npj computational materials

www.nature.com/npjcompumats

ARTICLE OPEN

Check for updates

MatSciBERT: A materials domain language model for text mining and information extraction

Tanishq Gupta[1], Mohd Zaki[2], N. M. Anoop Krishnan[2,3]✉ and Mausam[3,4]✉

A large amount of materials science knowledge is generated and stored as text published in peer-reviewed scientific literature. While recent developments in natural language processing, such as Bidirectional Encoder Representations from Transformers (BERT) models, provide promising information extraction tools, these models may yield suboptimal results when applied on materials domain since they are not trained in materials science specific notations and jargons. Here, we present a materials-aware language model, namely, MatSciBERT, trained on a large corpus of peer-reviewed materials science publications. We show that MatSciBERT outperforms SciBERT, a language model trained on science corpus, and establish state-of-the-art results on three downstream tasks, named entity recognition, relation classification, and abstract classification. We make the pre-trained weights of MatSciBERT publicly accessible for accelerated materials discovery and information extraction from materials science texts.

npj Computational Materials (2022)8:102 ; https://doi.org/10.1038/s41524-022-00784-w

INTRODUCTION

Discovering materials and utilizing them for practical applications is an extremely time-consuming process that may span decades[1,2]. To accelerate this process, we need to exploit and harness the knowledge on materials that has been developed over the centuries through rigorous scientific procedure in a cohesive fashion[3–8]. Textbooks, scientific publications, reports, handbooks, websites, etc., serve as a large data repository that can be mined for obtaining the already existing information[9,10]. However, it is a challenging task to extract useful information from these texts since most of the scientific data is semi- or un-structured in the form of text, paragraphs with cross reference, image captions, and tables[10–12]. Extracting such information manually is extremely time- and resource-intensive and relies on the interpretation of a domain expert.

Natural language processing (NLP), a sub-domain in artificial intelligence, presents an alternate approach that can automate information extraction from text. Earlier approaches in NLP relied on non-neural methods based on *n*-grams such as Brown et al. (1992)[13], structural learning framework by Ando and Zhang (2005)[14], or structural correspondence learning by Blitzer et al. (2006)[15], but these are no longer state of the art. Neural pre-trained embeddings like word2vec[16,17] and GloVe[18] are quite popular, but they lack domain-specific knowledge and do not produce contextual embeddings. Recent progress in NLP has led to the development of a computational paradigm in which a large, pre-trained language model (LM) is finetuned for domain-specific tasks. Research has consistently shown that this pretrain-finetune paradigm leads to the best overall task performance[19–23]. Statistically, LMs are probability distributions for a sequence of words such that for a given set of words, it assigns a probability to each word[24]. Recently, due to the availability of large amounts of text and high computing power, researchers have been able to pre-train these large neural language models. For example, Bidirectional Encoder Representations from Transformers (BERT)[25] is trained on BookCorpus[26] and English Wikipedia, resulting in state-of-the-art performance on multiple NLP tasks like question answering and entity recognition, to name a few.

Researchers have used NLP tools to automate database creation for ML applications in the materials science domain. For instance, ChemDataExtractor[27], an NLP pipeline, has been used to create databases of battery materials[28], Curie and Néel temperatures of magnetic materials[29], and inorganic material synthesis routes[30]. Similarly, NLP has been used to collect the composition and dissolution rate of calcium aluminosilicate glassy materials[31], and zeolite synthesis routes to synthesize germanium containing zeolites[32], and to extract process and testing parameters of oxide glasses, thereby enabling improved prediction of the Vickers hardness[11]. Researchers have also made an automated NLP tool to create databases using the information extracted from computational materials science research papers[33]. NLP has also been used for other tasks such as topic modeling in glasses, that is, to group the literature into different topics in an unsupervised fashion and to find images based on specific queries such as elements present, synthesis, or characterization techniques, and applications[10].

A comprehensive review by Olivetti et al. (2019) describes several ways in which NLP can benefit the materials science community[34]. Providing insights into chemical parsing tools like OSCAR4[35] capable of identifying entities and chemicals from text, Artificial Chemist[36], which takes the input of precursor information and generates synthetic routes to manufacture optoelectronic semiconductors with targeted band gaps, robotic system for making thin films to produce cleaner and sustainable energy solutions[37], and identification of more than 80 million materials science domain-specific named entities, researches have prompted the accelerated discovery of materials for different applications through the combination of ML and NLP techniques. Researchers have shown the domain adaptation capability of word2vec and BERT in the field of biological sciences as BioWordVec[38] and BioBERT[19], other domain-specific BERTs like SciBERT[21] trained on scientific and biomedical corpus[39], clinical-BERT[40] trained on 2 million clinical notes in MIMIC-III v1.4

[1]Department of Mathematics, Indian Institute of Technology Delhi, Hauz Khas, New Delhi 110016, India. [2]Department of Civil Engineering, Indian Institute of Technology Delhi, Hauz Khas, New Delhi 110016, India. [3]School of Artificial Intelligence, Indian Institute of Technology Delhi, Hauz Khas, New Delhi 110016, India. [4]Department of Computer Science and Engineering, Indian Institute of Technology Delhi, Hauz Khas, New Delhi 110016, India. ✉email: krishnan@iitd.ac.in; mausam@iitd.ac.in

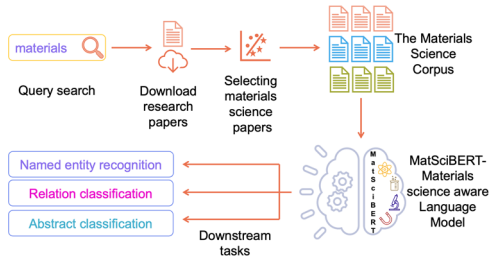

Fig. 1 Methodology for training MatSciBERT. We create the Materials Science Corpus (MSC) through query search followed by selection of relevant research papers. MatSciBERT, pre-trained on MSC, is evaluated on various downstream tasks.

database[41], mBERT[42] for multilingual machine translations tasks, PatentBERT[23] for patent classification and FinBERT for financial tasks[22]. This suggests that a materials-aware LM can significantly accelerate the research in the field by further adapting to downstream tasks[9,34]. Although there were no papers on developing materials-aware language models prior to this work[43], in a recent preprint[44], Walker et al. (2021) emphasize the impact of domain-specific language models on named entity recognition (NER) tasks in materials science.

