Title

Named Entity Recognition and Normalization Applied to Large-Scale Information Extraction from the Materials Science Literature

Journal

Journal of Chemical Information and Modeling, 59(9)

ISSN

1549-9596

Authors

Weston, L
Tshitoyan, V
Dagdelen, J
et al.

Publication Date

2019-09-23

DOI

10.1021/acs.jcim.9b00470

Peer reviewed

doi.org/10.26434/chemrxiv.8226068.v1

Named Entity Recognition and Normalization Applied to Large-Scale Information Extraction from the Materials Science Literature

Leigh Weston, Vahe Tshitoyan, John Dagdelen, Olga Kononova, Kristin Persson, Gerbrand Ceder, Anubhav Jain

Submitted date: 04/06/2019 • Posted date: 05/06/2019

Citation information: Weston, Leigh; Tshitoyan, Vahe; Dagdelen, John; Kononova, Olga; Persson, Kristin; Ceder, Gerbrand; et al. (2019): Named Entity Recognition and Normalization Applied to Large-Scale Information Extraction from the Materials Science Literature. ChemRxiv. Preprint.

Over the past decades, the number of published materials science articles has increased manyfold. Now, a major bottleneck in the materials discovery pipeline arises in connecting new results with the previously established literature. A potential solution to this problem is to map the unstructured raw-text of published articles onto a structured database entry that allows for programmatic querying. To this end, we apply text-mining with named entity recognition (NER), along with entity normalization, for large-scale information extraction from the published materials science literature. The NER is based on supervised machine learning with a recurrent neural network architecture, and the model is trained to extract summary-level information from materials science documents, including: inorganic material mentions, sample descriptors, phase labels, material properties and applications, as well as any synthesis and characterization methods used. Our classifer, with an overall accuracy (f1) of 87% on a test set, is applied to information extraction from 3.27 million materials science abstracts - the most information-dense section of published articles.Overall, we extract more than 80 million materials-science-related named entities, and the content of each abstract is represented as a database entry in a structured format. Our database shows far greater recall in document retrieval when compared to traditional text-based searches due to an entity normalization procedure that recognizes synonyms. We demonstrate that simple database queries can be used to answer complex \meta-questions" of the published literature that would have previously required laborious, manual literature searches to answer. All of our data has been made freely available for bulk download; we have also made a public facing application programming interface (https://github.com/materialsintelligence/matscholar) and website http://matscholar.herokuapp.com/search for easy interfacing with the data, trained models and functionality described in this paper. These results will allow researchers to access targeted information on a scale and with a speed that has not been previously available, and can be expected to accelerate the pace of future materials science discovery.

File list (1)

NER_chemrxiv.pdf (1.87 MiB) view on ChemRxiv • download file

Named Entity Recognition and Normalization Applied to Large-Scale Information Extraction from the Materials Science Literature

L. Weston,[1] V. Tshitoyan,[2] J. Dagdelen,[1], O. Kononova,[2] K. A. Persson,[1], G. Ceder[2] and A. Jain[1]
[1]*Lawrence Berkeley National Laboratory, Energy Technologies Area, 1 Cyclotron Road, Berkeley, CA 94720, United States and*
[2]*Lawrence Berkeley National Laboratory, Materials Science Division, 1 Cyclotron Road, Berkeley, CA 94720, United States*
(Dated: June 4, 2019)

Over the past decades, the number of published materials science articles has increased manyfold. Now, a major bottleneck in the materials discovery pipeline arises in connecting new results with the previously established literature. A potential solution to this problem is to map the unstructured raw-text of published articles onto a structured database entry that allows for programmatic querying. To this end, we apply text-mining with named entity recognition (NER), along with entity normalization, for large-scale information extraction from the published materials science literature. The NER is based on supervised machine learning with a recurrent neural network architecture, and the model is trained to extract summary-level information from materials science documents, including: inorganic material mentions, sample descriptors, phase labels, material properties and applications, as well as any synthesis and characterization methods used. Our classifier, with an overall accuracy ($f1$) of 87% on a test set, is applied to information extraction from 3.27 million materials science abstracts – the most information-dense section of published articles. Overall, we extract more than 80 million materials-science-related named entities, and the content of each abstract is represented as a database entry in a structured format. Our database shows far greater recall in document retrieval when compared to traditional text-based searches due to an entity normalization procedure that recognizes synonyms. We demonstrate that simple database queries can be used to answer complex "meta-questions" of the published literature that would have previously required laborious, manual literature searches to answer. All of our data has been made freely available for bulk download; we have also made a public facing application programming interface (https://github.com/materialsintelligence/matscholar) and website (http://matscholar.herokuapp.com/search) for easy interfacing with the data, trained models and functionality described in this paper. These results will allow researchers to access targeted information on a scale and with a speed that has not been previously available, and can be expected to accelerate the pace of future materials science discovery.

I. INTRODUCTION

Presently, the vast majority of historical materials science knowledge is stored as unstructured text across millions of published scientific articles. The body of research continues to grow rapidly; the magnitude of published data is now so large that individual materials scientists will only access a fraction of this information in their lifetime. With the increasing magnitude of available materials science knowledge, a major bottleneck in materials design arises from the need to connect new results with the mass of previously published literature.

Recent advances in machine learning and natural language processing have enabled the development of tools capable of extracting information from text on a massive scale. Of these tools, named entity recognition (NER) [1] is currently one of the most widely used. Historically, NER was developed as a text-mining technique for extracting information, such as the names of people and geographic locations, from unstructured text, such as newspaper articles [1]. The task is typically approached as a supervised machine learning problem, in which a model learns to identify the key quantities in a sentence; in this way, documents may be represented in a structured format based on the information contained within them. In the past decade, NER has become an important feature for text mining within the chemical sciences [2, 3].

In addition to entity recognition, a major challenge lies in mapping each entity onto a unique database identifier in a process called entity normalization. This issue arises due to the many ways in which a particular entity may be written. For example, "age hardening" and "precipitation hardening" refer to the same process. However, training a machine to recognize this equivalence is especially challenging. A significant research effort has been applied to entity normalization in the biomedical domain; for example, the normalization of gene names has been achieved by using external databases of known gene synonyms [4]. No such resources are available for the materials science domain, and entity normalization has yet to be reported.

In the field of materials science, there has been significant effort to apply NER to extracting inorganic materials synthesis recipes [5–8]; furthermore, a number of chemistry-based NER systems are capable of extracting inorganic materials mentions [9–12]. Some researchers have relied on chemical NER in combination with lookup tables to extract mentions of materials properties or processing conditions [13, 14]. However, there has been no

large-scale effort to extract summary-level information from material science texts. Materials informatics researchers often make predictions for many hundreds or thousands of materials [15, 16], and it would be extremely useful for researchers to be able to ask large-scale questions of the published literature, such as: "*for this list of 10,000 materials, which have been studied as a thermoelectric, and which are yet to be explored?*" An experimental materials science researcher might want to ask: "*what is the most common synthesis method for oxide ferroelectrics? And, give me a list of all documents related to this query*". Currently, answering such questions requires a laborious and tedious literature search, performed manually by a domain expert. However, by representing published articles as structured database entries, such questions may be asked programmatically, and can be answered in a matter of seconds.

