Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022), pages 4540–4545
Marseille, 20-25 June 2022

Using Sentence-level Classification Helps Entity Extraction from Material Science Literature

Ankan Mullick[†] **Shubhraneel Pal**[†] **Tapas Nayak**[††]
Seung-Cheol Lee[♠] **Satadeep Bhattacharjee**[♠] **Pawan Goyal**[†]
[†]Department of Computer Science and Engineering, IIT Kharagpur, India.
[††]TCS Research, India.
[♠]Indo-Korea Science and Technology
ankanm@kgpian.iitkgp.ac.in, shubhraneel@iitkgp.ac.in, tnk02.05@gmail.com,
seungcheol.lee@ikst.res.in, s.bhattacharjee@ikst.res.in, pawang@cse.iitkgp.ac.in

Abstract

In the last few years, several attempts have been made on extracting information from material science research domain. Material Science research articles are a rich source of information about various entities related to material science such as names of the materials used for experiments, the computational software used along with its parameters, the method used in the experiments, etc. But the distribution of these entities is not uniform across different sections of research articles. Most of the sentences in the research articles do not contain any entity. In this work, we first use a sentence-level classifier to identify sentences containing at least one entity mention. Next, we apply the information extraction models only on the filtered sentences, to extract various entities of interest. Our experiments for named entity recognition in the material science research articles show that this additional sentence-level classification step helps to improve the F1 score by more than 4%.

Keywords: Material Science Information Extraction, Material Science Entity Detection, Material Science Research

1. Introduction

Every year, a large number of research articles are published in the material science domain. It is very difficult to find informative entities in these articles as they spread across the articles. Recently, researchers try to solve this problem by using information extraction in the material science domain. Various models are proposed to extract informative entities such as the materials used in the research work, simulation software used and its parameters, the method used in the study, the used code languages, and finally, the outcome of the study, etc. These entities are very common in material science articles. It is critical to extract and store them in a structured way. But the task of information extraction from the entire articles becomes complex as the articles are quite long and they contain too many redundant sentences. Most of the text in the articles does not contain any such entities thus leading to erroneous extraction. In this paper, we address this issue of identifying the text in the articles that are devoid of such entities and the text that contains the informative entities. Successfully eliminating the text from the articles that do not contain any entities can help to apply the information extraction models to the entire articles.

In this work, we target five types of informative entities from the material science domain such as material names, method names, code or simulation software names, parameters of the simulation software, and structure type of the materials. We consider a sentence as *informative* if contains any of these five types of entities. First, we build deep neural network-based binary sentence classification models to identify the informative sentences from the articles. Our experiments show that separating the uninformative sentences before applying the entity extraction model significantly improves the model performance (more than 4% F1 score). Finally, we analyze the distribution of these five entities across different sections of the research articles and how this distribution varies across time.

2. Related Work

With the progress of deep neural architecture in natural language processing, information extraction from text is applied to different heterogeneous scientific domains such as chemistry, biomedical, and most recently material science. Chemical science entities are among the first to be extracted from chemistry literature using information extraction approaches. OSCAR4 recognizer (Jessop et al., 2011) is an n-gram based Bayesian binary classifier that classifies tokens to 'chemical' or 'non-chemical' classes. They build this n-gram model using a dictionary of chemical tokens and it often fails when tokens are out of this dictionary. ChemSpot (Rocktäschel et al., 2012), tmChem (Leaman et al., 2015), and ChemDataExtractor (Swain and Cole, 2016) are machine learning-based tools that can extract chemical entities from the chemistry literature. (Huang and Cole, 2020) use ChemDataExtractor tool to create a battery database from the material science articles related to battery materials. They extract head and tail entities for five types of relations from the articles to build this database. (Dragone et al., 2017) propose a system that can evaluate chemical reactivity and detect new reactions, rather than a predefined set of targets. (Hakimi et al., 2020) use machine learning-based NLP models for biomaterial text mining.