In this work, we train materials science domain-specific BERT, namely MatSciBERT. Figure 1 shows the graphical summary of the methodology adopted in this work encompassing creating the materials science corpus, training the MatSciBERT, and evaluating different downstream tasks. We achieve state-of-the-art results on domain-specific tasks as listed below.

a. NER on SOFC, SOFC Slot dataset by Friedrich et al. (2020)[45] and Matscholar dataset by Weston et al. (2019)[9]
b. Glass vs. Non-Glass classification of paper abstracts[10]
c. Relation Classification on MSPT corpus[46]

The present work, thus, bridges the gap in the availability of a materials domain language model, allowing researchers to automate information extraction, knowledge graph completion, and other downstream tasks and hence accelerate the discovery of materials. We have hosted the MatSciBERT pre-trained weights at https://huggingface.co/m3rg-iitd/matscibert and codes for pre-training and finetuning on downstream tasks at https://github.com/M3RG-IITD/MatSciBERT. Also, the codes with finetuned models for the downstream tasks are available at https://doi.org/10.5281/zenodo.6413296.

RESULTS AND DISCUSSION

Dataset

Textual datasets are an integral part of the training of an LM. There exist many general-purpose corpora like BookCorpus[26] and EnglishWikipedia, and domain-specific corpora like biomedical corpus[39], and clinical database[41], to name a few. However, none of these corpora is suitable for the materials domain. Therefore, with the aim of providing a materials specific LM, we first create a corpus spanning four important materials science families of inorganic glasses, metallic glasses, alloys, and cement and concrete. It should be noted that although these broad categories are mentioned, several other categories of materials, including two-dimensional materials, were also present in the corpus. Specifically, we have selected ~150 K papers out of ~1 M papers downloaded from the Elsevier Science Direct Database. The steps to create the corpus are provided in the Methods section. The details about the number of papers and words for each family are given in Supplementary Table 1. We have also provided the list of DOIs and PIIs of the papers used to pre-train MatSciBERT in the GitHub repository for this work.

The materials science corpus developed for this work has ~285 M words, which is nearly 9% of the number of words used to pre-train SciBERT (3.17B words) and BERT (3.3B words). Since we continue pre-training SciBERT, MatSciBERT is effectively trained on a corpus consisting of $3.17 + 0.28 = 3.45$B words. From Supplementary Table 1, one can observe that 40% of the words are from research papers related to inorganic glasses and ceramics, and 20% each from bulk metallic glasses (BMG), alloys, and cement. Although the number of research papers for "cement and concrete" is more than "inorganic glasses and ceramics", the latter has higher words. This is because of the presence of a greater number of full-text documents retrieved associated with the latter category. The Supplementary Table 2 represents the word count of important strings relevant to the field of materials science. It should be noted that the corpus encompasses the important fields of thermoelectric, nanomaterials, polymers, and biomaterials. Also, note that the corpora used for training the language model consists of both experimental and computational works as both these approaches play a crucial role in understanding material response. The average paper length for this corpus is ~1848 words, which is two-thirds of the average paper length of 2769 words for the SciBERT corpus. The lower average paper length can be attributed to two things: (a) In general, materials science papers are shorter than biomedical papers. We verified this by computing the average paper length of full-text materials science papers. The number came out to be 2366. (b) There are papers without full text also in our corpus. In that case, we have used the abstracts of such papers to arrive at the final corpus.

Pre-training of MatSciBERT

For MatSciBERT pre-training, we follow the domain adaptive pre-training proposed by Gururangan et al. (2020). In this work, authors continued pre-training of the initial LM on corpus of domain-specific text[20]. They observed a significant improvement in the performance on domain-specific downstream tasks for all the four domains despite the overlap between initial LM vocabulary and domain-specific vocabulary being less than 54.1%. BioBERT[19] and FinBERT[22] were also developed using the similar approach where the vanilla BERT model was further pre-trained on domain-specific text, and tokenization is done using the original BERT vocabulary. We initialize MatSciBERT weights with that of some suitable LM and then pre-train it on MSC. To determine the appropriate initial weights for MatSciBERT, we trained an uncased wordpiece[47] vocabulary based on the MSC using the tokenizers library[48]. The overlap of MSC vocabulary is 53.64% with the uncased SciBERT[21] vocabulary and 38.90% with the uncased BERT vocabulary. Because of the larger overlap with the vocabulary of SciBERT, we tokenize our corpus using the SciBERT vocabulary and initialize the MatSciBERT weights with that of SciBERT as made publicly available by Beltagy et al. (2019)[21]. It is worth mentioning that a materials science domain-specific vocabulary would likely represent the corpus with a lesser number of wordpieces and potentially lead to a better language model. For e.g., "yttria-stabilized zirconia" is tokenized as ["yt", "##tri", "##a", "-", "stabilized", "zircon", "##ia"] by the SciBERT vocabulary, whereas a domain-specific tokenization might have resulted in ["yttria", "-", "stabilized", "zirconia"]. However, using a domain-specific tokenizer does not allow the use of SciBERT weights and takes advantage of the scientific knowledge already learned by SciBERT. Further, using the SciBERT vocabulary for the materials domain is not necessarily detrimental since the deep neural language models have the capacity to learn repeating patterns that represent new words using the existing tokenizer. For instance, when the wordpieces "yt", "##tri", and "##a" occur consecutively, SciBERT indeed recognizes that some material is being discussed, as demonstrated in the downstream tasks.

Table 1. Macro-F1 scores on the test set for SOFC-Slot and SOFC datasets averaged over three seeds and five cross-validation splits.