In the present report, we apply NER along with entity normalization for large-scale information extraction from the materials science literature. We apply information extraction to over 3.27 million materials science journal articles; we focus solely on the article abstract, which is the most information-dense portion of the article and also readily available from various publisher application programming interfaces (*e.g.*, Scopus). The NER model is a neural network trained using 800 hand-annotated abstracts and achieves an overall $f1$ score of 87%. Entity normalization is achieved using a supervised machine learning model that learns to recognize whether or not two entities are synonyms with an $f1$ score of 95%. We find that entity normalization greatly increases the number of relevant items identified in document querying. The unstructured text of each abstract is converted into a structured database entry containing summary-level information about the document: inorganic materials studied, sample descriptors or phase labels, mentions of any material properties or applications, as well as any characterization or synthesis methods used in the study. We demonstrate how this type of large-scale information extraction allows researchers to access and exploit the published literature on a scale that has not previously been possible. In addition, we release the following data sets: (i) 800 hand-annotated materials science abstracts to be used as training data for NER [17], (ii) JSON files containing details for mapping named entities onto their normalized form [18], and (iii) the extracted named entities, and corresponding digital object identifier (DOI), for 3.27 million materials science articles [19]. Finally, we have released a public facing website and API for interfacing with the data and trained models, as outlined in Section. V.

II. METHODOLOGY

Our full information-extraction pipeline is shown in Fig. 1. Detailed information about each step in the pipeline is presented below, along with an analysis of extracted results. Machine learning models described below are trained using the Scikit-learn [20], Tensorflow [21] and Keras [22] python libraries.

A. Data collection and preprocessing

Document collection. This work focuses on text-mining the abstracts of materials science articles. Our aim is to collect a majority of all English-language abstracts for materials-focused articles published between the years 1900 and 2018. To do this, we create a list of over 1100 relevant journals indexed by Elsevier's Scopus and collect articles published in these journals that fit these criteria via the Scopus and ScienceDirect APIs [24], the Springer-Nature API [25], and web scraping for journals published by the Royal Society of Chemistry [26] and the Electrochemical Society [27]. The abstracts of these articles (and associated metadata including title, authors, publication year, journal, keywords, doi, and url) are then each assigned a unique ID and stored as individual documents in a dual MongoDB/ElasticSearch database. Overall, our corpus contains more than 3.27 million abstracts.

Text preprocessing. The first step in document preprocessing is tokenization, which we performed using ChemDataExtractor [9]. This involves splitting the raw text into sentences, followed by the splitting of each sentence into individual tokens. Following tokenization, we developed several rule-based pre-processing steps. Our preprocessing scheme is outlined in detail in the supplemental material (S.1); here, we provide a brief overview. Tokens that are identified as valid chemical formulae are normalized, such that the order of elements and common multipliers do not matter (e.g. NiFe is the same as Fe50Ni50); this is achieved using regular expression and rule based techniques. Valence states of elements are split into separate tokens (e.g. Fe(III) becomes two separate tokens, Fe and (III)). Additionally, if a token is neither a chemical formula nor an element symbol, and if only the first letter is uppercase, we lowercase the word. This way chemical formulae as well as abbreviations stay in their common form, whereas words at the beginning of sentences as well as proper nouns are converted to lowercase. Numbers with units are often not tokenized correctly with ChemDataExtractor. We address this in the processing step by splitting the common units from numbers and converting all numbers to a special token <nUm>. This reduces the vocabulary size by several tens of thousand words.

Document selection. For this work, we focus on inorganic materials science papers. However, our materials science corpus contains some articles that fall outside the scope of the present work; for example, we consider articles on polymer science or biomaterials to not be relevant. In general, we only consider an abstract to be of interest (*i.e.*, useful for training and testing our NER algorithm) if the abstract mentions at least one inorganic material along with at least one method for the synthe-

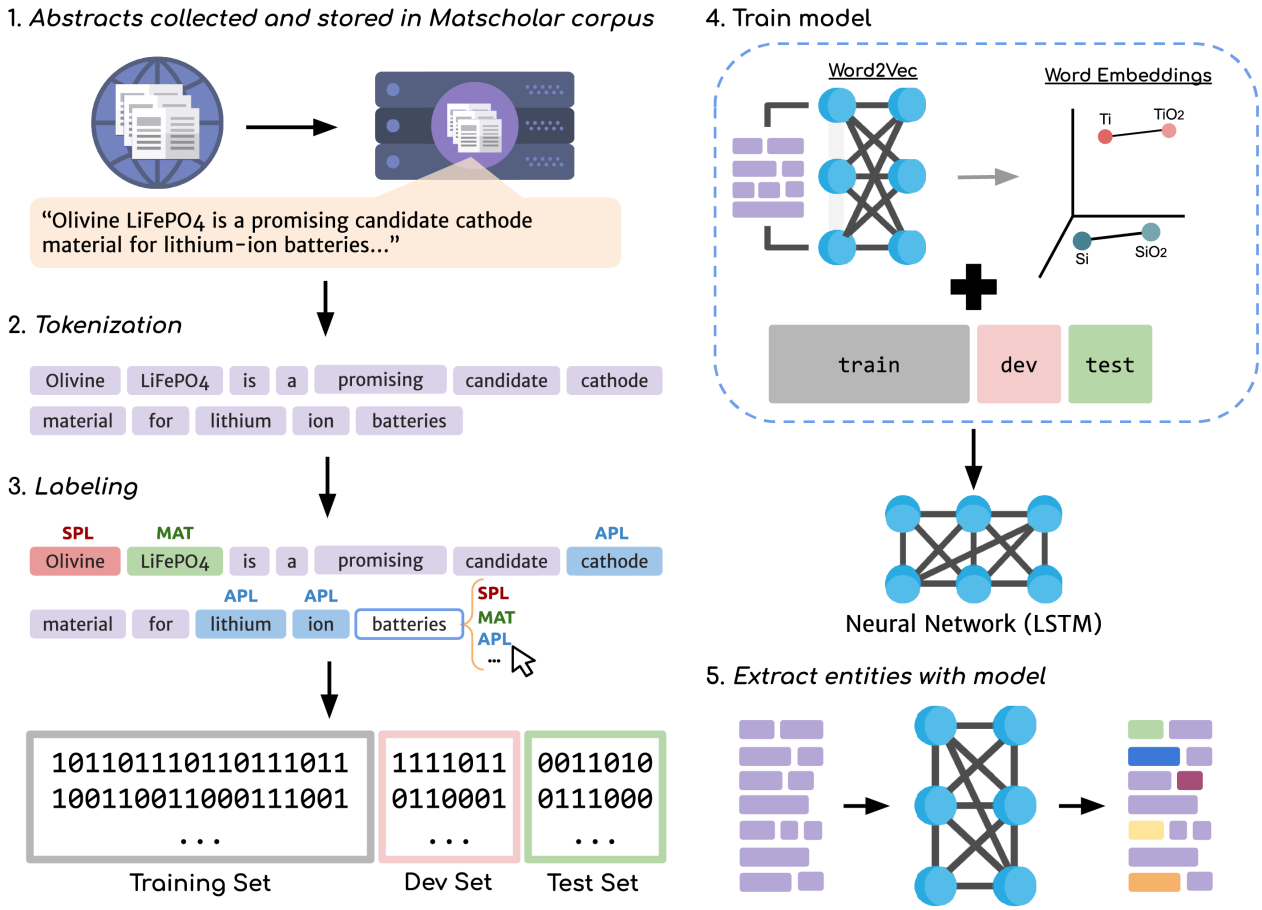

FIG. 1: Workflow for named entity recognition. The key steps are as follows: (i) documents are collected and added to our corpus, (ii) the text is preprocessed (tokenized and cleaned), (iii) for training data, a small subset of documents are labeled (SPL = symmetry/phase label, MAT = material, APL = application), (iv) the labelled documents are combined with word embeddings (Word2vec [23]) generated from unlabelled text to train a neural network for named entity recognition, and finally (v) entities are extracted from our text corpus.