Recently, researchers extend the idea of word embeddings and deep neural architectures to the material science domain also. (Tshitoyan et al., 2019) use the idea of Word2Vec (Mikolov et al., 2013) to obtain the word embeddings of material science tokens and show that the obtained embeddings can capture latent knowledge from the text - mat2vec (material science embedding). (Kim et al., 2017a; Kim et al., 2017b) apply information extraction and machine learning algorithms to extract the parameters of synthesis procedures from material science articles. Similarly, (Court and Cole, 2020) explore machine learning to extract transition temperatures and phase diagrams of magnetic materials and superconducting materials from text. (Correa-Baena et al., 2018) study machine learning and natural language processing to accelerate the research of novel materials development. (Goldsmith et al., 2018) show how machine learning can be useful for aiding heterogeneous catalyst understanding, design and discovery. (Mysore et al., 2017) extract graph structures from material science literature using neural network approaches. (Weston et al., 2019) use word embeddings for named entity recognition in material science articles. They apply a long short-term memory network (Hochreiter and Schmidhuber, 1997) and conditional random field for this task. They consider this task as a sequence labeling task and use a model inspired from (Lample et al., 2016) for the same. (Guha et al., 2021) develop tool to generate database for material science literature[1]. All the previous works consider the entire research articles as informative and run their models on it, thus making the end task complex. But as we describe before, the majority of portions of an article do not contain any informative information. In this paper we address this issue. We distinguish the informative and uninformative parts of the articles and then run the information extraction module only on the informative part of it which leads to better performance on the task.

Table 1: Statistics of the entities in our annotated dataset.

Entity type	Example	Count w.r.t. total entities	Percentage
CODE	BOLTZTRAP	304	1.75%
MATERIAL	EuCd2As2	9,161	52.74%
METHOD	DFT (Density Functional Theory	4,602	26.49%
PARAMETER	4*4*4 K-Point	1,387	7.98%
STRUCTURE	Hexagonal	1,918	11.04%
Total		17,372	100%

3. Dataset

We collect material science articles from (Guha et al., 2021) where total 10,500 articles[2] of material science domain. We use spacy[3] for tokenization and extract tokens with their labels. There is a total of 47,262 sentences in the extracted dataset. Out of 10,500 articles, 214 randomly selected articles are annotated using material science domain experts using Pdfanno[4]. Two annotators annotate independently and Inter-annotator agreement (Cohen κ) is 0.81. Any conflict is resolved by the third annotator. Total annotation time is three weeks. Five informative entity types are labeled by annotators- a) material b) method c) code d) parameter e) structure.

We use the inside-outside-beginning (IOB) tagging format for the five entity classes. Any token which is not associated with these classes is marked as 'Other' class, "O". We label a sentence as "informative" if it contains an entity from any of the five class; otherwise the sentence is labeled as "uninformative". The informative sentence is also labeled with the entity class labels (code, materials, etc.). This dataset contains a total 15,699 ($\sim 31.64\%$) informative sentences among a total of 49,610 sentences. Among the informative sentences, the total count (not unique) and examples of material, method, code, parameter, structure entities along with their percentages are shown in Table 1. In Table 2, we include the distribution of the sentences in different sections of the articles (abstract, introduction, experiment, conclusion and others) to the five entity type categories. It should be noted here that one sentence can belong to multiple type categories if they contain more than one type of entity. That is why the total row sum may be greater than 100%.

4. Evaluation Metric

We report precision, recall, F1 score, and accuracy for our sentence identification models. Accuracy is measured for informative and uninformative sentences together. Since there is an imbalance in informative and uninformative sentences in the dataset and only informative sentences are used for named entity recognition, we report the precision, recall, and F1 score for the informative sentence class separately. For the entity extraction models, we report precision, recall, and F1 score. An extracted entity is assumed correct if the entire entity is matched with a ground truth entity and their corresponding type also matches.

5. Experiments

The task of identifying if a sentence contains any material science entity or not can be designed as a binary classification task. We label a sentence as 'informative' if it contains any informative entity, otherwise, the sentence is labeled as 'uninformative'. We explore traditional machine learning-based approaches like - Naïve Bayes (NB), Support Vector Machine (SVM), Logistic Regression (LR), Random Forest (RF), Bagging

[1]We have used the last three approaches as baselines.

[2]articles are crawled from 'cond-mat.mtrl-sci' category with at least one code listed on `https://psi-k.net/software`.