Architecture	LM = MatSciBERT	LM = SciBERT	LM = BERT	SOTA
SOFC-Slot dataset				
LM-Linear	63.82 ± 2.53 (67.53 ± 4.23)	58.64 ± 1.49 (64.58 ± 3.73)	57.06 ± 2.86 (61.68 ± 5.23)	62.6 (67.8 ± 12.9)
LM-CRF	65.35 ± 2.73 (70.07 ± 3.36)	59.07 ± 2.85 (68.31 ± 2.88)	58.26 ± 1.73 (65.38 ± 3.96)	
LM-BiLSTM-CRF	65.95 ± 2.69 (69.76 ± 3.72)	61.68 ± 1.42 (68.44 ± 3.15)	55.44 ± 1.97 (65.36 ± 3.68)	
SOFC dataset				
LM-Linear	82.28 ± 1.11 (81.60 ± 2.63)	79.91 ± 1.20 (80.91 ± 2.37)	77.08 ± 1.75 (79.61 ± 3.01)	81.5 (81.7 ± 4.2)
LM-CRF	82.39 ± 1.23 (82.61 ± 2.34)	81.07 ± 0.93 (82.04 ± 2.36)	78.93 ± 1.62 (81.26 ± 2.87)	
LM-BiLSTM-CRF	82.24 ± 1.12 (82.61 ± 2.77)	80.12 ± 1.00 (81.92 ± 2.27)	78.15 ± 1.55 (80.94 ± 2.72)	

Values in the parenthesis show the results on the validation set.

This is also why most domain-specific BERT-based LMs like FinBERT[22], BioBERT[19], and ClinicalBERT[40] extend the pre-training instead of using domain-specific tokenizers and learning from scratch.

The details of the pre-training procedure are provided in the Methods section. The pre-training was performed for 360 h, after which the model achieved a final perplexity of 2.998 on the validation set (see Supplementary Fig. 1a). Although not directly comparable due to different vocabulary and validation corpus, BERT[25], and RoBERTa[49] authors report perplexities as 3.99 and 3.68, respectively, which are in the same range. We also provide graphs for other evaluation metrics like MLM loss and MLM accuracy in Supplementary Fig. 1b, c. The final pre-trained LM was then used to evaluate different materials science domain-specific downstream tasks, details of which are described in the subsequent sections. The performance of the LM on the downstream tasks was compared with that of SciBERT, BERT, and other baseline models to evaluate the effectiveness of MatSciBERT to learn the materials' specific information.

In order to understand the effect of pre-training on the model performance, a materials domain-specific downstream task, NER on SOFC-slot, was performed using the model at regular intervals of pre-training. To this extent, the pre-trained model was finetuned on the training set of the SOFC-slot dataset. The choice of the SOFC-slot dataset was based on the fact that the dataset was comprised of fine-grained materials-specific information. Thus, this dataset is appropriate to distinguish the performance of SciBERT from the materials-aware LMs. The performance of these finetuned models was evaluated on the test set. LM-CRF architecture was used for the analysis since LM-CRF consistently gives the best performance for the downstream task, as shown later in this work. The macro-F1 averages across three seeds exhibited an increasing trend (see Supplementary Fig. 2a), suggesting the importance of training for longer durations. We also show a similar graph for the abstract classification task (Supplementary Fig. 2b).

Downstream tasks

Here, we evaluate MatSciBERT on three materials science specific downstream tasks namely, Named Entity Recognition (NER), Relation Classification, and Paper Abstract Classification.

We now present the results on the three materials science NER datasets as described in the Methods section. To the best of our knowledge, the best Macro-F1 on solid oxide fuel cells (SOFC) and SOFC-Slot datasets is 81.50% and 62.60%, respectively, as reported by Friedrich et al. (2020), who introduced the dataset[44]. We run the experiments on the same train-validation-test splits as done by Friedrich et al. (2020) for a fair comparison of results. Moreover, since the authors reported results averaged over 17 entities (the extra entity is "Thickness") for the SOFC-Slot dataset, we also report the results taking the 'Thickness' entity into account.

Table 1 shows the Macro-F1 scores for the NER task on the SOFC-Slot and SOFC datasets by MatSciBERT, SciBERT, and BERT. We observe that LM-CRF always performs better than LM-Linear. This can be attributed to the fact that the CRF layer can model the BIO tags accurately. Also, all SciBERT architectures perform better than the corresponding BERT architecture. We obtained an improvement of ~6.3 Macro F1 and ~3.2 Micro F1 (see Supplementary Table 3) on the SOFC-Slot test set for MatSciBERT vs. SciBERT while using the LM-CRF architecture. For the SOFC test dataset, MatSciBERT-BiLSTM-CRF performs better than SciBERT-BiLSTM-CRF by ~2.1 Macro F1 and ~2.1 Micro F1. Similar improvements can be seen for other architectures as well. These MatSciBERT results also surpass the current best results on SOFC-Slot and SOFC datasets by ~3.35 and ~0.9 Macro-F1, respectively.

It is worth noting that the SOFC-slot dataset consists of 17 entity types and hence has more fine-grained information regarding the materials. On the other hand, SOFC has only four entity types representing coarse-grained information. We notice that the performance of MatSciBERT on SOFC-slot is significantly better than that of SciBERT. To further evaluate this aspect, we analyzed the F1-score of both SciBERT and MatSciBERT on all the 17 entity types of the SOFC-slot data individually, as shown in Fig. 2. Interestingly, we observe that for all the materials related entity types, namely anode material, cathode material, electrolyte material, interlayer material, and support material, MatSciBERT performs better than SciBERT. In addition, for materials related properties such as open circuit voltage and degradation rate, MatSciBERT is able to significantly outperform SciBERT. This suggests that MatSciBERT is indeed able to capitalize on the additional information learned from the MSC to deliver better performance on complex problems specific to the materials domain.

Now, we present the results for the Matscholar dataset[9] in Table 2. For this dataset too, MatSciBERT outperforms SciBERT, BERT as well as the current best results, as can be seen in the case of LM-CRF architecture. The authors obtained Macro-F1 of 85.41% on the validation set and 85.10% on the test set, and Micro-F1 of 87.09% and 87.04% (see Supplementary Table 4). We observe that our

best model MatSciBERT-CRF has Macro-F1 values of 88.66% and 86.38%, both better than the existing state of the art.

In order to demonstrate the performance of MatSciBERT, we demonstrate an example from the validation set of the dataset in Supplementary Figs. 3 and 4. The overall superior performance of MatSciBERT is evident from Table 2.

