Title

Quantifying the advantage of domain-specific pre-training on named entity recognition tasks in materials science

Journal

Patterns, 3(4)

ISSN

2666-3899

Authors

Trewartha, Amalie
Walker, Nicholas
Huo, Haoyan
et al.

Publication Date

2022-04-01

DOI

10.1016/j.patter.2022.100488

Quantifying the Advantage of Domain-Specific Pre-Training on Named Entity Recognition Tasks in Materials Science

Amalie Trewartha[b,1,3], Nicholas Walker[a,1,2,3,*], Haoyan Huo[b,3], Sanghoon Lee[a,3], Kevin Cruse[b,3], John Dagdelen[a,3], Alexander Dunn[a,3], Kristin A. Persson[a,4], Gerbrand Ceder[b,4], Anubhav Jain[a,4]

[a]*Lawrence Berkeley National Laboratory, Energy Technologies Area, 1 Cyclotron Road, Berkeley, 94720, CA, United States of America*
[b]*Lawrence Berkeley National Laboratory, Materials Science Division, 1 Cyclotron Road, Berkeley, 94720, CA, United States of America*

Abstract

A bottleneck in efficiently connecting new materials discoveries to established literature has arisen due to a massive increase in publications. This problem may be addressed by using named entity recognition (NER) to extract structured summary-level data from unstructured materials science text. We compare the performance of four NER models on three materials science datasets. The fours models include a BiLSTM and three Transformer models (BERT, SciBERT, and MatBERT) with increasing degrees of domain-specific materials science pre-training. MatBERT improves over the other two $\text{BERT}_{\text{BASE}}$-based models by $1 \sim 12\%$, implying that domain-specific pre-training provides measurable advantages. Despite its relative architectural simplicity, the BiLSTM model consistently outperforms the original BERT model, perhaps due to its domain-specific pre-trained word embeddings. Furthermore, MatBERT and SciBERT models outperform the original BERT model to a greater extent in the small data limit. MatBERT's higher quality predictions should accelerate the extraction of structured data from materials science literature.

Keywords: Natural Language Processing, NLP, Named Entity Recognition, NER, BiLSTM, BERT, Transformer, Language Model, Pre-train, Materials Science, Solid State, Doping, Gold Nanoparticles
PACS: 07.00.00, 81.00.00

1. Introduction

Recently, the number of publications in the field of materials science has grown exponentially.[1] As a result, it has become increasingly difficult for researchers to follow research progress as it emerges, even within relatively restricted sub-domains. The size of the materials science literature means that even relatively simple questions, such as which material candidates have previously been studied for a particular application, can be difficult or impossible to comprehensively answer. This has created a need for new, more efficient ways to engage with the literature and extract the relevant information therein.

Natural Language Processing (NLP), the analysis of unstructured text using computers, provides a natural candidate for such an alternative approach. NLP has successfully been applied to a number of materials science applications and is the topic of several recent investigations in materials informatics.[2, 3, 4, 5] Additionally, work has been done to develop meta-learning strategies for NER.[6, 7, 8] Recently, the advent of Transformer ML architectures such as BERT [9] have revolutionized NLP; leading benchmarks such as GLUE [10] are now dominated by models utilizing attention-based encoder-decoder architectures called Transformers[11] and perform comparably to humans on some tasks. Transformer models have ushered in a new NLP paradigm where large and general NLP models are

*Corresponding author
Email address: walkernr@lbl.gov (Nicholas Walker)
[1]First author
[2]Lead contact
[3]These authors contributed equally
[4]Senior author

'pre-trained' on semi-supervised tasks before being fine-tuned for downstream tasks.[9, 12, 13, 14, 15, 16, 17] The pre-training approach allows for task-specific models to be trained using relatively few hand-annotated examples; this is a useful feature for practical applications of NLP bottlenecked by annotation such as scientific tasks which contain technical text and esoteric vocabulary.

Although a single pre-trained model may address multiple NLP tasks (e.g., question answering, named entity recognition, next sentence prediction), the success of models with domain-specific pretraining such as BioBERT[18], CaseHOLD[19], and FinBERT[20] begs the question: can Transformer models be further improved with even *more* domain-specific pretraining? We hypothesize that the measurable advantages previously shown with domain-specific pretraining - for example, of SciBERT over BERT[21] - can again be extended to models specific to narrower scientific disciplines such as materials science. Improved domain-specific model performance implies improved ability for automated knowledge extraction from even the most complex and vexing (from the perspective of NLP models) scientific domains. Exploring this problem in-depth presents an opportunity for the collation and synthesis of massive numbers of highly complex scientific publications into otherwise inaccessible structured databases and models for knowledge generation.

In this work, we apply Transformer models to the task of Named Entity Recognition (NER)[22] in order to extract and label important scientific entities relevant to materials chemistry from unstructured text. A well-trained NER model will be capable of automatically mapping the unstructured text of materials science publications to a queryable database of key terms. Historically, NER has been used to extract information such as names and locations from various articles, though recently it has been employed in the chemical, medical, and materials sciences as well.[23, 24, 25, 26, 27, 28, 2, 29, 30, 1, 3, 31, 32, 33, 34, 35, 36, 37, 4, 38, 39, 40] For material science, this may include terms that refer to materials and their geometries, properties, syntheses, methods of characterization, and downstream applications. Strongly related work in text mining and language modeling has also been employed in the same fields.[41, 42, 43, 44, 45, 46, 5, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62] BERT has additionally found use in biology, medicine, and materials science.[18, 63, 64]

Specific to the field of materials science, there have been significant efforts to apply NER to the extraction of materials synthesis recipes, including using BERT.[28, 2, 29, 62, 64] In the past, these have employed a combination of the aforementioned work in the chemical sciences to extract inorganic material entities with syntax trees and lookup tables to extract properties and processing conditions. The recently developed Transformer-based models have been shown to offer significant performance improvements on NLP tasks.[9] This provides an excellent opportunity to evaluate the performance of these new models on NER tasks specific to materials science.

In this work, we apply four different NER models to three different materials science datasets and analyze their performance. The simplest model considered is a bidirectional long short-term memory (BiLSTM) recurrent neural network. The other three models, variants of the popular Transformer-based $\text{BERT}_{\text{BASE}}$ neural network structure[9], have identical model structures but use pre-training corpora of varying domain specificity. The considered datasets consist of one general-purpose materials science dataset (referred to as the solid state dataset) and two topic-specific datasets that respectively focus on doping and gold nanoparticle synthesis. We use the results of NER on these materials science datasets to relate the domain specificity of the pre-training corpus to measurable performance differences in extracting named entities.