[3]https://spacy.io/

[4]https://github.com/paperai/pdfanno

Table 2: Distribution of sentences in different sections of the articles to five types of entities in the annotated dataset.

Section	Inf	Code	Mat	Meth	Param	Struct
abstract	40.45	1.77	62.32	38.63	5.85	4.78
introduction	31.93	0.97	61.76	45.30	0.80	9.03
experiment	39.95	2.81	58.04	41.62	3.82	8.90
conclusion	28.46	0.57	61.60	39.38	6.34	7.97
other	31.51	3.65	48.30	40.87	10.82	10.95

Table 3: Precision (P), Recall (R), F1 score for informative sentences and overall Accuracy (A) [in %] with respective standard deviations (SD) of the models on the binary informative sentence identification task from entire articles.

Model	P, SD(P)	R, SD(R)	F1, SD(F1)	A, SD(A)
NB	74.08, 0.92	78.1, 0.92	76.03, 0.91	85.11, 0.42
SVM	61.17, 0.8	55.91, 0.56	58.42, 0.56	75.93, 0.39
LR	91.19, 0.78	77.61, 1.39	83.85, 0.89	91.16, 0.45
RF	92.75, 0.35	77.71, 0.92	84.56, 0.66	91.42, 0.33
Bg	92.06, 0.88	69.04, 1.9	78.91, 1.02	88.83, 0.39
BiLSTM (M2V)	85.61, 0.33	86.4, 0.35	86.01, 0.18	87.8, 0.06
CNN (Sci)	91.27, 1.56	92.76, 0.64	92.01, 0.84	92.10, 1.01
BERT	**98.69**, 0.24	**97.67**, 0.22	**98.18**, 0.13	**98.84**, 0.08
SciBERT	90.16, 2.72	91.45, 0.29	90.81, 1.43	92.03, 1.16
DistilBERT	98.54, 0.17	97.29, 0.33	97.92, 0.11	98.75, 0.06

(Bg), and deep neural network-based models - Long Short Term Memory (LSTM), Convolutional Neural Network (CNN) and different embeddings to solve this classification task.

We can broadly classify identified features into four categories - (i) *Parts of speech (POS) tag-based features*: We use Stanford POS tagger (Manning et al., 2014) to find the number of nouns, verbs, adjectives, presence of adverbs, etc. (ii) *Tf-Idf based features*: n-gram (one, two etc.) based features. (iii) *Dependency parse based features* (using Stanford Dependency parser (De Marneffe and Manning, 2008)): dobj (direct object), amod (adjective modifier), acomp (adjectival complement) etc. (iv) *Others*: no. of characters, presence of wh-words, numbers, strong, weak adjectives, words specific to particular categories. Based on all these features, we classify the sentences and report precision, recall, F1 score for informative sentences and overall accuracy along with respective standard deviations corresponding to five models - Naïve Bayes (NB), Support Vector Machine (SVM), Logistic Regression (LR), Random Forest (RF), Bagging (Bg) as shown in Table 3. Among all the features, bi-gram, amod, acomp, number of nouns, adjectives have the highest information gain.

We explore several deep neural network models (BiLSTM, CNN) on top of different embeddings - material science domain specific mat2vec (m2v) vectors (Tshitoyan et al., 2019) and scientific text embedding SciBERT (Beltagy et al., 2019). Each word is converted to a vector using the embedding model and these vectors are sequentially passed into an BiLSTM or CNN network to compute the vector for the entire sentence. Final softmax layer predicts the class probability. We run the experiments with different batch sizes (16, 32, 64), epochs (10-50) and learning rates (5e-4, 1e-4, 5e-5, 1e-5, 5e-6, 1e-6), and other hyper-parameters[5]. Table 3 shows best results for BiLSTM and CNN (for best embedding). We also use the fine tuned version of BERT (Devlin et al., 2018) (uncased with linear model and base), SciBERT (Beltagy et al., 2019) and DistilBERT (Sanh et al., 2019) for classification (using CLS embedding). We perform 5-fold cross-validation test on the dataset for various approaches and include the precision (P), recall (R), F1 score for identifying sentences with at least one entity and overall classification accuracy (A) along with their respective standard deviations (SD) in Table 3. We randomly select 5% data as validation set for parameter tuning also. For the proposed task, an ideal classifier would be one with very good recall with decent precision. A lower precision would simply amount to indexing of some extra sentences, which may never be used. We observe that the best recall, with good precision, F1 and overall accuracy are obtained with BERT embeddings and the best result is provided by fine tuned BERT model. We also see that deep neural network-based models outperform the traditional machine learning models.