Table 3 shows the results for the Relation Classification task performed on the Materials Synthesis Procedures dataset[46]. We also compare the results with two recent baseline models, MaxPool and MaxAtt[50], details of which can be found in the Methods section. Even in this task, we observe that MatSciBERT performs better than SciBERT, BERT, and baseline models consistently, although with a lower margin.

In Paper Abstract Classification downstream task, we consider the ability of LMs to classify a manuscript into glass vs. non-glass topics based on an in-house dataset[10]. This is a binary classification problem, with the input being the abstract of a manuscript. Here too, we use the same baseline models MaxPool and MaxAtt[50]. Table 4 shows the comparison of accuracies achieved by MatSciBERT, SciBERT, BERT, and baselines. It can be clearly seen that MatSciBERT outperforms SciBERT by more than 2.75% accuracy on the test set.

Altogether, we demonstrate that the MatSciBERT, pre-trained on a materials science corpus, can perform better than SciBERT for all the downstream tasks such as NER, abstract classification, and relation classification on materials datasets. These results also suggest that the scientific literature in the materials domain, on which MatSciBERT is pre-trained, is significantly different from the computer science and biomedical domains on which SciBERT is trained. Specifically, each scientific discipline exhibits significant variability in terms of ontology, vocabulary, and domain-specific notations. Thus, the development of a domain-specific language model, even within the scientific literature, can significantly enhance the performance in downstream tasks related to text mining and information extraction from literature.

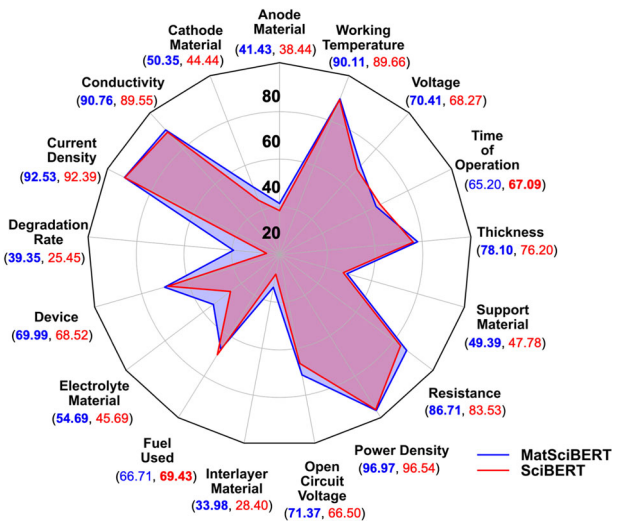

Fig. 2 Comparison of MatSciBERT and SciBERT on validation sets of SOFC-Slot dataset. The entity-level F1-score for MatSciBERT and Scibert models in blue and red color respectively. The bold colored text represents the best model's score.

Applications in materials domain

Now, we discuss some of the potential areas of application of MatSciBERT in materials science. These areas can range from the simple topic-based classification of research papers to discovering materials or alternate applications for existing materials. We demonstrate some of these applications as follows: (i) Document classification: A large number of manuscripts have been published on materials related topics, and the numbers are increasing exponentially. Identifying manuscripts related to a given topic is a challenging task. Traditionally, these tasks are carried out employing approaches such as term frequency-inverse document frequency (TFIDF) or Word2Vec, which is used along with a classification algorithm. However, these approaches directly vectorize a word and are not context sensitive. For instance, in the phrases "flat glass", "glass transition temperature", "tea glass", the word "glass" is used in a very different sense. MatSciBERT will be able to extract the contextual meaning of the embeddings. Thus, MatSciBERT will be able to effectively classify the topics thereby enabling improved topic classification. This is evident from the binary classification results presented earlier in Table 4, where we observe that the accuracy obtained using MatSciBERT (96.22%) was found to be significantly higher than the results obtained using pooling based BiLSTM models (91.44%). This approach can be extended to a larger set of abstracts for the accurate classification of documents from the literature.

(ii) Topic modeling: Topic modeling is an unsupervised approach of grouping documents belonging to similar topics together. Traditionally, topic modeling employs algorithms such as latent Dirichlet allocation (LDA) along with TF-IDF or Word2Vec to cluster documents having the same or semantically similar words together. Note that these approaches rely purely on the frequency of word (in TF-IDF) or the embeddings of the word (in Word2Vec) for clustering without taking into account the context. The use of context-aware embeddings as learned in MatSciBERT could significantly enhance the topic modeling task. As a preliminary study, we perform topic modeling using MatSciBERT on an in-house corpus of abstracts on glasses and ceramics. Note that the same corpus was used in an earlier work[10] for topic modeling using LDA. Specifically, we obtain the output embeddings of the [CLS] token for each abstract using MatSciBERT. Further, these embeddings were projected into two dimensions using the UMAP algorithm[51] and then clustered using the k-means algorithm[52]. We then concatenate all the abstracts belonging to the same cluster and calculate the most frequent words for each cluster/topic.

Table 2. Macro-F1 scores on the test set for Matscholar averaged over three seeds.

Architecture	LM = MatSciBERT	LM = SciBERT	LM = BERT	SOTA
LM-Linear	85.46 ± 0.13 (87.83 ± 1.21)	83.80 ± 0.32 (86.05 ± 0.55)	82.10 ± 0.81 (82.79 ± 0.20)	85.10 (85.41)
LM-CRF	86.38 ± 0.49 (88.66 ± 0.88)	85.04 ± 0.77 (88.07 ± 0.96)	84.07 ± 0.19 (84.61 ± 0.81)	
LM-BiLSTM-CRF	86.09 ± 0.46 (89.15 ± 0.57)	85.66 ± 0.24 (87.66 ± 0.29)	83.39 ± 0.20 (84.07 ± 0.29)	

Values in the parenthesis show the results on the validation set.

Table 3. Test set results for Materials Synthesis Procedures dataset averaged over three seeds.