2. Results

2.1. Datasets

Here we consider three different NER datasets, chosen to represent a diversity of text sources and problems relevant to materials science; a set of solid state materials science abstracts with entities of broad interest,[28] a set of abstracts with inorganic doping information, and a set of methods/results sections relevant to gold nanoparticle synthesis. Each of these is described in detail below. The solid state data set is publicly available,[65] though only the DOIs and annotated entities are available for the other two.[66]

2.1.1. Solid State Dataset

The solid state dataset discussed in this work consists of 800 annotated abstracts from solid state materials publications collected using Elsevier's Scopus/ScienceDirect [67] and Springer-Nature [68] APIs as well as web scraping

for journals published by the Royal Society of Chemistry [69] and the Electrochemical Society.[70] Abstracts are considered relevant if they mention at least one inorganic material and at least one synthesis or characterization method for inorganic materials. The entity labels are chosen to represent a broad domain of materials science knowledge with eight different labeled entity types; inorganic materials (MAT), symmetry/phase labels (SPL), sample descriptors (DSC), material properties (PRO), material applications (APL), synthesis methods (SMT), and characterization methods (CMT). Details of the collection and pre-processing of these abstracts, and detailed definitions of the labels are available in Weston et al.[28]

A condensed example is shown in Figure 1.[71] This dataset is intended to provide a "catch-all" of relevant information without focusing on any specific facet of solid state materials. Due to the broad definitions of the entities, the solid state dataset generally contains more entities per paragraph than the other datasets. Additionally, an inter-annotator agreement of 87.4% was evaluated utilizing 25 annotations from a second annotator.[28]

2.1.2. Doping Dataset

The properties of doped materials used for applications requiring semiconductors are determined by critical pieces of information such as the base material (BASEMAT), the doping agent (DOPANT), and quantities associated with the doped material such as the doping density or the charge carrier density (DOPMODQ). The intention of this dataset is to capture the information relevant to the doping of a material and any other relevant quantitative measurements. Abstracts that specifically contain information about doping, i.e., those containing regular expressions matching "dop*" (such as "dopant", "doped", and "co-doping") or "n-type" or "p-type", were queried from the Matscholar database of materials science abstracts.[72] A set of 500 abstracts was randomly sampled from the queried set, from which 455 abstracts were identified by human annotators as relevant to inorganic materials science and were annotated by three annotators.

A condensed example is shown in Figure 2.[73] As opposed to the solid-state and gold nanoparticle dataset, tokens were annotated one sentence at a time (one sample = one sentence). Sentences were annotated only when they contain specific and direct information about the doping of solid state materials, e.g., "X was doped with Y," "X:Y," or "Y doping.". Sentences describing byproducts or targeted properties (e.g., magnetization) without direct reference to a dopant or a host material (e.g., "The layered TiO2 phase did not incorporate the dopant specie and had an anatase structure with measured lattice parameters of a = 3.61Å, c = 9.45Å.") were not annotated.

2.1.3. Gold Nanoparticle Dataset

Gold nanoparticles (AuNPs) are used widely in biomedicine (e.g., in vitro diagnostics), semiconductor technology, and cosmetics.[74, 75, 76, 77, 78] Despite the strong reliance of AuNP properties on size and shape,[79, 80] only recently have synthesis methods been able to control AuNP morphology, particularly anisotropic nanorods. This dataset aims to capture gold nanoparticle morphologies and descriptions from relevant sections of the full text of gold nanoparticle synthesis literature. A single annotator annotated a set of 85 characterization paragraphs from 73 articles on gold nanoparticle synthesis.

A condensed example is shown in Figure 3.[81] The entities for this model include general shape-based morphological information for the synthesized gold nanoparticles, including noun-based morphological entities (MOR) and adjective-based, descriptive entities (DES). Entities like "particle" or "AuNP" were annotated as MOR entities so that at least some target could be identified with which to attribute size information in the future since many nanoparticle articles only refer to the particles as the less descriptive "nanoparticle" or "NP." Note that other aspects such as the dimensions of particles were not included due to very low levels of support for such labels in the original data. This is similar to past work on information extraction from nanomaterial synthesis literature.[62] Furthermore, limiting the number of labels will tend to provide better performance, particularly for smaller datasets.

2.2. Methods

Four different models are trained and evaluated on each dataset, including a bidirectional long short-term memory network (BiLSTM) and three variations of networks using the bidirectional encoder representations from Transformers (more specifically, BERT) structure. The three BERT networks considered include $\text{BERT}_{\text{BASE}}$ (uncased), SciBERT (uncased),[21] and a pre-trained model introduced with this work, MatBERT (uncased). Each model, when given an abstract for a materials science publication in the form of a sequence of tokens, learns to classify each token into predefined categories. The token categories correspond to combinations of token position and entity type, i.e. $B-MAT$

for the beginning token of a material entity. In this way, the named entity recognition models described here can be understood as sequence-to-sequence models (Seq2Seq) which transform a sequence of words into a sequence of labels. Unless otherwise specified, for each experiment, 80% of the data was used for training, 10% for validation, and 10% for testing. Sixteen different seeds (integer powers of 2 from 0 to 15) were used to determine the order of the training data as well as the model weight initialization.

2.2.1. Tokenizers

The Materials Tokenizer was used with the BiLSTM model.[28] First, the tokenization step is carried out using ChemDataExtractor with additional pre-processing to split tokens that are either composed of a number and a unit or an element and a valence state.[82] Processing the tokens then consists of filtering numbers to become "⟨nUm⟩" since they are often not tokenized correctly with ChemDataExtractor, normalizing simple chemical formulas such that the order of the elements is standardized, lowercasing tokens with only the first letter capitalized that are not elements or chemical formulas, and removing accents.

BERT models, however, use the WordPiece subword tokenization algorithm, which is very similar to Byte-Pair Encoding (BPE).[83, 84] BPE relies on a pre-tokenizer that splits the training data into words. After determining the unique words in the training data and their frequencies, BPE constructs a base vocabulary consisting of all symbols that occur in the words and is trained to learn merging rules such that two symbols from the base vocabulary can be combined to form a new symbol until the vocabulary has grown to the desired size. The learned merging rules can then be applied to new words as long as they are composed of symbols from the base vocabulary. In contrast to BPE, WordPiece learns symbol pairs that maximize the likelihood of the training data rather than the most frequent symbol pairs.

2.2.2. Tagging Schemes

This work uses the IOBES tagging scheme.[85] With this scheme, any token that does not correspond to an entity (or part of an entity) is labeled with O, denoting an "outside" classification. Single-token entities will be labeled $S-X$ where the $S-$ prefix denotes a "single" token entity and the X is the entity type. For multi-token entities, the prefix $B-$ is used to denote the "beginning" token, $E-$ for the "end" token, and $I-$ for the tokens "inside" the span of the beginning and end tokens. The IOBES tagging scheme has been shown to provide higher F-scores than other similar tagging schemes while retaining the ability to identify consecutive entities.[86]

2.2.3. Conditional Random Field

For all of the models considered, a conditional random field (CRF) is utilized for decoding sequences in addition to calculating the training and validation loss, taking the classification layer output logits as inputs.[87, 88, 89, 90] As opposed to a classification layer that outputs logits to predict labels without the consideration of neighboring labels, a CRF layer is capable of taking context from these neighboring labels into account when making predictions. Invalid transitions as defined by the tagging scheme (such as $I-X$ being followed by $B-X$) are initialized to incur large loss penalties.

2.2.4. BiLSTM Model

The BiLSTM network is an example of a gated recurrent neural network (RNN) in which the connections between the nodes in the LSTM layers compose a directed graph along a temporal sequence, in this case, a sequence of words. This allows the network to track arbitrarily long-term dependencies in the input sequence, demonstrating temporal dynamic behavior. The bidirectional implementation allows for the LSTM layers to consider both the forward and backward directions of the sequence. Multi-head attention is also used in order to allow the network to attend to different parts of the sequence differently, i.e. responding to longer-term vs. shorter-term dependencies.[11] These dependency-sensitive representations of the tokens in the sequence can then be used for the downstream classification task via a classification layer. In this work, the word embeddings are initialized using pre-trained Mat2Vec embeddings with a vocabulary size of $529,688$.[91] During training, additional word features are learned using character-level convolutions. These features are then concatenated with the pre-trained Mat2Vec embeddings before being fed into the LSTM layers. The character-level convolutions can aid in improving embeddings for infrequent or even out-of-vocabulary words and have been shown to be useful on relatively small benchmark datasets.[92]

Table 1 summarizes the parameters used to construct the BiLSTM model. The only change in comparison to the BiLSTM model used in past work is the use of convolutional layers instead of BiLSTM layers for the character fields.[28] For training the BiLSTM model with CRF output and loss, the pre-trained Mat2Vec embeddings were held constant by convention. The RangerLARS optimizer (also known as Over9000),[93] a combination of a rectified adaptive moment estimation (RAdam)[94] and Lookahead[95] to produce the Ranger optimizer[96] alongside least-angle regression (LARS)[97], was used for all experiments. A learning rate schedule called "flat and anneal" was utilized, which consists of a constant learning rate for 72% of the training epochs followed by cosine annealing to decay the learning rate to 0.[93] An initial learning rate of $4 \cdot 10^{-2}$ was used alongside gradient clipping with a maximum norm of 1.0 in order to prevent exploding gradients. The training was conducted for 64 epochs and the embeddings were held frozen throughout training.