We hypothesize that identifying the informative and uninformative sentences can improve performance on the information extraction tasks. We choose named entity recognition (NER) of the mentioned entity classes as the end task. We use different models for NER task: SciBERT, BERT, DistilBERT and Bi-LSTM-CRF Elmo model. In addition to tho above methods, we use several baseline approaches - (i) DCNN (Diluted CNN) and Bi-LSTM-CRF model by (Mysore et al., 2017) (ii) BiLSTM named entity recognizer for specific material science articles by (Weston et al., 2019). (iii) Bi-LSTM CRF with noise (Mimicking Model) by (Guha et al., 2021). The above NER models are trained on informative sentences identified by the BERT model. We also explore different joint approaches for entity extractions. Multi-Granularity model (MGM) (Da San Martino et al., 2019) and SC-NER (Wang et al., 2019) joint models are experimented to extract entities.

We randomly select 20% data as test set, 5% data as

[5]Due to space shortage only the best results are shown for the model. We also explore LSTM, RNN and other combinations but results are poor comparatively.

Table 4: Precision, Recall, F1 score [in %] of different NERs for entity extraction from all sentences and only informative sentences in the articles.

	Precision		Recall		F1	
Method	All	Inf	All	Inf	All	Inf
Sci-BERT (Beltagy et al., 2019)	77.32	81.93	70.24	71.21	73.61	76.19
BERT (Devlin et al., 2018)	74.08	76.45	70.72	72.86	72.36	74.61
DistilBERT (Sanh et al., 2019)	71.87	72.69	69.48	70.59	70.66	71.62
DCNN (Mysore et al., 2017)	79.43	83.02	78.89	79.53	79.15	81.24
BiLSTM-CRF (Mysore et al., 2017)	81.56	84.29	79.37	80.52	80.45	82.36
BiLSTM (Weston et al., 2019)	79.61	82.53	73.82	75.98	76.60	79.12
MGM (Da San Martino et al., 2019)	74.33	-	70.91	-	72.58	-
SC-NER (Wang et al., 2019)	75.64	-	79.12	-	77.34	-
Mimicking (Guha et al., 2021)	82.08	85.93	83.72	86.08	82.89	86.01
BiLSTM-CRF Elmo	87.35	**91.71**	82.19	**86.09**	84.57	**88.76**

Table 5: Entity-wise Precision, Recall, F1 score [in %] of BiLSTM-CRF ELMO NER for entity extraction from all sentences and only informative sentences.

	Precision		Recall		F1	
Entity type	All	Inf	All	Inf	All	Inf
MATERIAL	85.49	91.66	88.44	90.08	86.94	90.86
METHOD	95.53	96.03	82.80	86.72	88.71	91.13
STRUCTURE	95.92	96.71	89.81	93.87	92.76	95.27
PARAMETER	73.27	81.76	62.51	70.31	67.46	75.60
CODE	86.54	92.39	87.38	89.47	86.96	90.91
Overall	87.35	**91.71**	82.19	**86.09**	84.57	**88.76**