	MatSciBERT	SciBERT	BERT	MaxPool	MaxAtt
Macro-F1	89.02 ± 0.27(88.31 ± 0.14)	87.22 ± 0.58(87.21 ± 0.17)	85.40 ± 1.45(85.95 ± 0.78)	81.19 ± 1.54(80.93 ± 0.71)	80.39 ± 0.85(81.53 ± 2.23)
Micro-F1	91.94 ± 0.20(91.50 ± 0.20)	91.04 ± 0.32)(91.03 ± 0.08	90.16 ± 0.69(90.44 ± 0.54)	86.81 ± 1.55(86.68 ± 0.84)	87.16 ± 0.60(87.62 ± 1.34)

Values in the parenthesis represent the results on the validation set.

Table 4. Test set results for glass vs. non-glass dataset averaged over three seeds.

	MatSciBERT	SciBERT	BERT	MaxPool	MaxAtt
Accuracy	96.22 ± 0.16 (95.33 ± 0.27)	93.44 ± 0.57 (94.00 ± 0.00)	93.89 ± 0.68 (93.33 ± 0.98)	91.44 ± 0.31 (92.22 ± 0.56)	91.44 ± 0.68 (91.22 ± 0.16)

Values in the parenthesis represent the results on the validation set.

The Supplementary Tables 5 and 6 shows the top ten topics obtained using LDA and MatSciBERT, respectively. The top 10 keywords associated with each topic are also provided in the table. We observe that the topics and keywords from MatSciBERT-based topic modeling are more coherent than the ones obtained from LDA. Further, the actual topics associated with the keywords are not very apparent from Supplementary Table 5. Specifically, Topic 9 by LDA contains keywords from French, suggesting that the topic represents French publications. Similarly, Topic 5 and Topic 3 have several generic keywords that don't represent a topic clearly. On the other hand, the keywords obtained by MatSciBERT enable a domain expert to identify the topics well. For instance, some of the topics identified based on the keywords by three selected domain experts are dissolution of silicates (9), oxide thin films synthesis and their properties (8, 6), materials for energy (0), electrical behavior of ceramics (1), and luminescence studies (5). Despite their efforts, the same three domain experts were unable to identify coherent topics based on the keywords provided by LDA. Altogether, MatSciBERT can be used for topic modeling, thereby providing a broad overview of the topics covered in the literature considered.

(iii) Information extraction from images: Images hold a large amount of information regarding the structure and properties of materials. A proxy to identify relevant images would be to go through the captions of all the images. However, each caption may contain multiple entities, and identifying the relevant keywords might be a challenging task. To this extent, MatSciBERT finetuned on NER can be an extremely useful tool for extracting information from figure captions.

Here, we extracted entities from the figure captions used by Venugopal et al. (2021)[10] using MatSciBERT finetuned on the Matscholar NER dataset. Specifically, entities were extracted from ~110,000 image captions on topics related to inorganic glasses. Using MatSciBERT, we obtained 87,318 entities as DSC (sample descriptor), 10,633 entities under APL (application), 145,324 as MAT (inorganic material), 76,898 as PRO (material property), 73,241 as CMT (characterization method), 33,426 as SMT (synthesis method), and 2,676 as SPL (symmetry/phase label). Figure 3 shows the top 10 extracted entities under the seven categories proposed in the Matscholar dataset. The top entities associated with each of the categories are coating (application), XRD (characterization), glass (sample descriptor, inorganic material), composition (material property), heat (synthesis method), and hexagonal (symmetry/phase). Further details associated with each category can also be obtained from these named entities. It should be noted that each caption may be associated with more than one entity. These entities can then be used to obtain relevant images for specific queries such as "XRD measurements of glasses used for coating" or "emission spectra of doped glasses", or "SEM images of bioglasses with Ag", to name a few.

Further, Fig. 4 shows some of the selected captions from the image captions along with the corresponding manual annotation by Venugopal et al. (2021)[10]. The task of assigning tags to each caption was carried out by human experts. Note that only one word was assigned per image caption in the previous work. Using the MatSciBERT NER model, we show that multiple entities are extracted for the selected five captions. This illustrates the large amount of information that can be captured using the LM proposed in this work.

(iv) Materials caption graph: In addition to the queries as mentioned earlier, graph representations can provide in-depth insights into the information spread in figure captions. For instance, questions such as "which synthesis and characterization methods are commonly used for a specific material?", "what are the methods for measuring a specific property?" can be easily answered using knowledge graphs. Here, we demonstrate how the information in figure captions can be represented using materials caption graphs (MCG). To this extent, we first randomly select 10,000 figure captions from glass-related publications. Further, we extract the entities and their types from the figure captions using the MatSciBERT finetuned on Matscholar NER dataset. For each caption, we create a fully connected graph by connecting all the entities present in that caption. These graphs are then joined together to form a large MCG. We demonstrate some insights gained from the MCGs below.

Figure 5 shows two subsets of graphs extracted from the MCGs. In Fig. 5a, we identified two entities that are two-hop neighbors, namely, T_g and anneal. Note that these entities do not share an edge. In other words, these two entities are not found simultaneously in any given caption. We then identified the intersection of all the one-hop neighbors of both the nodes and plotted the graph as shown in Fig. 5a. The thickness of the edge represents the strength of the connection in terms of the number of occurrences. We observe that there are four common one-hop neighbors for T_g and anneal, namely, XRD, doped, glass, and amorphous. This means that these four entities occur in captions along with T_g and anneal, even though these two entities are not directly connected in the captions used for generating the graph. Figure 5a suggests that T_g is related to glass, amorphous, and doped materials and that these materials can be synthesized by annealing. Similarly, the structures obtained by annealing can be characterized by XRD. From these results, we can also infer that T_g is affected by annealing, which agrees with the conventional knowledge in glass science.

Similarly, Fig. 5b shows all the entities connected to the node XRD. To this extent, we select all the captions having XRD as CMT.