2.2.5. BERT Models

The three BERT models we investigate share the same $\text{BERT}_{\text{BASE}}$ network structure as well as the same tokenizer algorithm with a maximum vocabulary size of $30,552$ tokens. Input sequences are limited to a maximum of 512 tokens. Refer to the original BERT paper for details on its architecture.[9]

Table 2 summarizes the parameters used to construct the $\text{BERT}_{\text{BASE}}$ model. The three BERT models considered in this work differ only in pre-training, which is largely determined by the corpora that they are trained on. Before training the actual BERT model parameters can take place, the WordPiece tokenizer must be trained on the corpora in order to establish the vocabulary of the model. After the tokenizer is trained, the corresponding BERT model is pre-trained on the same corpora. This consists of two tasks: masked language modeling (MLM) and next sentence prediction (NSP).[9] The MLM task requires that the BERT model predicts missing words in input sequences where 15% of the words are masked. The NSP task requires that given two sequences, the BERT model predicts the likelihood that one follows the other. It has been shown that pre-training on different corpora can lead to different performances.[21] This is of particular interest in technical fields where commonly used words and phrases may not be well-represented or even carry the same meaning in other contexts.

The original BERT model was trained on the BooksCorpus (800 million tokens) and English Wikipedia (2.5 billion tokens).[9] By contrast, SciBERT was trained on 1.14 million scientific papers from Semantic Scholar (3.1 billion tokens) across a variety of fields.[21] SciBERT was shown to outperform BERT on scientific tasks as a result.

Building on this, we present MatBERT as a BERT model trained using scientific papers specifically from the field of materials science. For training MatBERT, we randomly sampled 2 million papers, or around 61 million paragraphs, from a corpus mostly consisting of peer-reviewed materials science journal articles.[2] To optimize MatBERT models for materials science terminologies, two WordPiece tokenizers (cased and uncased) were trained using these paragraphs with no additional pre-processing. Following BERT practices, the vocabulary sizes for the tokenizers are both 30,522. After tokenization, paragraphs with less than 20 or more than 510 tokens were removed, leaving a pretraining corpus consisting of around 50 million paragraphs (8.8 billion tokens). The two variants were trained using only the masked language modeling (MLM) task. An AdamW optimizer was used with a weight decay of 0.01 and the learning rate of $5 \cdot 10^{-5}$ decayed linearly to zero during five training epochs. A batch size of 192 paragraphs per gradient update step was used. The convergence of the MLM loss v.s. training steps can be found in Supplementary Materials. Each model was trained on 8 NVIDIA V100 GPUs and took about one month to complete. The pre-training code and pre-trained MatBERT model weights are publicly available.[98, 99] In this work, the uncased version is used for all BERT variants.

For training of the BERT models (MatBERT, SciBERT, and BERT) with CRF output and loss, the pre-trained model parameters were fine-tuned. The model structures as well as the BERT pre-trained parameters were provided by the "transformers" library.[100] The SciBERT pre-trained parameters compatible with this library were acquired using the SciBERT AllenAI repository.[21] All experiments were performed using the PyTorch library.[101] The LAMB optimizer was used for all experiments.[102] Different initial learning rates for the BERT embeddings, BERT encoders, and the classification layers (the linear and CRF layers) were employed to reach optimum results. They were respectively chosen as $1 \cdot 10^{-4}$, $2 \cdot 10^{-3}$, and $1 \cdot 10^{-2}$. For the first epoch, only the classification layers are trained, after which the BERT layers are fine-tuned alongside the classification layers for four epochs. For the learning rate schedule, all learning rates are subjected to exponential decay to 10% of the initial value at the final epoch, starting at the end of the second epoch. Gradient clipping with a maximum norm of 1.0 was employed in order to prevent exploding gradients. For the BERT models (MatBERT, SciBERT, and BERT), the WordPiece tokenizer will often

split up words into multiple subtokens. For label predictions, only the embedding of the first subtoken of each word is used for classification. This is consistent with conventional usage.[9] The code used for training the BERT models on the NER tasks is publicly available.[103]

2.3. NER Results

In this section, model performances on the aforementioned datasets are reported along with model performance as a function of dataset size. An input sample consists of an entire paragraph from the dataset. The model classification performances are judged according to their achieved precision, recall, and F1-scores using the "micro" averaging scheme to accurately reflect the class imbalances in the datasets. In all experiments, the set of parameters at the end of an epoch that results in the best validation F1-score are evaluated on the test set. In all experiments, training was carried out for 64 epochs for the BiLSTM model and 5 epochs for the BERT, SciBERT, and MatBERT models. We reiterate that the only difference between the BERT models considered here is the choice of pre-training corpus.

In Figure 4, the performances of the models on the considered datasets are shown. Each point on the scatter plot depicts the 95% confidence interval (assuming a normal distribution) across 16 seeds for the chosen metric, model, and dataset. The precision is the ratio of correctly predicted entities to all predicted entities and the recall is the ratio of correctly predicted entities to all true entities. The F1-score is the harmonic mean of the precision and recall.

In Figure 4a, it is shown that the MatBERT and SciBERT models perform better than the BERT and BiLSTM models (within statistical error as shown by the confidence intervals) on the solid state set as determined by the F1-score. For precision, recall, and F1-score, the MatBERT model performs slightly better than the SciBERT model. Interestingly, although the BERT and BiLSTM models achieve very similar F1-scores, there is actually a tradeoff between the precision and recall with the models, as the BiLSTM model achieves higher precision whereas the BERT model achieves higher recall. This means that the BiLSTM model is less susceptible to predicting false positives while the BERT model is less susceptible to predicting false negatives. The precision and recall are much closer in value for the BERT model than for the BiLSTM model.

Furthermore, in Figure 4b, the same metrics for the doping dataset are shown. Once again, the MatBERT and SciBERT models perform better than the BERT and BiLSTM models. Additionally, the MatBERT model once again demonstrates better performance than the SciBERT model for precision, recall, and F1-score. Compared to the BERT model, the BiLSTM model achieves slightly higher precision (0.71 ± 0.03 vs. 0.70 ± 0.02). The respective performances are nearly identical for the recall (0.68 ± 0.03) and F1-score (0.69 ± 0.02). However, the confidence intervals are slightly higher with the BiLSTM model.

Finally, in Figure 4c, the same metrics are once again shown for the gold nanoparticle dataset. The MatBERT model again achieves a higher F1-score than the other models, but for this dataset, the BiLSTM model and the SciBERT model achieve a similar F1-score with the BERT model trailing behind. For the recall, it can be seen that the BERT model performs significantly worse than the other models, with the MatBERT model achieving the best performance followed by the BiLSTM model and then the SciBERT model in turn. For the precision, all of the models perform similarly, with the BERT model actually achieving the best performance, followed by the MatBERT model and then the SciBERT model with the BiLSTM model trailing.