sis or characterization of that material. For this reason, before training an NER model, we first train a classifier for document selection. The model is a binary classifier capable of labeling abstracts as "relevant" or "not relevant". For training data, we label 1094 randomly selected abstracts as "relevant" or "not relevant"; of these, 588 are labeled as "relevant", and 494 are labeled "not relevant". Details and specific rules used for relevance labeling are included in the supplemental material (S.3). The labeled abstracts are used to train a classifier; we use a linear classifier based on Logistic Regression [28], where each document is described by a term frequency – inverse document frequency (tf-idf) vector. The classifier achieves an accuracy ($f1$) score of 89% based on 5-fold cross validation. Only documents predicted to be relevant are considered as training/testing data in the NER study; however, we perform NER over the full 3.27 million abstracts regardless of relevance. We are currently developing text-mining tools that are optimized for a greater scope of topics (*e.g.*, the polymer literature).

B. Named Entity Recognition

Using NER, we are interested in extracting specific entity types that can be used to summarize a document. To date, there have been several efforts to define an ontology or schema for information representation in materials science (see Ref. [29] for a review of these efforts); in the present work, for each document we wish to know what was studied, and how it was studied. To extract this information, we design seven entity labels: inorganic material (MAT), symmetry/phase label (SPL), sample descriptor (DSC), material property (PRO), material application (APL), synthesis method (SMT), and characterization method (CMT). The selection of these labels is somewhat motivated by the well-known material science tetrahedron: "processing", "structure", "properties", and "performance". Examples for each of these tags are given in the supplemental material (S.4), along with a detailed explanation regarding the rules for annotating each of these tags.

Using the tagging scheme described above, 800 materials science abstracts are annotated by hand; only abstracts that are deemed to be relevant based on the relevance classifier described earlier are annotated. The annotations are performed by a single materials scientist. We stress that there is no necessarily "correct" way to annotate these abstracts, however to ensure that the labelling scheme is reasonable, a second materials scientist annotated a subset of 25 abstracts to assess the inter-annotator agreement, which was 87.4%. This was calculated as the percentage of tokens for which the two annotators assigned the same label.

For annotation, we use the inside-outside-beginning (IOB) format [30]. This is necessary to account for multi-word entities, such as "thin film". In this approach, there are special tags representing a token at the beginning (B), inside (I), or outside (O) of an entity. For example, the text fragment "*Thin films of $SrTiO_3$ were deposited*", would be labeled as (token; IOB-tag) pairs in the following way: (Thin; B-DSC), (films; I-DSC), (of; O), ($SrTiO_3$; B-MAT), (were; O), (deposited; O).

Before training a classifier, the 800 annotated abstracts are split into training, development (validation) and test sets. The development set is used for optimizing the hyperparameters of the model, and the test set is used to assess the final accuracy of the model on new data. We use an 80% -10% -10% split, such that there are 640, 80, and 80 abstracts in the train, development, and test sets respectively.

C. Neural Network model

The neural network architecture for our model is based on that of Lample *et al.* [31]. A schematic of this architecture is shown in Fig. 2(a). We explain the key features of the model below.

The aim is to train a model in such a way that materials science knowledge is encoded; for example, we wish to teach a computer that the words "alumina" and "$SrTiO_3$" represent materials, whereas "sputtering" and "MOCVD" correspond to synthesis methods. There are three main types of information that can be used to teach a machine to recognize which words correspond to a specific entity type: (i) word representation, (ii) local (within sentence) context, and (iii) word shape.

For (i), word representation, we use word embeddings. Word embeddings map each word onto a dense vector of real numbers in a high-dimensional space. Words that have a similar meaning, or are frequently used in a similar context, will have a similar word embedding. For example, entities such as "sputtering" and "MOCVD" will have similar vector representations; during training, the model learns to associate these word vectors as synthesis methods. The word embeddings are generated using the Word2vec approach of Mikolov *et al.* [23]. The embeddings are 200-dimensional and are based on the skip-gram approach; word embeddings are generated by

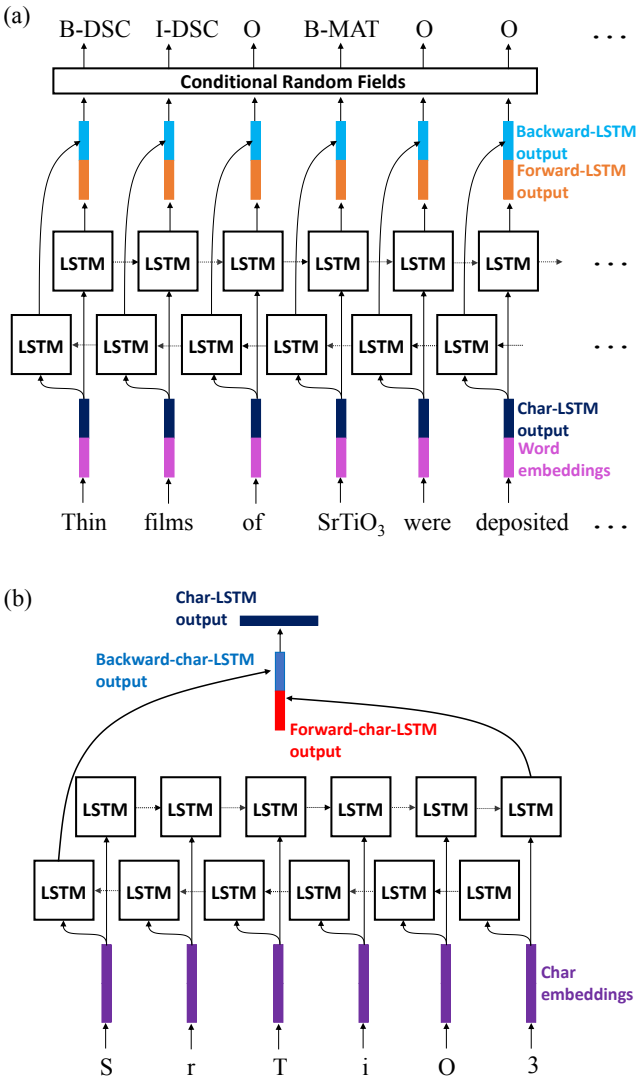

FIG. 2: Neural network architecture for named entity recognition. The colored rectangles represent vectors that are inputs/outputs from different components of the model. The model in (a) represents the word-level bi-directional LSTM, that takes as input a sequence of words, and returns a sequence of entity tags in IOB format. The word-level features for this model are the word embeddings for each word, which are concatenated with the output of the character-level LSTM run over the same word. The character-level LSTM is shown in (b). This model takes a single word such as "$SrTiO_3$", and runs a bi-directional LSTM over each character to encode the morphological properties of each word.

training on our corpus of 3.27 million materials science abstracts. More information about the training of word embeddings is included in the supplemental information (S.2).