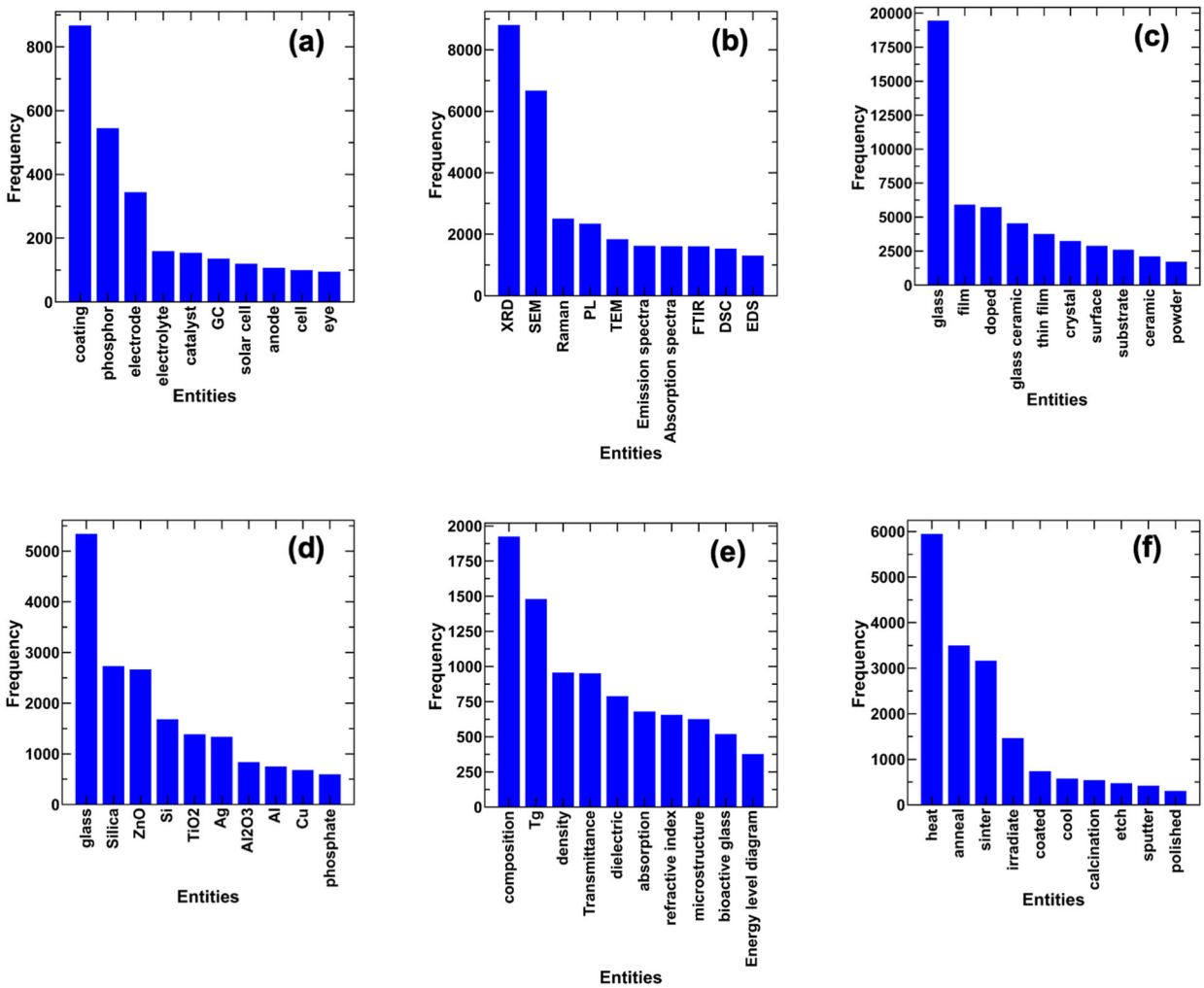

Fig. 3 Top-10 entities for various categories. a APL Application, **b** CMT Characterization method, **c** DSC Sample descriptor, **d** MAT Inorganic material, **e** PRO Material Property, and **f** SMT Synthesis method.

After obtaining all the entities in those captions, we randomly sample 20 pairs and then plotted them as shown in Fig. 5b. Note that the number of edges is 18 and the number of nodes is 19 because of one pair being (XRD, XRD) and two similar pairs (XRD, glass). The node color represents the entity type, and the edge width represents the frequency of the pair in the entire database of entities extracted from the captions where "XRD" is present. Using the graph, we can obtain the following information:

1. XRD is used as a characterization method for different material descriptors like glass, doped materials, nanofibers, and films.
2. Materials prepared using synthesis methods (SMT) like aging, heat-treatment, and annealing are also characterized using XRD.
3. While studying the property (PRO) glass transition temperature (Tg), XRD was also performed to characterize the samples.
4. In the case of silica glass ceramics (SGCs), phosphor, and phosphor-in-glass (PiG) applications (APL), XRD is used as CMT.
5. For different materials like ZnO, glasses, CsPBr3, yttria partially stabilized zirconia (YPSZ), XRD is a major CMT which is evident from the thicker edge widths.

Note this information covers a wide range of materials and applications in materials literature. Similar graphs can be generated for different entities and entity types using the MCG to gain insights into the materials literature.

(v) Other applications such as relation classification: MatSciBERT can also be applied for addressing several other issues such as relation classification and question answering. The relation classification task demonstrated in the present manuscript can provide key information regarding several aspects in materials science which are followed in a sequence. These would include synthesis and testing protocols, and measurement sequences. This information can be further used to discover an optimal pathway for material synthesis. In addition, such approaches can also be used to obtain the effect of different testing and environmental conditions, along with the relevant parameters, on the measured property of materials. This could be especially important for those properties such as hardness or fracture toughness, which are highly sensitive to sample preparation protocols, testing conditions, and the equipment used. Thus, the LM can enable the extraction of information regarding synthesis and testing conditions that are otherwise buried in the text.