Figure 5 shows a heatmap of the entity-wise average F1-scores attained for each model across the datasets. The highest score for each entity is in bold. MatBERT claims the best performance for all entities except for one, solid state material descriptions (DSC), where it only slightly lags behind SciBERT. SciBERT then claims the second-best performance for the rest of the entities aside from gold nanoparticle descriptions (DES), which the BiLSTM instead claims. Between the BiLSTM and the original BERT, the BiLSTM generally performs better across the entities, only performing much worse compared to BERT for dopant quantities (DOPMODQ), slightly trailing behind BERT for the solid state applications (APL), solid state properties (PRO), solid state synthesis methods (SMT), and dopants (DOPANT) entities and performing much better for the solid state symmetry/phase labels (SPL), doping base materials (BASEMAT), gold nanoparticle descriptions (DES), and gold nanoparticle morphologies (MOR) entities. Of particular interest is the very poor score of zero obtained by BERT on the gold nanoparticle descriptions (DES) entity, which was caused by the failure to predict any entities. Since SciBERT also scored poorly on the gold nanoparticle descriptions (DES) entity (0.29), with the BiLSTM (0.53) and MatBERT (0.67) models significantly outperforming BERT and SciBERT, this would suggest that the domain-specific pre-training is important to gold nanoparticle descriptions (DES) entity recognition performance.

Generally, the models tend to consistently perform better or worse on the same entities. All of the models tended to perform the poorest on the doping base materials (BASEMAT), dopant quantities (DOPMODQ), and gold nanoparticle descriptions (DES) entities and the best on the solid state descriptions (DSC) and solid state materials (MAT) entities. There are some exceptions, however, with BERT performing relatively poorly on the symmetry/phase label (SPL) and gold nanoparticle morphologies (MOR) entities despite very good performances from the other models. The model performances on the gold nanoparticle descriptions (DES) entity vary far more than on the other entities, with very large performance gaps between the models.

To study the effect of the number of training examples on model performance, we plot learning curves for each model on each dataset in Figure 6. Curating and annotating even modestly-sized datasets can entail considerable effort from domain experts in physics, chemistry, and materials science due to the highly technical nature of many publications in those fields. This is in contrast to canonical NER tasks such as CoNLL-2003[104] (a NER set used in the original BERT publication[9]) which aim to identify less technical entities such as organizations, people, or places. Thus, models which can perform well on small training datasets will be of interest to domain experts looking to create structured technical datasets from text using NER.

In Figure 6, we observe MatBERT and SciBERT exhibiting large performance improvements over BERT at low numbers of training samples, in particular with fewer than 200 samples for the solid state dataset and with fewer than 50 samples with the gold nanoparticle dataset. The BiLSTM model exhibits the best performance as the training set size approaches zero, but asymptotically approaches a lower limit than the SciBERT and MatBERT models as the number of training points increases. On the solid state dataset, the larger number of annotated examples allows for BERT to close the gap in F1-score such that the confidence intervals are overlapping at 400 samples and are indistinguishable at 600 samples. As opposed to the SciBERT and MatBERT models, however, BERT does not exceed the BiLSTM performance at any of the training sample intervals for any task. This is not to imply that BERT is approaching the same limit as the BiLSTM; rather, we expect that as the number of training samples is further increased, the general BERT model will exceed or reach the BiLSTM due to its much more complex architecture as seen with the solid state dataset (though this is less clear for the two smaller datasets). Determining whether adding more NER training data for any one task will outweigh the effects of domain-specific pretraining - that is, whether the general BERT model will overlap SciBERT or MatBERT - requires further investigation with larger numbers of annotated technical text samples. Generally, we observe that more specific pretraining results in increased performance (by substantial margins, e.g., $\sim$ 0.05 micro F1-score improvement of MatBERT over general BERT at 320 solid-state training samples) for BERT-derived models at every training set size, particularly at small training set sizes.

Another contributing factor to the difference in performance is class support (the number of labels in the testing dataset for a given class). Figure 7 illustrates the disparity among entities' F1-score by class support for each of the three datasets. As expected, classes with higher support generally have higher F1-scores and classes with low support stratify according to the level of pre-training. We would intuitively expect MatBERT to perform much better on rarely mentioned entities than BERT given its higher exposure to materials-related text during pre-training. This can be readily seen with the gold nanoparticle dataset's description entity (DES) and the doping dataset's dopant quantity entity (DOPMODQ), in which model performances suffer likely suffer from very low support (respectively $\sim$ 10 and $\sim$ 20). For the gold nanoparticle dataset's description entity (DES) which has the lowest support, the models pre-trained on materials-related text perform significantly better than those trained on general scientific text or just general text. However, the large degree of stratification among BERT models for entities with higher support is of note. Particularly for the solid state dataset's property entity (PRO, e.g., "Voight-Reuss-Hill average bulk moduli") with a relatively large level of support ($\sim$ 700 samples), MatBERT and SciBERT both make a substantial $\sim$ 0.03 and 0.04 F1-score improvement over BERT. This improvement may imply that highly-specialized entities, such as materials science properties which do not appear frequently in general corpora but appear frequently in domain-specific corpora, benefit the most from more specialized pre-training even when there are relatively many samples for fine-tuning. For entities which are more commonly mentioned in general text corpora, such as AuNP morphologies (MOR) (e.g., "particles", "rods", "spheres") doping quantities (DOPMODQ) (e.g., "3%"), and solid state descriptions (DSC) (e.g., "crystalline", "amorphous", "powder") the level of pre-training appears less important at every level of support.

3. Discussion

Whether domain-specific pre-training is needed for large Transformer models remains an open question in the field of NLP. Although large models trained on massive general-purpose corpora are complex enough to allow for fine-tuning for various downstream tasks (question/answer, NSP, NER) as opposed to expensive from-scratch re-training, our results show evidence that domain-specific pre-training can measurably improve F1-score performance in the domain of materials science. The overall best performance of MatBERT across the three materials science datasets corroborates a growing body of evidence that domain-specific pre-training is not only a trivial improvement over generally pre-trained models but is indeed worth the effort of retraining large models like BERT. For instance, BioBERT[18] demonstrated as much as 2.8% F1-score improvement over BERT in the biomedical domain; similarly, both CaseHOLD[19] (legal corpora) and FinBERT[20] (financial corpora) yield improvements over base BERT in their respective domains' downstream tasks. The word distribution shift from a general-purpose corpus to an exclusively technical corpus is large enough to encourage full re-training of large Transformer models.

Our results now introduce the question: How specialized should a pre-training corpus be so that it is both highly performant within a domain of knowledge and general enough to address a variety of NER problems within that domain? Although MatBERT improves on BiLSTM, SciBERT, and BERT for all but the smallest training set sizes, the MatBERT model we introduce is limited by the distribution of pre-training data. As detailed in Methods, pre-training data was taken from a general material science corpus.[2] However, as shown by the most frequent title keywords in Fig. 8, this corpus is designed to be biased towards trending materials science topics describing experimental syntheses. For example, paragraphs from full texts tend to favor popular compounds (such as oxides, energy materials, and magnetic materials) or synthesis techniques (such as conventional solid-state or hydrothermal synthesis). The MatBERT pre-training corpus, therefore, puts less weight on computational papers containing density functional theory results, theoretical but yet-to-be-synthesized stoichiometries, and unusual but important phase labels. Thus, MatBERT may be improved by expanding the pre-training corpus beyond the set compiled in Kononova et al.[2] The goal in selecting a pre-training corpus should be to strike a balance between the specificity needed to capture particular facets of materials science and transferability between disparate fields within materials science. Exploring other methods to sample the materials science literature for the purposes of model training is one possible avenue for future work.