For (ii), context, the model considers a sentence as a sequence of words, and it takes into account the local context of each word in the sentence. For example, in the sentence "The band gap of _____ is 4.5 eV", it is quite clear that the missing word is a material, rather

than a synthesis method or some other entity, and this is obvious from the local context alone. To include such contextual information, we use a recurrent neural network (RNN), a type of sequence model that is capable of sequence-to-sequence (many-to-many) classification. As traditional RNNs suffer from problems in dealing with long-range dependencies, we use a variant of the RNN called Long Short-Term Memory (LSTM) [32]. In order to capture both forward and backward context, we use a bi-directional LSTM; in this way, one LSTM reads the sentence forwards, and the other reads it backwards, with the results being combined.

For (iii), word shape, we include character-level information about each word. For example, material formulae like "$SrTiO_3$" have a distinct shape, containing uppercase, lowercase and numerical characters, in a specific order; this word shape can be used to help in entity classification. Similarly, prefixes and suffixes provide useful information about entity type; for example, the suffix "ium", for example in "strontium", is commonly used for elemental metals, and so a word that has this suffix has a good chance of being part of a material name. In order to encode this information into the model, we use a character-level bi-directional LSTM over each word [Fig. 2(b)]. The final outputs from the character-level LSTM are concatenated with the word embeddings for each word; these final vectors are used as the word representations for the word-level LSTM [Fig. 2(a)].

For the word-level LSTM, we use pre-trained word embeddings that have been trained on our corpus of over 3.27 million abstracts. For the character-level LSTM, the character embeddings are not pretrained, but are learned during the training of the model.

The output layer of the model is a Conditional Random Fields (CRF) classifier, rather than a typical softmax layer. Being a sequence-level classifier, the CRF is better at capturing the strong inter-dependencies of the output labels [33].

The model has a number of hyperparameters, including the word- and character-level LSTM size, the character embedding size, the learning rate, and drop out. The NER performance was optimized by repeatedly training the model with randomly selected hyperparameters; the final model chosen was the one with the highest accuracy when assessed on the development set.

D. Entity Normalization

After entity recognition, the final step is entity normalization. This is necessarily required as each entity may be written in numerous forms; for example, "TiO_2", "titania", "AO_2 (A = Ti)" and "titanium dioxide", all refer to the same specific stoichiometry: TiO_2. For document querying, it is important to store these entities in a normalized format, so that a query for documents that mention "titania", also returns documents that mention "titanium dioxide". In order to normalize material mentions, we convert all material names into a canonical normalized formula. The normalized formula is alphabetized and divided by the highest common factor of the stoichiometry. In this way, "TiO_2", "titania", and "titanium dioxide" are all normalized to "O2Ti". In some cases, multiple stoichiometries are extracted from a single material mention; for example, "$Zr_xTi_{1-x}O_3$ (x = 0, 0.5, 1)" is converted to "O2Ti", "O4TiZr" and "O2Zr". When a continuous range is given, e.g, $0 \geq x \leq 1$, we increment over this range in steps of 0.1. Material mentions are normalized using regular expressions, rule-based methods, as well as by making use of the PubChem look up tables [34]; the final validity checks on the normalized formula are performed using the pymatgen library [35].

Normalization of other entity types is also crucial for comprehensive document querying. For example, "chemical vapor deposition", "chemical-vapour deposition", and "CVD" all refer to the same synthesis technique, i.e., they are synonyms for this entity type. In order to determine that two entities have the same meaning, we trained a classifier that is capable of determining whether or not two entities are synonyms.

The model uses the word embeddings for each entity as features; after performing NER, each multi-word entity is concatenated into a single word, and new word embeddings are trained such that every multi-word entity has a single vector representation. For synonym detection, each data instance is an entity pair, and so the word embeddings for the two entities are concatenated before being fed into the model. In addition to the word embeddings, which mostly capture the context in which an entity is used, several other hand-crafted features are included (supplemental S.5). To train the model, 10000 entity pairs are labelled as being either synonyms or not (see supplemental S.6). Using this data, a binary random forest classifier is trained to be able to predict whether or not two entities are synonyms of one another.

Using the synonym classifier, each extracted entity can be normalized to a canonical form. Each entity is stored as its most frequently occurring synonym (we exclude acronyms as a normalized form); for example, "chemical vapor deposition", "chemical-vapour deposition", and "CVD" are all stored as "chemical vapor deposition", as this is the most frequently occurring synonym that is not an acronym.

III. RESULTS

A. NER classifier performance

Using the trained classifier, we are able to accurately extract information from materials science texts. The NER classifier performance is demonstrated in Fig. 3; the model can clearly identity the key information in the text.

Model performance can be assessed more quantitatively by assessing the accuracy on the development and

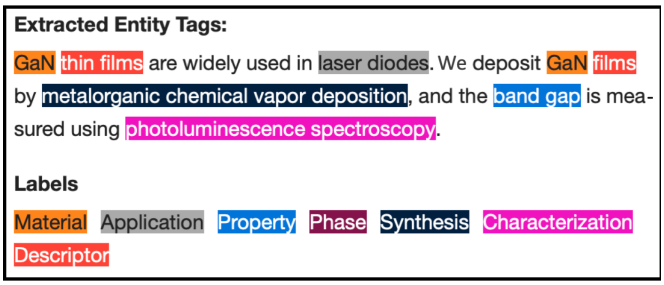

FIG. 3: Example predictions of the materials science NER classifier. The highlighting indicates regions of text that the model has associated with a particular entity type.

test sets. We define our accuracy using the $f1$ score, which is the harmonic mean of precision (p) and recall (r),

$$p = \frac{t_p}{t_p + f_p}, \tag{1}$$

$$r = \frac{t_p}{t_p + f_n}, \tag{2}$$

$$f1 = 2\frac{p \cdot r}{p + r}, \tag{3}$$

where, t_p, f_p, and f_n represent the number of true positives, false positives, and false negatives, respectively. We use the CoNLL scoring system, which requires an exact match for the entire multi-word entity [36]; for example, if a CMT (characterization method) is labeled "photoluminescence spectroscopy", and the model only labels the "photoluminescence" portion as a CMT, the entire entity is considered as being labeled incorrectly.

The accuracy of our neural network model is presented in Table I. We include the overall $f1$ score as well as the score for each entity type for both the development and test sets. The accuracy on the development and test sets is similar, suggesting that the model has not been overfit to the development set during hyperparameter tuning. The overall $f1$ score on the test set is 87.04%. This score is fairly close to the current state of the art NER system (90.94%) [31] based on the same neural network architecture (which is trained and evaluated on hand-labelled newspaper articles [36]). However, we caution against a direct comparison of scores on these tasks, as the modeling of materials science text is expected to be quite different than newspaper articles. The $f1$ score of 90.30% for inorganic material extraction is as good or better than previously reported chemical NER systems. For example, Mysore *et al.* achieved an $f1$ score of 82.11% for extraction material mentions [6]; Swain *et al.* reported an $f1$ score of 93.4% for extracting chemical entity mentions [9], however their model is not specifically trained to identify inorganic materials. The $f1$ scores for other entity tags are all over 80%, suggesting that the model is performing well at extracting each entity type. Materials mentions and sample descriptors have the highest scores; this is most likely due to their frequency in the training set and because these entities are often single tokens. We note that this work makes available all 800 hand-labelled abstracts for the testing of future algorithms against the current work.

TABLE I: Accuracy metric $f1$ for named entity recognition on the development and test sets. Results are shown for the total, material (MAT), phase (SPL), sample descriptor (DSC), property (PRO), application (APL), synthesis method (SMT), and characterization (CMT) tags. Also shown is the total number extracted for each entity type over the full corpus of abstracts.