validation set and rest of the data are considered for model training. We include the overall macro average percentage of Precision, Recall and F1 score of all models for testing on all sentences ("All") and only informative sentences ("Inf") in Table 4. Precision , Recall and F1-score are reported on test data. Thus, in the test dataset, if a span is present in the training dataset with a class annotation, it is given that particular label. We use pre-trained Elmo (Peters et al., 2018) embedding [6] for material science articles. The input to the Bi-LSTM-CRF model is thus a concatenation of pre-trained Word2Vec embedding (Mikolov et al., 2013), character embedding, and pre-trained ELMO embedding (Peters et al., 2018) along with the IOB tags as the target of each word. We fine-tune the model with Adam optimizer, dropout of 0.5, hidden dimension of 200, number of epochs at 120, and a batch size of 8 to get the overall optimum (for all entities together) F1 score. The trained model is tested on 'all' sentences as well as only the identified 'informative' sentences. Joint models (MGM (Da San Martino et al., 2019) and SC-NER (Wang et al., 2019)) are applied on all sentences directly. From table 4, we see that precision, recall and F1 score improve for all the NER models when we apply it on informative sentences compared to applying on all sentences. BiLSTM-CRF Elmo model performs the best in terms of precision, recall and F1-score and also achieves 4% higher F1 score in this setup (BERT based informative sentence identification and Bi-LSTM-CRF Elmo NER model). It also outperforms joint models. Mimicking model (Guha et al., 2021) also performs well but may be due to the added noise, former method outperforms it. In Table 5, we include the performance of BiLSTM-CRF Elmo model on different entity type classes. We see that precision, recall and F1 score improve across all different entity classes. We train models on 2 NVidia Tesla P100 GPUs (12GB and 16GB RAM) with 3584 CUDA cores.

5.1. Analysis

For analysis of our models on a larger set of material science articles (unannotated dataset), we randomly select another set of 7,798 articles from the rest crawled dataset. We use the BERT (uncased) based classifier to analyze this unannotated dataset. In the first step, we identify the informative sentences in this dataset. This dataset (7798 articles) contains a total of $\sim$ 1.9 million sentences, out of which $\sim$ 0.675 million (35.5%) sentences are found to be informative. We analyze the distribution of informative sentences and different entity classes across various sections (abstract, introduction, experiment, conclusion, and others) of the research articles by the Bi-LSTM-CRF Elmo model. Table 6 shows that our two-stage model predicts 1/3 of the sentences as informative ones across all major sections and material (Mat) and method (Meth) entity types are having a very large percentage. Parameter (Param) and Structure (Struct) entity types have the lesser portions in the articles. Code entity is having the minimum share in the dataset. We see that the distribution of sentences in different sections of articles to the five entity type categories in the unannotated dataset is following the actual distribution in the annotated dataset.

Table 6: Distribution of the sentences (in %) in different sections of the articles from the unannotated dataset to five types of entities predicted by 2-stage model.

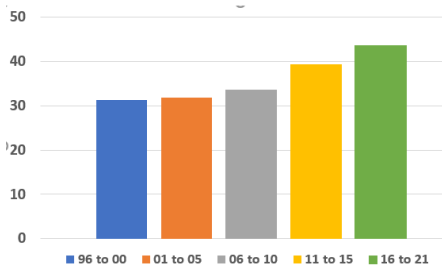
Section	Inf	Code	Mat	Meth	Param	Struct
abstract	36.12	0.15	66.87	35.29	4.48	2.97
introduction	37.41	0.17	59.42	40.71	9.56	3.42
experiment	35.01	1.87	65.41	23.51	12.41	4.26
conclusion	34.75	0.56	63.97	38.65	7.60	3.11
other	30.99	1.18	65.75	28.09	12.61	3.22

There are multiple entity types present in one sen-

[6] https://figshare.com/s/ec677e7db3cf2b7db4bf

tence - method and material types have the most overlap (10.95%), parameter and method types have the second-highest overlap (4.06%), whereas code and structure types have the least overlap (0.001%).
We show the yearly distribution of average informative sentences in the material science articles from 1996 to 2021 in Figure 1 (in the bucket of five years). It shows the gradual increment of informative sentences from 1996 to 2021 across the entire article which signifies that nowadays, people are interested in writing articles with more domain-oriented informative entities in material science literature.

Figure 1: Yearly distribution [in %] of informative sentences as predicted by the BERT model on the unannotated dataset.



6. Conclusion

In this work, we address the issue of identifying informative sentences and extract entities in the material science research articles. We propose deep neural network-based models to classify sentences into these two classes concerning five types of entities such as material, method, code, parameter, and structure. Our experiments show that the two-stage framework (identify informative sentences and then extract entity) leads to significant improvement in the performance of the end task of extracting these five types of entities from the articles than direct extraction of entities from all sentences. Our fine tuned BiLSTM-CRF Elmo model performs the best. We also analyze in detail the distribution of these entities in the articles. In the future, we would like to extend this work to include more types of entities and to other domains of scientific articles.