As seen in the presented results and ensuing discussion, the MatBERT model achieves the best overall performance out of the considered models. The $1 \sim 4\%$ F1-score improvement over SciBERT demonstrates that domain-specific pre-training provides a measurable advantage for NER in materials science. Furthermore, SciBERT improving upon BERT by $3 \sim 9\%$ F1-score reinforces the importance of scientific pre-training in general for materials science text. Interestingly, it was even found that a comparatively simple BiLSTM model enhanced with embeddings pre-trained on materials science text provides better overall performance than the original BERT model. This suggests that pre-training on a domain-specific corpus can be more impactful on performance than employing modern large transformer-based models. Learning curves additionally show that in the low data limit, the BiLSTM outperforms the BERT models, albeit still with poor overall performance due to the lack of data. For larger datasets, though, MatBERT provides a definitive improvement in NER predictions that can be expected to accelerate the construction of structured materials science datasets.

4. Experimental Procedures

4.1. Resource Availability

4.1.1. Lead Contact

Requests for additional information should be directed to the lead contact, Nicholas Walker (walkernr@lbl.gov).

4.1.2. Materials Availability

This study did not generate physical materials.

4.1.3. Data and Code Availability

The pre-trained MatBERT model as well as the trained MatBERT NER models are publicly available. The code used to pre-train MatBERT is publicly available. The code used to train MatBERT NER is publicly available. The DOIs of the articles used for the new datasets alongside the associated extracted entities are publicly available.

5. Supplemental Information

Supplemental information can be found online at [PLACEHOLDER].

6. Acknowledgments

This work was funded by Toyota Research Institute through the Accelerated Materials Design and Discovery program. Secondary funding to develop the gold nanoparticle dataset as well as MatBERT pre-training was provided for this work by the U.S. Department of Energy, Office of Science, Office of Basic Energy Sciences, Materials Sciences and Engineering Division under Contract No. DE-AC02-05-CH11231 (D2S2 program KCD2S2). This research used resources of the National Energy Research Scientific Computing Center (NERSC), a U.S. Department of Energy Office of Science User Facility operated under Contract No. DE-AC02-05CH11231. This work also used the Extreme Science and Engineering Discovery Environment (XSEDE) GPU resources, specifically the Bridges-2 supercomputer at the Pittsburgh Supercomputing Center, through allocation TG-DMR970008S.

7. Author Contributions

A.J., G.C., and K.A.P. supervised the research. J.D. wrote the data collection infrastructure and performed the data collection. S.L., N.W., A.D., and J.D. annotated the doping dataset. K.C. annotated the gold nanoparticle dataset. H.H. wrote the MatBERT pre-training code and performed the pre-training. N.W., A.T., K.C., S.L., and A.D. wrote the MatBERT NER training code. N.W. wrote the BiLSTM NER training code. N.W. performed the NER experiments and prepared the results. N.W., A.D., and H.H. prepared the figures. All authors contributed to the discussion and writing of the manuscript.

8. Declaration of Interests

The authors declare no competing interests.

References

[1] O. Kononova, T. He, H. Huo, A. Trewartha, E. A. Olivetti, G. Ceder, Opportunities and challenges of text mining in materials research, iScience 24 (2021) 102155. URL: https://www.sciencedirect.com/science/article/pii/S2589004221001231. doi:https://doi.org/10.1016/j.isci.2021.102155.

[2] O. Kononova, H. Huo, T. He, Z. Rong, T. Botari, W. Sun, V. Tshitoyan, G. Ceder, Text-mined dataset of inorganic materials synthesis recipes, Scientific Data 6 (2019) 203. URL: https://doi.org/10.1038/s41597-019-0224-1. doi:10.1038/s41597-019-0224-1.

[3] E. A. Olivetti, J. M. Cole, E. Kim, O. Kononova, G. Ceder, T. Y.-J. Han, A. M. Hiszpanski, Data-driven materials research enabled by natural language processing and information extraction, Applied Physics Reviews 7 (2020) 041317. URL: https://doi.org/10.1063/5.0021106. doi:10.1063/5.0021106. arXiv:https://doi.org/10.1063/5.0021106.

[4] M. Krallinger, O. Rabal, F. Leitner, M. Vazquez, D. Salgado, et al., The chemdner corpus of chemicals and drugs and its annotation principles, J. Cheminform. 7 (2015) S2. doi:10.1186/1758-2946-7-S1-S2.

[5] H. Gurulingappa, A. Mudi, L. Toldo, M. Hofmann-Apitius, J. Bhate, Challenges in mining the literature for chemical information, RSC Advances 3 (2013) 16194. doi:10.1039/c3ra40787j.

[6] J. Li, P. Han, X. Ren, J. Hu, L. Chen, S. Shang, Sequence labeling with meta-learning, IEEE Transactions on Knowledge and Data Engineering (2021) 1–1. doi:10.1109/TKDE.2021.3118469.

[7] J. Li, B. Chiu, S. Feng, H. Wang, Few-shot named entity recognition via meta-learning, IEEE Transactions on Knowledge and Data Engineering (2020) 1–1. doi:10.1109/TKDE.2020.3038670.

[8] J. Li, S. Shang, L. Chen, Domain generalization for named entity boundary detection via metalearning, IEEE Transactions on Neural Networks and Learning Systems 32 (2021) 3819–3830. doi:10.1109/TNNLS.2020.3015912.

[9] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. arXiv:1810.04805.

[10] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, S. R. Bowman, GLUE: A multi-task benchmark and analysis platform for natural language understanding, 2019. In the Proceedings of ICLR.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, 2017. arXiv:1706.03762.

[12] J. Howard, S. Ruder, Universal language model fine-tuning for text classification, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 328–339. URL: https://www.aclweb.org/anthology/P18-1031. doi:10.18653/v1/P18-1031.

[13] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, Deep contextualized word representations, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 2227–2237. URL: https://www.aclweb.org/anthology/N18-1202. doi:10.18653/v1/N18-1202.

[14] B. McCann, J. Bradbury, C. Xiong, R. Socher, Learned in translation: Contextualized word vectors, 2018. arXiv:1708.00107.

[15] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, A. Bordes, Supervised learning of universal sentence representations from natural language inference data, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Copenhagen, Denmark, 2017, pp. 670–680. URL: https://www.aclweb.org/anthology/D17-1070. doi:10.18653/v1/D17-1070.

[16] K. Zhang, S. Bowman, Language modeling teaches you more than translation does: Lessons learned through auxiliary syntactic task analysis, in: Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 359–361. URL: https://www.aclweb.org/anthology/W18-5448. doi:10.18653/v1/W18-5448.

[17] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language models are few-shot learners, 2020. arXiv:2005.14165.

[18] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, J. Kang, Biobert: a pre-trained biomedical language representation model for biomedical text mining, Bioinformatics (2019). URL: http://dx.doi.org/10.1093/bioinformatics/btz682. doi:10.1093/bioinformatics/btz682.

[19] L. Zheng, N. Guha, B. R. Anderson, P. Henderson, D. E. Ho, When does pretraining help? assessing self-supervised learning for law and the casehold dataset, 2021. arXiv:2104.08671.

[20] D. Araci, Finbert: Financial sentiment analysis with pre-trained language models, 2019. arXiv:1908.10063.

[21] I. Beltagy, K. Lo, A. Cohan, SciBERT: A pretrained language model for scientific text, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 3615–3620. doi:10.18653/v1/D19-1371.