	Accuracy ($f1$)		
Label	Dev. set	Test set	Extracted (millions)
Total	87.09	87.04	81.23
MAT	92.58	90.30	19.07
SPL	85.24	82.05	0.53
DSC	91.40	92.13	9.36
PRO	80.19	83.19	31.00
APL	80.60	80.63	7.46
SMT	81.32	81.37	5.01
CMT	86.52	86.01	8.80

B. Entity normalization

Following entity extraction, the next step is entity normalization. The trained random forest binary classifier exhibits an $f1$ score of 94.5% for entity normalization. The model is assessed using 10-fold cross-validation with a 9000:1000 train/test split. The precision and recall of the model are 94.6% and 94.4%, respectively, when assessed on this test set. While these accuracy scores are high, we emphasize that the training/test data are generated in a synthetic manner (see supplemental material S.6), such that the test set has roughly balanced classes. In reality, the vast majority of entities are not synonyms. By artificially reducing the number of negative test instances, the number of false positives is also artificially reduced, and so the actual precision and $f1$ scores are overestimated. In production, we find that the model is prone to making false-positive predictions. In terms of false positives, the two most common mistakes made by the synonym classifier are: (i) when two entities have an opposite meaning but are used in a similar context, e.g. "p-type doping" and "n-type doping", or (ii) when two entities refer to related but not identical concepts, e.g, "hydrothermal synthesis" and "solvothermal synthesis".

In order to overcome the large number of false positives, we perform a final, manual error check on any

normalized entities. If an entity is incorrectly normalized, the incorrect synonym is manually removed, and the entity is converted to the most frequently occurring correct prediction of the ML model. With this manual error check, the precision of the overall approach is close to 100%. The full list of normalized entities is provided as Supporting Information.

C. Text-mined information extraction

We run the trained NER classifier described in Sec. II C over our full corpus of 3.27 million materials science abstracts. The entities are normalized using the procedure described in Sec. II D. Overall, more than 80 million materials-science-related entities are extracted; a quantitative breakdown of the entity extraction is shown in Table I.

After entity extraction, the unstructured raw-text of an abstract can be represented as a structured database entry, based on the entity tags contained within it. We represent each abstract as a document in a non-relational (NoSQL) MongoDB [37] database collection. This type of representation allows for efficient document retrieval and allows the user to ask summary-level or "meta-questions" of the published literature. In section IV, we demonstrate several ways by which large-scale text mining allows for researchers to access the published literature programmatically.

IV. APPLICATIONS

A. Document retrieval

Online databases of published literature work by allowing users to query based on some keywords. In the simplest case, the database will return documents that contain an exact match to the keywords in the query. More advanced approaches may employ some kind of "fuzzy" matching; for instance, many text-based search engines are built upon Apache Lucene [38], which allows for fuzzy matching based on edit distance. While advanced approaches to document querying have been developed, to date there has been no freely-available "materials-science aware" database of published documents; i.e., there is no materials science knowledge base for the published literature. For example, querying for "$BaTiO_3$" in Web of Science returns 19,134 results, where as querying for "barium titanate" returns 12,986 documents, and querying for "$TiBaO_3$" returns only a single document. Ideally, all three queries should return the same result (as is the case for our method), as all three queries are based on the same underlying stoichiometry.

We find that that by encoding materials science knowledge into our entity extraction and normalization scheme, performance in document retrieval can be significantly increased. In Fig. 4, the effectiveness of our normalized entity database for document retrieval is explored. To test this, we created two databases, in which: (i) each document is represented by the *normalized* entities extracted, and (ii) each document is represented by the *non-normalized* entities extracted. We then randomly choose an entity of any type, and query each database based on this entity to see how many documents are returned. The process of querying on a randomly selected entity is repeated 1000 times to generate statistics on the efficiency of each database in document retrieval. This test is also performed for two-entity queries, by randomly selecting two entities and querying each database. We note that because the precision of of our manually checked normalization is close to unity, this test effectively measures the relative recall (Eq. 2) of each database (by comparing the false-negative rate).

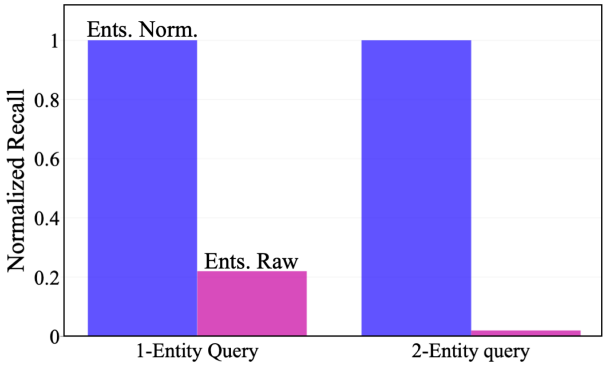

FIG. 4: Recall statistics for document retrieval are generated by performing 1000 queries for each for each database. Results are shown for the entities that have been properly normalized using the scheme described in Sec. II D (Ents. Norm), as well as for non-normalized entities (Ents. Raw).

From Fig. 4, it is clear that normalization of the entities is crucial for achieving high recall in document queries. When querying on a single entity, the recall for the normalized database is about 5 times higher than for the non-normalized case. When querying on two entities this is compounded, and the non-normalized database only returns about 2% of the documents that are returned from the normalized database. Our results highlight that a domain-specific approach such as the one we described can greatly improve performance. As described in Section V, we have built a materials-science aware document search engine that can be used to more comprehensively access the materials science literature.

B. Material search

In the field of materials informatics, researchers often use machine learning models to predict chemical compositions that may be useful for a particular application. A practical major bottleneck in this process comes from needing to understand which of these chemical compositions have been previously studied for that application,

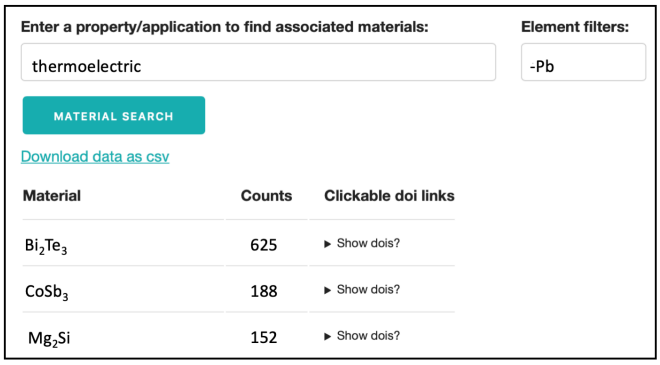

FIG. 5: A screenshot of the materials search functionality on our web application, which allows one to search for materials associated with a specified keyword (ranked by count of hits) and with element filters applied. The application returns clickable hyperlinks so that users can directly access the original research articles.

which typically requires painstaking manual literature review. In section I, we posed the following example question "*for this list of 10,000 materials, which have been studied as a thermoelectric, and which are yet to be explored?*". Analysis of the extracted named entities can be a useful tool in automating the literature review process.

In Fig 5, we illustrate how one can analyze the co-occurrence of various property/application entities with specific materials on a large scale. Our web application has a "Materials Search" functionality. Users can enter a specific property or application entity, and the Materials Search will return a table of all materials that co-occur with that entity, along with document counts and the corresponding DOIs. As demonstrated, the results can be further filtered based on composition; in the example shown in Fig 5, all of the known thermoelectrics compositions that do *not* contain lead are returned. The results can then by downloaded in comma-separated values (CSV) format, to allow for programmatic literature search.