At this juncture, it is worth noting that there are very few annotated datasets available for the material corpus. This contrasts with the biomedical corpus, where several annotated datasets are available for different downstream tasks such as relation extraction, question-answering, and NER. While the development of materials science specific language model can significantly accelerate the NLP-related applications in materials,

Captions with entities extracted using MatSciBERT NER model	Manual labels [10]
The comparison of XRD patterns of glass ceramic heat treated at 725 °C for 5h and rhombohedral Ba4Gd3F17. The superstructure reflections are marked with o. Inset: enlarged sections of XRD patterns.	Reflection
HRTEM image and the corresponding FFT pattern taken from as-deposited sample B(80/1) (a) and annealed sample D(30/1) (b); identifying rutile TiO2 crystal grains.	FFT
The illustrative schemes: a) The bonding of hexagonal ZnO nanocrystals to the glass surface. b) The structure of multi-layers coatings.	Crystal
(a) XRD patterns of the glass-ceramics sintered different holding times; (b) intensity of μ- and α-cordierite peaks count at (101) and (110) plane respectively as a function of sintering holding time.	XRD
Photoluminescence spectra of PbBr-based layered perovskites with an organic layer of naphthalene-linked ammonium molecules. Profiles: (a) 1; (b) 2; (c) 3; (d) 4; (e) 5.	Luminescence
Labels: Application (APL), Characterization method (CMT), Descriptor (DSC), Material (MAT), Property (PRO), Synthesis method (SMT), Symmetry/phase label (SPL)	

Fig. 4 Comparison of MatSciBERT based NER tagging with manually assigned labels. MatSciBERT-based NER model is able to extract multiple entities as compared to single manual label for each caption.

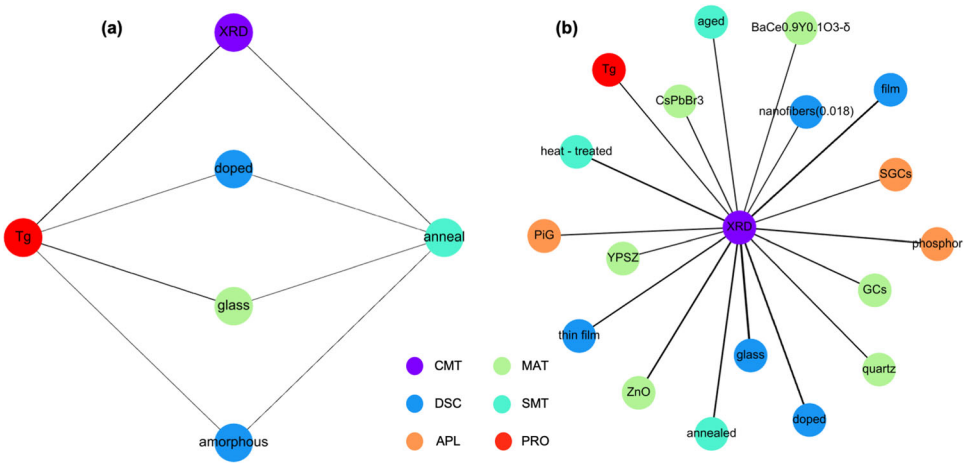

Fig. 5 Materials caption graph. a Connecting two unconnected entities, **b** exploring entities related to characterization method "XRD".

the development of annotated datasets is equally important for accelerating materials discovery.

In conclusion, we developed a materials-aware language model, namely, MatSciBERT, that is trained on materials science corpus derived from journals. The LM, trained from the initial weights of SciBERT, exploits the knowledge on computer science and biomedical corpora (on which the original SciBERT was pre-trained) along with the additional information from the materials domain. We test the performance of MatSciBERT on several downstream tasks such as document classification, NER, and relation classification. We demonstrate that MatSciBERT exhibits superior performance on all the datasets tested in comparison to SciBERT. Finally, we discuss some of the applications through which MatSciBERT can enable accelerated information extraction from the materials science text corpora. To enable accelerated text mining and information extraction, the pre-trained weights of MatSciBERT are made publicly available at https://huggingface.co/m3rg-iitd/matscibert.

METHODS

Dataset collection and preparation

In the training of an LM in a generalizable way, a considerable amount of dataset is required. For example, BERT[25] was pre-trained on BookCorpus[26] and English Wikipedia, containing a total of 3.3 billion words. SciBERT[21], an LM trained on scientific literature, was pre-trained using a corpus consisting of 82% papers from the broad biomedical domain and 18% papers from the computer science domain. However, we note that none of these LMs includes text related to the materials domain. Here, we consider materials science literature from four broad categories, namely, inorganic glasses and ceramics, metallic glasses, cement and concrete, and alloys, to cover the materials domain in a representative fashion.

The first step in retrieving the research papers is to query search from the Crossref metadata database[53]. This resulted in a list of more than 1 M articles. Although Crossref gives the search results from different journals and publishers, we downloaded papers only from the Elsevier Science Direct database using their sanctioned API[54]. Note that the Elsevier API returns the research articles in XML format; hence, we wrote a custom XML parser for extracting the text. Occasionally, there were papers having only abstract and not full text depending upon the journal and publication date. Since the abstracts contain concise information about the problem statement being discussed in the paper and what the research contributions are, therefore, we have included them in our corpus. Therefore, we have included all the sections of the paper when available and abstracts otherwise. For glass science-related papers, the details are given in our previous work[10]. For concrete and alloys, we first downloaded many research papers for each material category using several queries such as “cement”, “interfacial transition zone”, “magnesium alloy”, and “magnesium alloy composite materials”, to name a few.

Since all the downloaded papers did not belong to a particular class of materials, we manually annotated 500 papers based on their abstracts, whether they were relevant to the field of interest or not. Further, we finetuned SciBERT classifiers[21,55], one for each category of material, on these labeled abstracts for identifying relevant papers among the downloaded 1 M articles. We consider these selected papers from each category of materials for training the language model. A detailed description of the Materials Science Corpus (MSC) is given in the Results and Discussion section of the paper. Finally, we divided this corpus into training and validation, with 85% being used to train the language model and the remaining 15% as validation to assess the model’s performance on unseen text.

Note that the texts in the scientific literature may have several symbols, including some random characters. Sometimes the same semantic symbol has many Unicode surface forms. To address these anomalies, we also performed Unicode normalization of MSC to:

a. get rid of random Unicode characters like , , , and
b. map different Unicode characters having similar meaning and appearance to either a single standard character or a sequence of standard characters.

For example, % gets mapped to %, > to >, ⋙ to ⋙, = and = to =, ¾ to 3/4, to name a few. First, we normalized the corpus using BertNormalizer from the tokenizers library by Hugging Face[56,57]. Next, we created a list containing mappings of the Unicode characters appearing in the MSC. We mapped random characters to space so that they do not interfere during pre-training. It’s important to note that we also perform this normalization step on every dataset before passing it through the MatSciBERT tokenizer.