[22] J. Li, A. Sun, J. Han, C. Li, A survey on deep learning for named entity recognition, 2020. arXiv:1812.09449.

[23] S. Eltyeb, N. Salim, Chemical named entities recognition: a review on approaches and applications, Journal of cheminformatics 6 (2014) 17–17. URL: https://pubmed.ncbi.nlm.nih.gov/24834132. doi:10.1186/1758-2946-6-17, 24834132[pmid].

[24] P. Corbett, J. Boyle, Chemlistem: chemical named entity recognition using recurrent neural networks, Journal of Cheminformatics 10 (2018) 59. URL: https://doi.org/10.1186/s13321-018-0313-8. doi:10.1186/s13321-018-0313-8.

[25] Z. Liang, J. Chen, Z. Xu, Y. Chen, T. Hao, A pattern-based method for medical entity recognition from chinese diagnostic imaging text, Frontiers in Artificial Intelligence 2 (2019) 1. URL: https://www.frontiersin.org/article/10.3389/frai.2019.00001. doi:10.3389/frai.2019.00001.

[26] A. Sniegula, A. Poniszewska-Maranda, L. Chomatek, Study of named entity recognition methods in biomedical field, Procedia Computer Science 160 (2019) 260–265. URL: https://www.sciencedirect.com/science/article/pii/S1877050919316813. doi:https://doi.org/10.1016/j.procs.2019.09.466, the 10th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN-2019) / The 9th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare (ICTH-2019) / Affiliated Workshops.

[27] K. r. Kanakarajan, B. Kundumani, M. Sankarasubbu, BioELECTRA:pretrained biomedical text encoder using discriminators, in: Proceedings of the 20th Workshop on Biomedical Language Processing, Association for Computational Linguistics, Online, 2021, pp. 143–154. URL: https://aclanthology.org/2021.bionlp-1.16. doi:10.18653/v1/2021.bionlp-1.16.

[28] L. Weston, V. Tshitoyan, J. Dagdelen, O. Kononova, A. Trewartha, K. Persson, G. Ceder, A. Jain, Named entity recognition and normalization applied to large-scale information extraction from the materials science literature, J. Chem. Inf. Model. 59 (2019) 3692–3702. doi:10.1021/acs.jcim.9b00470.

[29] T. He, W. Sun, H. Huo, O. Kononova, Z. Rong, V. Tshitoyan, T. Botari, G. Ceder, Similarity of precursors in solid-state synthesis as text-mined from scientific literature, Chemistry of Materials 32 (2020) 7861–7873. URL: https://doi.org/10.1021/acs.chemmater.0c02553. doi:10.1021/acs.chemmater.0c02553. arXiv:https://doi.org/10.1021/acs.chemmater.0c02553.

[30] K. Hatakeyama-Sato, K. Oyaizu, Integrating multiple materials science projects in a single neural network, Communications Materials 1 (2020) 49. URL: https://doi.org/10.1038/s43246-020-00052-8. doi:10.1038/s43246-020-00052-8.

[31] T. Dieb, M. Yoshioka, S. Hara, M. Newton, Framework for automatic information extraction from research papers on nanocrystal devices, Beilstein J. Nanotechnol. 6 (2015) 1872–1882. doi:10.3762/bjnano.6.190.

[32] M. Gaultois, T. Sparks, C. Borg, R. Seshadri, W. Bonificio, D. Clarke, Data-driven review of thermoelectric materials: Performance and resource considerations, Chem. Mater. 25 (2013) 2911–2920. doi:10.1021/cm400893e.

[33] N. Pang, L. Qian, W. Lyu, J.-D. Yang, Transfer learning for scientific data chain extraction in small chemical corpus with bert-crf model, 2019. arXiv:1905.05615.

[34] P. Corbett, A. Copestake, Cascaded classifiers for confidence-based chemical named entity recognition, BMC Bioinformatics 9 (2008) S4. doi:10.1186/1471-2105-9-S11-S4.

[35] M. Krallinger, O. Rabal, A. Lourenço, J. Oyarzabal, A. Valencia, Information retrieval and text mining technologies for chemistry, Chem. Rev. 117 (2017) 7673–7761. doi:10.1021/acs.chemrev.6b00851.

[36] S. Eltyeb, N. Salim, Chemical named entities recognition: A review on approaches and applications, J. Cheminform. 6 (2014) 1–12. doi:10.1186/1758-2946-6-17.

[37] T. Rocktäschel, M. Weidlich, U. Leser, Chemspot: A hybrid system for chemical named entity recognition, Bioinformatics 28 (2012) 1633–1640. doi:10.1093/bioinformatics/bts183.

[38] R. Leaman, C.-H. Wei, Z. Lu, tmchem: a high performance approach for chemical named entity recognition and normalization, J.