C. Guided literature searches and summaries

One major barrier for accessing the literature is that researchers may not know *a priori* what are the relevant terms or entities to search for a given material. This is especially true for researchers that may be entering a new field, or may be studying a new material for the first time. In Fig. 6, we demonstrate how "materials summaries" can be automatically generated based on the entities. The summaries show common entities that co-occur with a material; in this way, a researcher can learn how a material is typically synthesized, common crystal structures, or how and for what application a material is typically studied. The top entities are presented along with a score (0 – 100%), indicating how frequently each entity co-occurs with a chosen material. We note that the summaries are automatically generated without any human intervention or curation.

As an example, in Fig. 6 summaries are presented for PbTe and $BiFeO_3$. PbTe is most frequently studied as a thermoelectric [39], and so many of the entities in the summary deal with the thermoelectric and semiconducting properties of the material, including dopability. PbTe has a cubic (rock salt, fcc) structure as the ground state crystal structure under ambient conditions [40]. PbTe does have some metastable phases such as orthorhombic [41], and this is also reflected in the summary; some of the less-frequently-occurring phases (*i.e.*, scores less than 0.01) arise simply due to co-occurrence with other materials, indicating a weakness of the co-occurrence assumption.

The summary in Fig. 6(b) for $BiFeO_3$ mostly reflects the extensive research into this material as a multiferroic (i.e., a ferromagnetic ferroelectric) [42]. The summary shows that $BiFeO_3$ has several stable or metastable polymorphs, including the rhombohedral (hexagonal) and orthorhombic [43], as well as tetragonal [44] and others; these phases are all known and have been studied in some form. $BiFeO_3$ is most frequently synthesized by either sol - gel or solid state reaction routes, which is common for oxides.

The types of summaries described above are very powerful, and can be used by non-domain experts to get an initial overview of a new material or new field, or for domain experts to gain a more quantitative understanding of the published literature or broader study of their topic (*e.g.*, learning about other application domains that have studied their target material). For example, knowledge of the most common synthesis techniques is often used to justify the choice of chemical potentials in thermodynamic modelling, and these quantities are used in calculations of surface chemistry [45] and charged point defects [46] in inorganic materials.

We note that there may be some inherent bias in the aforementioned literature summaries that may arise due to the analysis of only the abstracts. For example, more novel synthesis techniques such as "molecular beam epitaxy" may be more likely to be mentioned in the abstract than a more common technique such as "solid-state reaction". This could be easily overcome by performing our analysis on full text rather than just abstracts (see Sec. VI for a discussion on the current limitations and future work).

D. Answering meta-questions

Rather than simply listing which entity tags co-occur with a given material, one may want to perform more complex analyses of the published literature. For example, one may be interested in multifunctional materials; a researcher might ask "*which materials have the most diverse range of applications?*". To answer such a question, the co-occurrence of each material with the different

(a) **PbTe**

Property	score
thermoelectric	16.78
thermoelectric properties	13.11
structure	11.48
thermal conductivity	11.38
composition	10.90
n - type	10.32
seebeck coefficient	10.22
p - type	9.93

Application	score
thermoelectric applications	2.99
quantum wells	2.41
electrode	1.25
electrolyte	1.06
diodes	0.96
multiple quantum wells	0.96
thermoelectric generators	0.77
thermoelectrics	0.68

Phase	score
cubic	5.21
rock salt	1.93
fcc	0.87
rhombohedral	0.87
orthorhombic	0.87
hexagonal	0.68
zinc - blende	0.58
trigonal	0.48
pnma	0.48

Characterization	score
x-ray diffraction	13.40
scanning electron microscopy	6.56
transmission electron microscopy	6.08
density functional theory	3.09
energy dispersive x-ray spectroscopy	2.31
electron microscopy	2.22

Synthesis	score
doping	7.14
molecular beam epitaxy	4.53
annealing	4.44
spark plasma sintering	3.38
hot pressing	3.18
sintering	3.09
alloying	2.99
bridgman method	2.12

Sample Descr.	score
substrate	17.45
doped	16.39
films	15.62
layer	13.40
alloy	11.76
surface	9.26
bulk	9.06
single crystal	9.06
crystals	8.20

(b) **$BiFeO_3$**

Property	score
structure	30.79
magnetic	27.14
multiferroic	24.66
magnetic properties	23.00
ferroelectric	20.37
dielectric constant	14.83
ferromagnetic	13.59
dielectric	12.47
crystal structure	12.25

Application	score
multiferroics	5.00
photocatalyst	3.06
electrode	2.90
multiferroic materials	2.79
catalyst	1.93
capacitors	1.24
spintronic devices	1.02
device applications	0.64
sensor	0.59

Phase	score
rhombohedral	22.94
perovskite	16.87
r3c	10.32
orthorhombic	9.19
tetragonal	6.99
cubic	3.82
rhombohedral r3c	1.99
hexagonal	1.67
monoclinic	1.50

Characterization	score
x-ray diffraction	43.31
scanning electron microscopy	12.74
raman	12.20
transmission electron microscopy	7.79
magnetic measurements	6.99
x-ray photoelectron spectroscopy	6.18

Synthesis	score
sol – gel	16.98
doping	15.69
solid - state reaction	11.12
annealing	7.36
sintering	6.39
pulsed laser deposition	4.73
hydrothermal method	4.67
co-doping	4.62

Sample Descr.	score
doped	24.23
ceramics	22.94
films	21.28
thin films	21.23
substrate	19.94
nanoparticles	13.49
single - phase	10.26
powder	8.70
polycrystalline	8.38

FIG. 6: Automatically generated materials summaries for PbTe (a) and $BiFeO_3$ (b). The score is the percentage of papers that mention each entity at least once.

applications can be analyzed.

In order to ensure that materials with diverse applications are uncovered, we first perform a clustering analysis on the word embeddings for each application entity that has been extracted. Clustering on the full 200-dimensional word embeddings gives poor results – this is common for clustering of high-dimensional data. Improved results are obtained by reducing the dimensions of the embeddings using Principal Component Analysis (PCA) [47] and t-Distributed Stochastic Neighbor Embedding (t-SNE) [48]. It is found that t-SNE provides qualitatively much better clusters than PCA; t-SNE does not necessarily preserve the data density or neighbor distances, however empirically this approach often provides good results when used with certain clustering algorithms. Each word embedding is first reduced to three dimensions using t-SNE [48], and then a clustering analysis is performed using Density-based Spatial Clustering of Applications with Noise (DBSCAN) [49]. The results of this analysis is shown in Fig. 7(a) (the data is reduced to two dimensions for the purposes of plotting). The DBSCAN algorithm finds 244 distinct application clusters. In Fig. 7(a), the applications found in one small cluster are highlighted – all of the entities in this cluster relate to solid-state memory applications.