References

Beltagy, I., Cohan, A., and Lo, K. (2019). Scibert: Pretrained contextualized embeddings for scientific text. *arXiv preprint arXiv:1903.10676*.

Correa-Baena, J.-P., Hippalgaonkar, K., van Duren, J., Jaffer, S., Chandrasekhar, V. R., Stevanovic, V., Wadia, C., Guha, S., and Buonassisi, T. (2018). Accelerating materials development via automation, machine learning, and high-performance computing. *Joule*.

Court, C. J. and Cole, J. (2020). Magnetic and superconducting phase diagrams and transition temperatures predicted using text mining and machine learning. *npj Computational Materials*.

Da San Martino, G., Yu, S., Barrón-Cedeno, A., Petrov, R., and Nakov, P. (2019). Fine-grained analysis of propaganda in news article. 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 5636–5646.

De Marneffe, M.-C. and Manning, C. D. (2008). Stanford typed dependencies manual. Technical report, Technical report, Stanford University.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dragone, V., Sans, V., Henson, A. B., Granda, J. M., and Cronin, L. (2017). An autonomous organic reaction search engine for chemical reactivity. *Nature communications*.

Goldsmith, B. R., Esterhuizen, J., Liu, J.-X., Bartel, C. J., and Sutton, C. (2018). Machine learning for heterogeneous catalyst design and discovery. *Aiche Journal*.

Guha, S., Mullick, A., Agrawal, J., Ram, S., Ghui, S., Lee, S.-C., Bhattacharjee, S., and Goyal, P. (2021). Matscie: An automated tool for the generation of databases of methods and parameters used in the computational materials science literature. *Computational Materials Science*, 192:110325.

Hakimi, O., Krallinger, M., and Ginebra, M.-P. (2020). Time to kick-start text mining for biomaterials. *Nature Reviews Materials*, 5(8):553–556.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Huang, S. and Cole, J. (2020). A database of battery materials auto-generated using chemdataextractor. *Scientific Data*.

Jessop, D. M., Adams, S. E., Willighagen, E. L., Hawizy, L., and Murray-Rust, P. (2011). Oscar4: a flexible architecture for chemical text-mining. *Journal of cheminformatics*, 3(1):41.

Kim, E., Huang, K., Saunders, A., McCallum, A., Ceder, G., and Olivetti, E. (2017a). Materials synthesis insights from scientific literature via text extraction and machine learning. *Chemistry of Materials*.

Kim, E., Huang, K., Tomala, A., Matthews, S., Strubell, E., Saunders, A., McCallum, A., and Olivetti, E. (2017b). Machine-learned and codified synthesis parameters of oxide materials. *Scientific Data*.

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., and Dyer, C. (2016). Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

Leaman, R., Wei, C.-H., and Lu, Z. (2015). tmchem: a high performance approach for chemical

named entity recognition and normalization. *Journal of Cheminformatics*.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J. R., Bethard, S., and McClosky, D. (2014). The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mysore, S., Kim, E., Strubell, E., Liu, A., Chang, H.-S., Kompella, S., Huang, K., McCallum, A., and Olivetti, E. (2017). Automatically extracting action graphs from materials science synthesis procedures. *arXiv preprint arXiv:1711.06872*.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

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

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. (2019). Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Swain, M. and Cole, J. (2016). Chemdataextractor: A toolkit for automated extraction of chemical information from the scientific literature. *Journal of chemical information and modeling*.

Tshitoyan, V., Dagdelen, J., Weston, L., Dunn, A., Rong, Z., Kononova, O., Persson, K. A., Ceder, G., and Jain, A. (2019). Unsupervised word embeddings capture latent knowledge from materials science literature. *Nature*, 571(7763):95–98.

Wang, Y., Li, Y., Zhu, Z., Xia, B., and Liu, Z. (2019). Sc-ner: A sequence-to-sequence model with sentence classification for named entity recognition. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 198–209. Springer.

Weston, L., Tshitoyan, V., Dagdelen, J., Kononova, O., Trewartha, A., Persson, K. A., Ceder, G., and Jain, A. (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.