Cheminform. 7 (2015) S3. doi:10.1186/1758-2946-7-S1-S3.
[39] I. Korvigo, M. Holmatov, A. Zaikovskii, M. Skoblov, Putting hands to rest: efficient deep cnn-rnn architecture for chemical named entity recognition with no hand-crafted rules, J. Cheminform. 10 (2018) 28. doi:10.1186/s13321-018-0280-0.
[40] M. García-Remesal, A. García-Ruiz, D. Pérez-Rey, D. De La Iglesia, V. Maojo, Using nanoinformatics methods for automatically identifying relevant nanotoxicology entities from the literature, Biomed. Res. Int. 2013 (2013). doi:10.1155/2013/410294.
[41] O. Kononova, T. He, H. Huo, A. Trewartha, E. A. Olivetti, G. Ceder, Opportunities and challenges of text mining in materials research, iScience 24 (2021) 102155.
[42] C. C. Fischer, K. J. Tibbetts, D. Morgan, G. Ceder, Predicting crystal structure by merging data mining with quantum mechanics, Nat. Mater. 5 (2006) 641–646. doi:10.1038/nmat1691.
[43] S. R. Young, A. Maksov, M. Ziatdinov, Y. Cao, M. Burch, J. Balachandran, L. Li, S. Somnath, R. M. Patton, S. V. Kalinin, et al., Data mining for better material synthesis: The case of pulsed laser deposition of complex oxides, J. Appl. Phys. 123 (2018) 115303. doi:10.1063/1.5009942.
[44] B. Alperin, A. Kuzmin, L. Ilina, V. Gusev, N. Salomatina, V. Parmon, Terminology spectrum analysis of natural-language chemical documents: term-like phrases retrieval routine, J. Cheminform. 8 (2016) 22. doi:10.1186/s13321-016-0136-4.
[45] C. Court, J. M. Cole, Auto-generated materials database of curie and néel temperatures via semi-supervised relationship extraction, Sci. Data 5 (2018) 180111. doi:10.1038/sdata.2018.111.
[46] C. Court, J. Cole, Magnetic and superconducting phase diagrams and transition temperatures predicted using text mining and machine learning, npj Comput. Mater 6 (2020) 1–9. doi:10.1038/s41524-020-0287-8.
[47] D. M. Jessop, S. E. Adams, E. L. Willighagen, L. Hawizy, P. Murray-Rust, Oscar4: a flexible architecture for chemical text-mining, J. Cheminform. 3 (2011) 41. doi:10.1186/1758-2946-3-41.
[48] L. Hawizy, D. M. Jessop, N. Adams, P. Murray-Rust, Chemicaltagger: A tool for semantic text-mining in chemistry, J. Cheminform. 3 (2011) 1–13. doi:10.1186/1758-2946-3-17.
[49] C. Kolářik, R. Klinger, C. M. Friedrich, M. Hofmann-Apitius, J. Fluck, Chemical names: Terminological resources and corpora annotation, in: Workshop on Building and evaluating resources for biomedical text mining, 2008, pp. 51–58.
[50] S. Mysore, Z. Jensen, E. Kim, K. Huang, H.-S. Chang, E. Strubell, J. Flanigan, A. McCallum, E. Olivetti, The materials science procedural text corpus: Annotating materials synthesis procedures with shallow semantic structures, LAW 2019 - 13th Linguistic Annotation Workshop, Proceedings of the Workshop (2019) 56–64. arXiv:1905.06939.
[51] F. Kuniyoshi, K. Makino, J. Ozawa, M. Miwa, Annotating and extracting synthesis process of all-solid-state batteries from scientific literature, 2020. arXiv:2002.07339.
[52] Z. Jensen, E. Kim, S. Kwon, T. Gani, Y. Roman-Leshkov, M. Moliner, A. Corma, E. Olivetti, A machine learning approach to zeolite synthesis enabled by automatic literature data extraction, ACS Cent. Sci. 5 (2019) 892–899. doi:10.1021/acscentsci.9b00193.
[53] E. Kim, K. Huang, A. Saunders, A. McCallum, G. Ceder, E. Olivetti, Materials synthesis insights from scientific literature via text extraction and machine learning, Chem. Mater. 29 (2017) 9436–9444. doi:10.1021/acs.chemmater.7b03500.
[54] E. Kim, Z. Jensen, A. van Grootel, K. Huang, M. Staib, S. Mysore, H. S. Chang, E. Strubell, A. McCallum, S. Jegelka, E. Olivetti, Inorganic materials synthesis planning with literature-trained neural networks, J. Chem. Inf. Model. 60 (2020) 1194–1201. doi:10.1021/acs.jcim.9b00995.
[55] S. Mysore, E. Kim, E. Strubell, A. Liu, H.-S. Chang, S. Kompella, K. Huang, A. McCallum, E. Olivetti, Automatically extracting action graphs from materials science synthesis procedures (2017). arXiv:1711.06872.
[56] A. Vaucher, F. Zipoli, J. Geluykens, V. Nair, P. Schwaller, T. Laino, Automated extraction of chemical synthesis actions from experimental procedures, Nat. Commun. 11 (2020) 3601. doi:10.1038/s41467-020-17266-6.
[57] I. Tehseen, G. Tahir, K. Shakeel, M. Ali, Corpus based machine translation for scientific text, in: L. Iliadis, I. Maglogiannis, V. Plagianakos (Eds.), Artificial Intelligence Applications and Innovations, Springer International Publishing, Cham, 2018, pp. 196–206. doi:10.1007/978-3-319-92007-8_17.
[58] A. Hiszpanski, B. Gallagher, K. Chellappan, P. Li, S. Liu, H. Kim, B. Kailkhura, J. Han, D. Buttler, T.-J. Han, Nanomaterials synthesis insights from machine learning of scientific articles by extracting, structuring, and visualizing knowledge, J. Chem. Inf. Model. 60 (2020) 2876–2887. doi:10.1021/acs.jcim.0c00199.
[59] J.-D. Kim, T. Ohta, Y. Tateisi, J. Tsujii, Genia corpus – a semantically annotated corpus for bio-textmining, Bioinformatics 19 (2003) i180–i182. doi:10.1093/bioinformatics/btg1023.
[60] N. Milosevic, C. Gregson, R. Hernandez, G. Nenadic, A framework for information extraction from tables in biomedical literature, IJDAR 22 (2019) 55–78. doi:10.1007/s10032-019-00317-0.
[61] H. Huo, Z. Rong, O. Kononova, W. Sun, T. Botari, T. He, V. Tshitoyan, G. Ceder, Semi-supervised machine-learning classification of materials synthesis procedures, npj Comput. Mater 5 (2019) 1–7. doi:10.1038/s41524-019-0204-1.
[62] A. M. Hiszpanski, B. Gallagher, K. Chellappan, P. Li, S. Liu, H. Kim, J. Han, B. Kailkhura, D. J. Buttler, T. Y.-J. Han, Nanomaterial synthesis insights from machine learning of scientific articles by extracting, structuring, and visualizing knowledge, Journal of Chemical Information and Modeling 60 (2020) 2876–2887. URL: https://doi.org/10.1021/acs.jcim.0c00199. doi:10.1021/acs.jcim.0c00199.
[63] L. Rasmy, Y. Xiang, Z. Xie, C. Tao, D. Zhi, Med-BERT: pre-trained contextualized embeddings on large-scale structured electronic health records for disease prediction, 2020. arXiv:2005.12833.
[64] A. Friedrich, H. Adel, F. Tomazic, J. Hingerl, R. Benteau, A. Marusczyk, L. Lange, The SOFC-exp corpus and neural approaches to information extraction in the materials science domain, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, 2020, pp. 1255–1268. doi:10.18653/v1/2020.acl-main.116.
[65] Solid state abstract annotations, 2019. URL: https://figshare.com/articles/dataset/Materials_Science_Named_Entity_Recognition_train_development_test_sets/8184428.
[66] Doping and AuNP NER DOIs and Entities, 2022. URL: https://figshare.com/articles/dataset/NER_Datasets_DOIs_and_Entities_Doping_and_AuNP_/16864357.
[67] Elsevier scopus, 2022. URL: https://dev.elsevier.com/.