To determine which materials have the most diverse applications, we look at co-occurrence of each material with entities from each cluster. The top 5000 most commonly occurring materials and applications are considered. We generate a co-occurrence matrix, with rows representing materials and columns representing application clusters. A material is considered to be relevant to a cluster if it co-occurs with an application from that cluster in at least 100 abstracts. The results of this analysis is are presented in Fig. 7(b). Based on this analysis, the most widely used materials are SiO_2, $Al2O_3$, TiO_2, steel and ZnO. In Fig. 7(c) and (d), the most important applications for SiO_2 and steel are shown in a pie chart. The labels for each segment of the chart are created by manually inspecting the entities in each cluster (we note that automated topic modeling algorithms could also be used to create a summary of each application cluster [50], not pursued here). For SiO_2, the most important applications are in catalysis and complimentary metal-oxide semiconductor (CMOS) technologies (SiO_2 is an important dielectric in CMOS devices); for steel, the most important applications are in various engineering applications. However, both materials have widely varied applications across disparate fields.

Being able to answer complex meta-questions of the published literature can be a powerful tool for researchers. With all of the corpus represented as a structured database, generating an answer to these questions requires only a few lines of code, and just a few seconds for data transfer and processing – this is in comparison to a several-weeks-long literature search performed by a

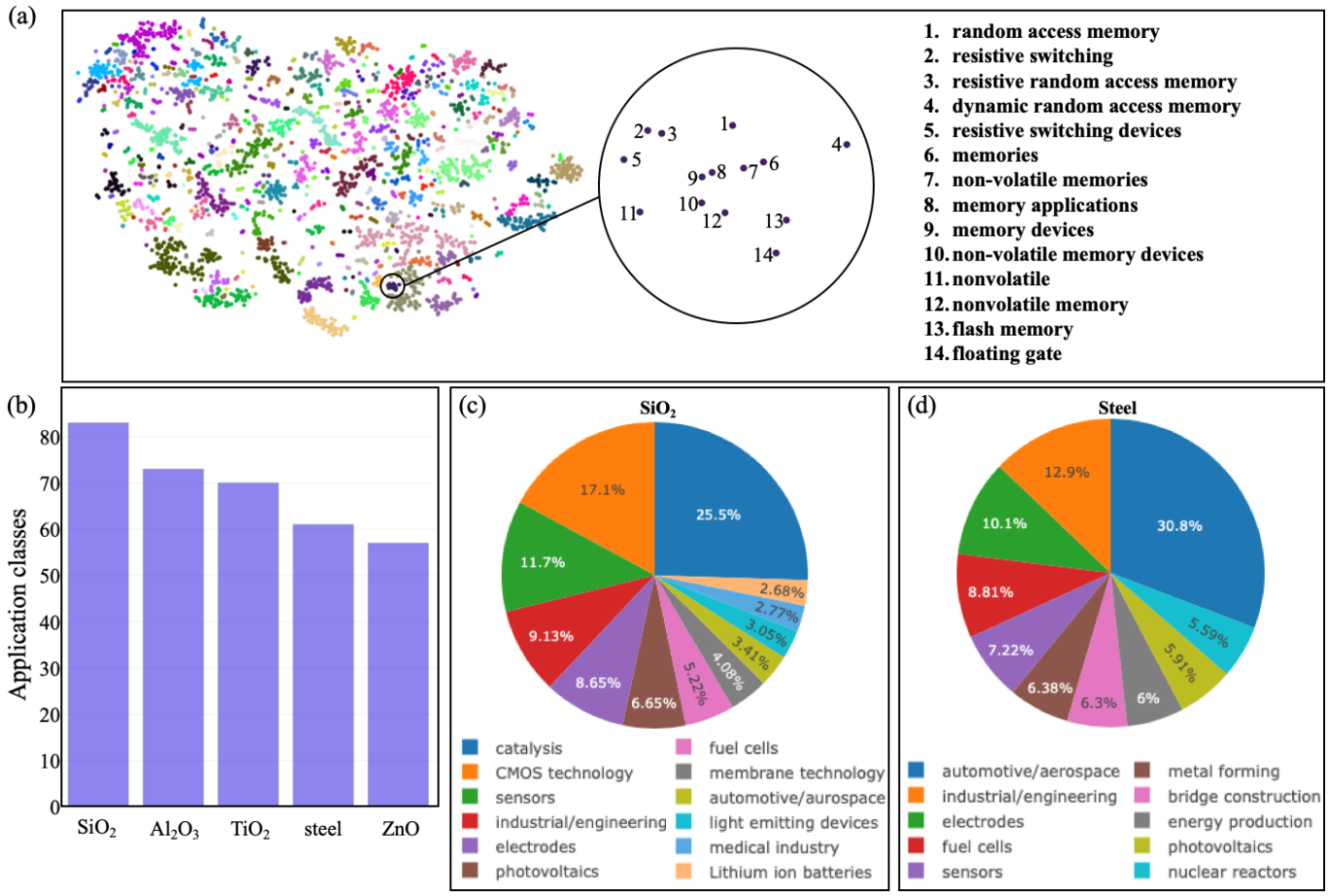

FIG. 7: (a) Word embeddings for the 5000 most commonly mentioned applications are projected onto a two-dimensional plane. The coloring represent clusters identified by the DBSCAN algorithm. The applications from one of these clusters are highlighted – all applications in this cluster relate to solid-state memory devices. In (b), the materials with the largest number of applications are plotted; the same is plotted in (c), however the data has been normalized to account for the total number of material mentions.

domain expert that would have been previously required.

V. DATABASE AND CODE ACCESS

All of the data associated with this project has been made publicly available, including the 800 hand-annotated abstracts used as training data for NER [17], as well as JSON files containing the information for entity normalization [18]. The entire entity database of 3.27 million documents is available online for bulk download in JSON format [19].

We have also created a public facing API to access the data and functionality developed in this project; the code to access the API is available via the Github repository (https://github.com/materialsintelligence/matscholar). The API allows users to not only access the entities in our data set, but also to perform NER on any desired materials science text. Finally, we have developed a web application, http://matscholar.herokuapp.com/search, for non-programmatic access to the data. The web application has various materials-science aware tools for accessing the published literature.

VI. SUMMARY AND DISCUSSION

Using the information extraction pipeline shown in Fig. 1, we have performed information extraction on over 3.27 million materials science abstracts with a high degree of accuracy. Over 80 million materials-science-related entities were extracted, and each abstract is represented as a structured document in a database collection. The entities in each document are normalized (*i.e.*, equivalent terms grouped together), greatly increasing the number of relevant results in document retrieval. Text-mining on such a massive scale provides tremendous opportunity for accessing information in materials science.

One of the most powerful applications of text mining is in information retrieval, which is the process of accurately retrieving the correct documents based on a given

query. In the field of biomedicine, the categorization and organization of information into a knowledge base has been found to greatly improve efficiency in information retrieval [51, 52]. Combining ontologies with large databases of published literature allows researchers to impart a certain semantic quality onto queries, which is far more powerful than simple text-based searches that are agnostic to the domain. We have stored the digital object identifier (DOI) for each abstract in our corpus, such that targeted queries can be linked to the original research articles. In addition to information retrieval, an even more promising and novel aspect of text mining on this scale, is the potential to ask "meta-questions" of the literature, as outlined in Sec. IV B and Sec. III C. The potential to connect the extracted entities of each document and provide summary-level information based on an analysis of many thousands of documents can greatly improve the efficiency of the interaction between researchers and the published literature.

We should note some limitations of the present information extraction system, and the prospects for improvement. First, certain issues arise as a result of only analyzing the abstracts of publications. Not all information about a publication is presented in the abstract; therefore, by considering only the abstract, the recall in document retrieval can be reduced. Second, the quantitative analysis of the literature, as in Fig. 6, can be affected. This can be overcome by applying our NER system to the full text of journal articles. The same tools developed here can be applied to full text, provided that the full text is freely available. A second issue with the current methodology is that the system is based on co-occurrence of entities, and we do not perform explicit entity relation extraction. By considering only co-occurrence, recall is not affected, but precision can be reduced. We do note that all of the other tools available for querying the materials science literature rely on co-occurrence.

