histone protein). Few other scientific communities have achieved such a high level of standardization, which is one of the reasons that we have chosen to focus on NER in domains where standard identifiers for named entities are not readily available.

Word embeddings have been shown to be effective at capturing latent information in scientific publications, including in materials science [14, 30]. Prior work has shown that the embeddings can capture complex materials science concepts such as structure-property relationships and the relative positions of elements in the periodic table [30]. Those results motivated our use of unsupervised word embeddings rather than hand-curated features to represent words as input to our model.

6 Conclusion

The exponential growth in the number of academic papers has made it infeasible for researchers to manually discover important scientific facts buried deep within these free text publications.

SciNER aims to address part of this problem by automatically identifying scientific named entities in free text publications. SciNER specifiably focuses on addressing challenges associated with the rare words and terminologies used in scientific texts. By leveraging external sources of knowledge and training on scientific texts, SciNER produces more meaningful vectors than traditional word embeddings.

Our experiments demonstrate that SciNER is able to accurately identify diverse named entities from materials science and social science publications. Our best result for identifying polymer names reached an F1 score of 0.909—far exceeding the 0.617 achieved by ChemDataExtractor, the state-of-the-art domain-specific toolkit. When applied to the task of extracting social science dataset names SciNER achieved an F1 score of 0.847, significantly better than the 0.592 achieved by a KNN-based classifier.

In future work we aim to expand SciNER to more domains (e.g. biomedical research) and test its performance against widely used domain ontologies (e.g. the FDA database). We will explore the use of deep neural network-based word embeddings (e.g., BERT [9] and ELMo [21]) to improve extraction performance and design a pipeline for identifying relations between entities, of which SciNER is the first component. Our hope is that the structured data extracted from publications will benefit many applications, such as discovering new molecule pathways and enabling targeted material design.

References

1. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: DBpedia: A nucleus for a web of open data. In: The semantic web, pp. 722–735. Springer (2007)
2. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. arXiv preprint arXiv:1607.04606 (2016)
3. Bonney, R., Cooper, C.B., Dickinson, J., Kelling, S., Phillips, T., Rosenberg, K.V., Shirk, J.: Citizen science: a developing tool for expanding science knowledge and scientific literacy. BioScience **59**(11), 977–984 (2009)
4. Bonney, R., Shirk, J.L., Phillips, T.B., Wiggins, A., Ballard, H.L., Miller-Rushing, A.J., Parrish, J.K.: Next steps for citizen science. Science **343**(6178), 1436–1437 (2014)
5. Brase, J.: DataCite–A global registration agency for research data. In: 4th International Conference on Cooperation and Promotion of Information Resources in Science and Technology. pp. 257–261. IEEE (2009)
6. Chiu, J.P., Nichols, E.: Named entity recognition with bidirectional LSTM-CNNs. Transactions of the Association for Computational Linguistics **4**, 357–370 (2016)
7. Cohn, J.P.: Citizen science: Can volunteers do real research? BioScience **58**(3), 192–197 (2008)
8. DBpedia: DBpdia ontology. https://wiki.dbpedia.org/services-resources/ontology (2019), [Online; accessed 11-April-2018]
9. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
10. Duggan, M.: System and method for generating unique and persistent identifiers (Jan 10 2008), US Patent App. 11/444,887
11. Enríquez, F., Troyano, J.A., López-Solaz, T.: An approach to the use of word embeddings in an opinion classification task. Expert Systems with Applications **66**, 1–6 (2016)
12. Fortunato, S., Bergstrom, C.T., Börner, K., Evans, J.A., Helbing, D., Milojević, S., Petersen, A.M., Radicchi, F., Sinatra, R., Uzzi, B., Vespignani, A., Waltman, L., Wang, D., Barabási, A.L.: Science of science. Science **359**(6379) (2018). https://doi.org/10.1126/science.aao0185, https://science.sciencemag.org/content/359/6379/eaao0185
13. Friedman, C., Kra, P., Yu, H., Krauthammer, M., Rzhetsky, A.: GENIES: A natural-language processing system for the extraction of molecular pathways from journal articles. In: ISMB (supplement of bioinformatics). pp. 74–82 (2001)
14. Isayev, O.: Text mining facilitates materials discovery. Nature **571**(7763), 42 (2019)
15. Krishnamurthy, R., Li, Y., Raghavan, S., Reiss, F., Vaithyanathan, S., Zhu, H.: SystemT: A system for declarative information extraction. ACM SIGMOD Record **37**(4), 7–13 (2009)
16. Mathiak, B., Boland, K.: Challenges in matching dataset citation strings to datasets in social science. D-Lib Magazine **21**(1/2), 23–28 (2015)
17. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
18. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems. pp. 3111–3119 (2013)