[68] Springer-nature, 2022. URL: https://dev.springernature.com/.
[69] Royal society of chemistry, 2022. URL: https://rsc.org/.
[70] Electrochemical society, 2022. URL: https://electrochem.org/.
[71] M. Baek, S. Park, D. Choi, Synthesis of zirconia (zro2) nanowires via chemical vapor deposition, Journal of Crystal Growth 459 (2017) 198–202. URL: https://www.sciencedirect.com/science/article/pii/S0022024816308922. doi:https://doi.org/10.1016/j.jcrysgro.2016.12.033.
[72] Matscholar, 2022. URL: https://matscholar.com/.
[73] T.-P. Tang, M.-R. Yang, K.-S. Chen, Photoluminescence of zns: Sm phosphor prepared in a reductive atmosphere, Ceramics International 26 (2000) 153–158. URL: https://www.sciencedirect.com/science/article/pii/S0272884299000346. doi:https://doi.org/10.1016/S0272-8842(99)00034-6.
[74] L. A. Dykman, N. G. Khlebtsov, Gold nanoparticles in biology and medicine: recent advances and prospects, Acta naturae 3 (2011) 34–55. URL: https://pubmed.ncbi.nlm.nih.gov/22649683, 22649683[pmid].
[75] X. Huang, M. A. El-Sayed, Gold nanoparticles: Optical properties and implementations in cancer diagnosis and photothermal therapy, Journal of Advanced Research 1 (2010) 13–28. URL: https://www.sciencedirect.com/science/article/pii/S2090123210000056. doi:https://doi.org/10.1016/j.jare.2010.02.002.
[76] K. Sandeep, B. Manoj, K. G. Thomas, Gold nanoparticle on semiconductor quantum dot: Do surface ligands influence fermi level equilibration, The Journal of Chemical Physics 152 (2020) 044710. URL: https://doi.org/10.1063/1.5138216. doi:10.1063/1.5138216. arXiv:https://doi.org/10.1063/1.5138216.
[77] M. Lau, A. Ziefuss, T. Komossa, S. Barcikowski, Inclusion of supported gold nanoparticles into their semiconductor support, Phys. Chem. Chem. Phys. 17 (2015) 29311–29318. URL: http://dx.doi.org/10.1039/C5CP04296H. doi:10.1039/C5CP04296H.
[78] S. Kaul, N. Gulati, D. Verma, S. Mukherjee, U. Nagaich, Role of nanotechnology in cosmeceuticals: A review of recent advances, Journal of pharmaceutics 2018 (2018) 3420204–3420204. URL: https://pubmed.ncbi.nlm.nih.gov/29785318. doi:10.1155/2018/3420204, 29785318[pmid].
[79] Y. C. Dong, M. Hajfathalian, P. S. N. Maidment, J. C. Hsu, P. C. Naha, S. Si-Mohamed, M. Breuilly, J. Kim, P. Chhour, P. Douek, H. I. Litt, D. P. Cormode, Effect of gold nanoparticle size on their properties as contrast agents for computed tomography, Scientific Reports 9 (2019) 14912. URL: https://doi.org/10.1038/s41598-019-50332-8. doi:10.1038/s41598-019-50332-8.
[80] S. A. Ng, K. A. Razak, A. A. Aziz, K. Y. Cheong, The effect of size and shape of gold nanoparticles on thin film properties, Journal of Experimental Nanoscience 9 (2014) 64–77. URL: https://doi.org/10.1080/17458080.2013.813651. doi:10.1080/17458080.2013.813651. arXiv:https://doi.org/10.1080/17458080.2013.813651.
[81] R. Kaur, B. Pal, Physicochemical and catalytic properties of au nanorods micro-assembled in solvents of varying dipole moment and refractive index, Materials Research Bulletin 62 (2015) 11–18. URL: https://www.sciencedirect.com/science/article/pii/S002554081400693X. doi:https://doi.org/10.1016/j.materresbull.2014.11.012.
[82] M. C. Swain, J. M. Cole, Chemdataextractor: A toolkit for automated extraction of chemical information from the scientific literature, Journal of Chemical Information and Modeling 56 (2016) 1894–1904. URL: https://doi.org/10.1021/acs.jcim.6b00207. doi:10.1021/acs.jcim.6b00207.
[83] M. Schuster, K. Nakajima, Japanese and korean voice search, in: 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2012, pp. 5149–5152. doi:10.1109/ICASSP.2012.6289079.
[84] R. Sennrich, B. Haddow, A. Birch, Neural machine translation of rare words with subword units, 2016. arXiv:1508.07909.
[85] V. Krishnan, V. Ganapathy, Named entity recognition, 2005.
[86] N. Alshammari, S. Alanazi, The impact of using different annotation schemes on named entity recognition, Egyptian Informatics Journal (2020). URL: https://www.sciencedirect.com/science/article/pii/S1110866520301596. doi:https://doi.org/10.1016/j.eij.2020.10.004.
[87] J. Lafferty, A. McCallum, F. Pereira, Conditional random fields: Probabilistic models for segmenting and labeling sequence data, in: Proceedings of the Eighteenth International Conference on Machine Learning, ICML '01, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2001, p. 282–289.
[88] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, C. Dyer, Neural architectures for named entity recognition, 2016. arXiv:1603.01360.
[89] W. Huang, X. Cheng, T. Wang, W. Chu, Bert-based multi-head selection for joint entity-relation extraction, 2019. arXiv:1908.05908.
[90] F. Souza, R. Nogueira, R. Lotufo, Portuguese named entity recognition using bert-crf, 2020. arXiv:1909.10649.
[91] V. Tshitoyan, J. Dagdelen, L. Weston, A. Dunn, Z. Rong, O. Kononova, K. A. Persson, G. Ceder, A. Jain, Unsupervised word embeddings capture latent knowledge from materials science literature, Nature 571 (2019) 95–98. doi:10.1038/s41586-019-1335-8.
[92] R. Jozefowicz, O. Vinyals, M. Schuster, N. Shazeer, Y. Wu, Exploring the limits of language modeling, 2016. arXiv:1602.02410.
[93] M. Grankin, over9000, https://github.com/mgrankin/over9000, 2019.
[94] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, J. Han, On the variance of the adaptive learning rate and beyond, 2020. arXiv:1908.03265.
[95] M. R. Zhang, J. Lucas, G. Hinton, J. Ba, Lookahead optimizer: k steps forward, 1 step back, 2019. arXiv:1907.08610.
[96] L. Wright, New deep learning optimizer, ranger: Synergistic combination of radam lookahead for the best of both., 2019. URL: https://lessw.medium.com/new-deep-learning-optimizer-ranger-synergistic-combination-of-radam-lookahead-for-the-best-of-2dc83f79a48d.
[97] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, Least angle regression, The Annals of Statistics 32 (2004) 407 – 499. URL: https://doi.org/10.1214/009053604000000067. doi:10.1214/009053604000000067.
[98] MatBERT, 2021. URL: https://github.com/lbnlp/MatBERT.
[99] MatBERT weights, 2022. URL: https://figshare.com/articles/software/MatBERT-NER_models/15087276.
[100] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, A. M. Rush, Transformers: State-of-the-art natural language processing, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System

Demonstrations, Association for Computational Linguistics, Online, 2020, pp. 38–45. URL: https://www.aclweb.org/anthology/2020.emnlp-demos.6.
[101] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, Pytorch: An imperative style, high-performance deep learning library, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems 32, Curran Associates, Inc., 2019, pp. 8024–8035. URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.
[102] Y. You, J. Li, S. Reddi, J. Hseu, S. Kumar, S. Bhojanapalli, X. Song, J. Demmel, K. Keutzer, C.-J. Hsieh, Large batch optimization for deep learning: Training bert in 76 minutes, 2020. arXiv:1904.00962.
[103] MatBERT NER, 2022. URL: https://zenodo.org/badge/latestdoi/315418846.
[104] E. F. T. K. Sang, F. D. Meulder, Introduction to the conll-2003 shared task: Language-independent named entity recognition, CoRR cs.CL/0306050 (2003). URL: http://arxiv.org/abs/cs/0306050.
[105] J. Towns, T. Cockerill, M. Dahan, I. Foster, K. Gaither, A. Grimshaw, V. Hazlewood, S. Lathrop, D. Lifka, G. D. Peterson, R. Roskies, J. R. Scott, N. Wilkins-Diehr, Xsede: Accelerating scientific discovery, Computing in Science & Engineering 16 (2014) 62–74. URL: doi.ieeecomputersociety.org/10.1109/MCSE.2014.80. doi:10.1109/MCSE.2014.80.

9. Figure Titles

1. Solid state annotation example
2. Doping annotation example
3. Gold nanoparticle annotation example
4. NER model precisions, recalls, and F1-scores
5. NER Entity score heatmap
6. NER learning curves
7. NER Entity scores as a function of support
8. MatBERT keywords

10. Tables

Table 1: BiLSTM parameters: A table of parameters for the BiLSTM model.

Embeddings			LSTM	
	Word	Character	Layers	2
Dimension	200	38	Hidden Dimension	64
Dropout	0.5	0.5	Dropout	0.1
Convolutions			Multi-head Attention	
Filters	4		Heads	16
Kernel Size	(3,)		Dropout	0.25
Kernel Stride	(1,)			
Dropout	0.25			

Table 2: $\text{BERT}_{\text{BASE}}$ parameters: A table of parameters for the $\text{BERT}_{\text{BASE}}$ model.

Hidden Layers	12	Embeddings	
Attention Heads	12	Hidden Dimension	768
Dropout	0.1	Intermediate Dimension	3072
Activation Function	GELU	Positions	512
Layer Normalization	$\epsilon = 1 \cdot 10^{-12}$	Token Types	2

11. Table Titles

1. BiLSTM parameters
2. $\text{BERT}_{\text{BASE}}$ parameters