Despite these limitations, we have demonstrated the ways in which large-scale text mining can be used to develop research tools, and the present work represents a first step towards artificially intelligent and programmatic access to the published materials science literature.

VII. ACKNOWLEDGMENTS

This work was supported by Toyota Research Institute through the Accelerated Materials Design and Discovery program. We are thankful to Matthew Horton for useful discussions. The abstract data was downloaded from Scopus API between October 2017 and September 2018 via http://api.elsevier.com and http://www.scopus.com.

[1] D. Nadeau and S. Sekine, Lingvist. Investig. **30**, 3 (2007).
[2] I. Segura-Bedmar, P. Martínez, and M. H. Zazo, in *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)* (2013), vol. 2, pp. 341–350.
[3] S. Eltyeb and N. Salim, J. Cheminformatics **6**, 17 (2014).
[4] H.-r. Fang, K. Murphy, Y. Jin, J. S. Kim, and P. S. White, in *Proceedings of the Workshop on Linking Natural Language Processing and Biology: Towards Deeper Biological Literature Analysis* (Association for Computational Linguistics, 2006), pp. 41–48.
[5] E. Kim, K. Huang, A. Saunders, A. McCallum, G. Ceder, and E. Olivetti, Chem. Mater. **29**, 9436 (2017).
[6] S. Mysore, E. Kim, E. Strubell, A. Liu, H.-S. Chang, S. Kompella, K. Huang, A. McCallum, and E. Olivetti, arXiv preprint arXiv:1711.06872 (2017).
[7] E. Kim, K. Huang, S. Jegelka, and E. Olivetti, npj Computational Materials **3**, 53 (2017).
[8] E. Kim, K. Huang, A. Tomala, S. Matthews, E. Strubell, A. Saunders, A. McCallum, and E. Olivetti, Scientific data **4**, 170127 (2017).
[9] M. C. Swain and J. M. Cole, J. Chem. Inf. Model. **56**, 1894 (2016).
[10] T. Rocktäschel, M. Weidlich, and U. Leser, Bioinformatics **28**, 1633 (2012).
[11] R. Leaman, C.-H. Wei, and Z. Lu, J. Cheminformatics **7**, S3 (2015).
[12] M. Krallinger, O. Rabal, A. Lourenco, J. Oyarzabal, and A. Valencia, Chemical reviews **117**, 7673 (2017).
[13] C. J. Court and J. M. Cole, Scientific data **5**, 180111 (2018).
[14] S. Shah, D. Vora, B. Gautham, and S. Reddy, Integrating Materials and Manufacturing Innovation **7**, 1 (2018).
[15] B. Meredig, A. Agrawal, S. Kirklin, J. E. Saal, J. W. Doak, A. Thompson, K. Zhang, A. Choudhary, and C. Wolverton, Phys. Rev. B **89**, 094104 (2014).
[16] F. Legrain, J. Carrete, A. van Roekeghem, G. K. Madsen, and N. Mingo, J. Phys. Chem. B **122**, 625 (2017).
[17] `https://doi.org/10.6084/m9.figshare.818442`.
[18] `https://doi.org/10.6084/m9.figshare.8184365`.
[19] `https://doi.org/10.6084/m9.figshare.8184413`.
[20] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al., J. Mach. Learn. Res. **12**, 2825 (2011).
[21] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al., in *OSDI* (2016), vol. 16, pp. 265–283.
[22] F. Chollet et al., *Keras* (2015).
[23] T. Mikolov, K. Chen, G. Corrado, and J. Dean, arXiv preprint arXiv:1301.3781 (2013).
[24] `https://dev.elsevier.com/`.
[25] `https://dev.springernature.com/`.
[26] `http://www.rsc.org/`.
[27] `https://www.electrochem.org/`.
[28] D. W. Hosmer Jr, S. Lemeshow, and R. X. Sturdivant, *Applied logistic regression*, vol. 398 (John Wiley & Sons, 2013).
[29] X. Zhang, C. Zhao, and X. Wang, Comput. Ind. **73**, 8

(2015).
[30] L. A. Ramshaw and M. P. Marcus, in *Natural language processing using very large corpora* (Springer, 1999), pp. 157–176.
[31] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, arXiv preprint arXiv:1603.01360 (2016).
[32] S. Hochreiter and J. Schmidhuber, Neural Comput. **9**, 1735 (1997).
[33] Z. Huang, W. Xu, and K. Yu, arXiv preprint arXiv:1508.01991 (2015).
[34] S. Kim, P. A. Thiessen, E. E. Bolton, J. Chen, G. Fu, A. Gindulyte, L. Han, J. He, S. He, B. A. Shoemaker, et al., Nucleic Acids Res. **44**, D1202 (2015).
[35] S. P. Ong, W. D. Richards, A. Jain, G. Hautier, M. Kocher, S. Cholia, D. Gunter, V. L. Chevrier, K. A. Persson, and G. Ceder, Comp. Mater. Sci. **68**, 314 (2013).
[36] E. F. Tjong Kim Sang and F. De Meulder, in *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4* (Association for Computational Linguistics, 2003), pp. 142–147.
[37] `https://www.mongodb.com/`.
[38] `https://lucene.apache.org/`.
[39] Y. Pei, A. LaLonde, S. Iwanaga, and G. J. Snyder, Energy Environ. Sci. **4**, 2085 (2011).
[40] K. M. Rabe and J. D. Joannopoulos, Phys. Rev. B **32**, 2302 (1985).
[41] Y. Wang, X. Chen, T. Cui, Y. Niu, Y. Wang, M. Wang, Y. Ma, and G. Zou, Phys. Rev. B **76**, 155127 (2007).
[42] J. Wang, J. Neaton, H. Zheng, V. Nagarajan, S. Ogale, B. Liu, D. Viehland, V. Vaithyanathan, D. Schlom, U. Waghmare, et al., Science **299**, 1719 (2003).
[43] D. C. Arnold, K. S. Knight, F. D. Morrison, and P. Lightfoot, Phys. Rev. Lett. **102**, 027602 (2009).
[44] Z. Fu, Z. Yin, N. Chen, X. Zhang, Y. Zhao, Y. Bai, Y. Chen, H.-H. Wang, X. Zhang, and J. Wu, Applied Physics Letters **104**, 052908 (2014).
[45] K. Reuter and M. Scheffler, Phys. Rev. B **65**, 035406 (2001).
[46] L. Weston, A. Janotti, X. Y. Cui, C. Stampfl, and C. G. Van de Walle, Phys. Rev. B **89**, 184109 (2014).
[47] I. Jolliffe, *Principal component analysis* (Springer, 2011).
[48] L. v. d. Maaten and G. Hinton, J. Mach. Learn. Res. **9**, 2579 (2008).
[49] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al., in *Kdd* (1996), vol. 96, pp. 226–231.
[50] D. M. Blei, A. Y. Ng, and M. I. Jordan, Journal of machine Learning research **3**, 993 (2003).
[51] H.-M. Müller, E. E. Kenny, and P. W. Sternberg, PLoS Biol. **2**, e309 (2004).
[52] O. Bodenreider, Yearb. Med. Inform. **17**, 67 (2008).