19. Ohama, Y.: Handbook of Polymer-modified Concrete and Mortars: Properties and Process tTchnology. William Andrew (1995)
20. Ono, T., Hishigaki, H., Tanigami, A., Takagi, T.: Automated extraction of information on protein–protein interactions from the biological literature. Bioinformatics **17**(2), 155–161 (2001)
21. Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. arXiv preprint arXiv:1802.05365 (2018)
22. Ramshaw, L.A., Marcus, M.P.: Text chunking using transformation-based learning. In: Natural Language Processing Using Very Large Corpora, pp. 157–176. Springer (1999)
23. Rzhetsky, A., Iossifov, I., Koike, T., Krauthammer, M., Kra, P., Morris, M., Yu, H., Duboué, P.A., Weng, W., Wilbur, W.J., et al.: GeneWays: A system for extracting, analyzing, visualizing, and integrating molecular pathway data. Journal of biomedical informatics **37**(1), 43–53 (2004)
24. Shaalan, K., Raza, H.: Arabic named entity recognition from diverse text types. In: International Conference on Natural Language Processing. pp. 440–451. Springer (2008)
25. Sommerfeld, A.K.: Education as a collective accomplishment: how personal, peer, and parent expectations interact to promote degree attainment. Social Psychology of Education **19**(2), 345–365 (2016)
26. Stewart, C.A., Cockerill, T.M., Foster, I., Hancock, D., Merchant, N., Skidmore, E., Stanzione, D., Taylor, J., Tuecke, S., Turner, G., et al.: Jetstream: A self-provisoned, scalable science and engineering cloud environment. In: XSEDE Conference (2015)
27. Sui, D., Elwood, S., Goodchild, M.: Crowdsourcing Geographic Knowledge: Colunteered Geographic Information (VGI) in Theory and Practice. Springer Science & Business Media (2012)
28. Swain, M.C., Cole, J.M.: ChemDataExtractor: A toolkit for automated extraction of chemical information from the scientific literature. Journal of Chemical Information and Modeling **56**(10), 1894–1904 (2016)
29. Tchoua, R.B., Ajith, A., Hong, Z., Ward, L.T., Chard, K., Belikov, A., Audus, D.J., Patel, S., de Pablo, J.J., Foster, I.T.: Creating training data for scientific named entity recognition with minimal human effort. In: International Conference on Computational Science. pp. 398–411. Springer (2019)
30. Tshitoyan, V., Dagdelen, J., Weston, L., Dunn, A., Rong, Z., Kononova, O., Persson, K.A., Ceder, G., Jain, A.: Unsupervised word embeddings capture latent knowledge from materials science literature. Nature **571**(7763), 95 (2019)
31. Ware, M., Mabe, M.: The STM report: An overview of scientific and scholarly journal publishing. Tech. rep., International Association of Scientific, Technical and Medical Publishers (2015)
32. Wiggins, A., Crowston, K.: From conservation to crowdsourcing: A typology of citizen science. In: 44th Hawaii international conference on system sciences. pp. 1–10. IEEE (2011)
33. Yang, Z., Fujii, Y., Lee, F.K., Lam, C.H., Tsui, O.K.: Glass transition dynamics and surface layer mobility in unentangled polystyrene films. Science **328**(5986), 1676–1679 (2010)