Pre-training of MatSciBERT

We pre-train MatSciBERT on MSC as detailed in the last sub-section. Pre-training LM from scratch requires significant computational power and a large dataset. To address this issue, we initialize MatSciBERT with weights from SciBERT and perform tokenization using the SciBERT uncased vocabulary. This has the additional advantage that existing models relying on SciBERT, which are pre-trained on biomedical and computer science corpora, can be interchangeably used with MatSciBERT. Further, the vocabulary existing in the scientific literature as constructed by SciBERT can be used to reasonably represent the new words in the materials domain.

To pre-train MatSciBERT, we employ the optimized training recipe, RoBERTa[49], suggested by Liu et al. (2019). This approach has been shown to significantly improve the performance of the original BERT. Specifically, the following simple modifications were adopted for MatSciBERT pre-training:

1. Dynamic whole word masking: It involves masking at the word level instead of masking at the wordpiece level, as discussed in the latest release of the BERT pre-training code by Google[58]. Each time a sequence is sampled, we randomly mask 15% of the words and let the model predict each masked wordpiece token independently.
2. Removing the NSP loss from the training objective: BERT was pre-trained using two unsupervised tasks: Masked-LM and Next-Sentence Prediction (NSP). NSP takes as input a pair of sentences and predicts whether the two sentences follow each other or not. RoBERTa authors claim that removing the NSP loss matches or slightly improves downstream task performance.
3. Training on full-length sequences: BERT was pre-trained with a sequence length of 128 for 90% of the steps and with a sequence of the length of 512 for the remaining 10% steps. RoBERTa authors obtained better performance by training only with full-length sequences. Here, input sequences are allowed to contain segments of more than one document and [SEP] token is used to separate the documents within an input sequence.
4. Using larger batch sizes: Authors also found that training with larger mini-batches improved the pre-training loss and increased the end-task performance.

Following these modifications, we pre-train MatSciBERT on the MSC with a maximum sequence length of 512 tokens for fifteen days on 2 NVIDIA V100 32GB GPUs with a batch size of 256 sequences. We use the AdamW optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.98$, $\varepsilon = 1e^{-6}$, weight decay $= 1e^{-2}$ and linear decay schedule for learning rate with warmup ratio $= 4.8\%$ and peak learning rate $= 1e^{-4}$. Pre-training code is written using PyTorch[59] and Transformers[57] library and is available at our GitHub repository for this work https://github.com/M3RG-IITD/MatSciBERT.

Downstream tasks

Once the LM is pre-trained, we finetune it on various supervised downstream tasks. Pre-trained LM is augmented with a task-specific output layer. Finetuning is done to adapt the model to specific tasks as well as to learn the task-specific randomly initialized weights present in the output layer. Finetuning is done on all the parameters end-to-end. We evaluate the performance of MatSciBERT on the following three downstream NLP tasks:

1. Named Entity Recognition (NER) involves identifying domain-specific named entities in a given sentence. Entities are encoded using the BIO scheme to account for multi-token entities[53]. Dataset for the NER task includes various sentences, with each sentence being split into multiple tokens. Gold labels are provided for each token. More formally, Let $E = \{e_1, \ldots e_k\}$ be the set of k entity types for a given dataset. If $[x_1, \ldots x_n]$ are tokens of a sentence and $[y_1, \ldots y_n]$ are labels for these tokens, then each $y_i \in L = \{\text{B-e}_1, \text{I-e}_1, \ldots \text{B-e}_k, \text{I-e}_k, \text{O}\}$. Here, B-$e_i$ and I-e_i represent the beginning and inside of entity e_i.
2. Input for the Relation Classification[60] task consists of a sentence and an ordered pair of entity spans in that sentence. Output is a label

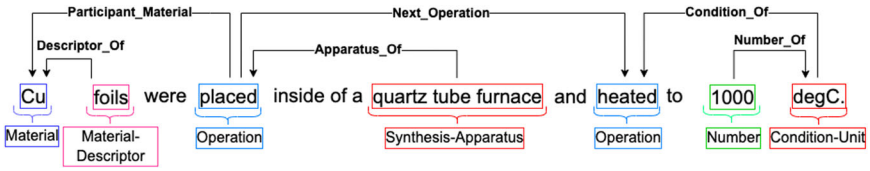

Fig. 6 Relation classification task. The different entities are enclosed in boxes with their respective labels. The related entities are connected using arrows labeled with the relation.

denoting the directed relationship between the two entities. The two entity spans can be represented as $s_1 = (i, j)$ and $s_2 = (k, l)$, where i and j denote the starting and ending index of the first entity and similarly k and l denote the starting and ending index of the second entity in the input statement. Here, $i \le j$, $k \le l$, and ($j < k$ or $l < i$). The last constraint guarantees that the two entities do not overlap with each other. The output label belongs to L, where L is a fixed set of relation types. An example of a sentence from the task is given in Fig. 6. The task is to predict the labels like "Participant_Material", "Apparatus_Of" given the sentence and pair of entities as input.

3. In the Paper Abstract Classification task, we are given an abstract of a research paper, and we have to classify whether the abstract is relevant to a given field or not.

Datasets

We use the following three Materials Science-based NER datasets to evaluate the performance of MatSciBERT against SciBERT:

1. Matscholar NER dataset[9] by Weston et al. (2019): This dataset is publicly available and contains seven different entity types. Training, validation, and test sets consist of 440, 511, and 546 sentences, respectively. Entity types present in this dataset are inorganic material (MAT), symmetry/phase label (SPL), sample descriptor (DSC), material property (PRO), material application (APL), synthesis method (SMT), and characterization method (CMT).
2. Solid Oxide Fuel Cells – Entity Mention Extraction (SOFC) dataset by Friedrich et al. (2020)[45]: This dataset consists of 45 open-access scholarly articles annotated by domain experts. Four different entity types have been annotated by the authors, namely Material, Experiment, Value, and Device. There are 611, 92, and 173 sentences in the training, validation, and test sets, respectively.
3. Solid Oxide Fuel Cells – Slot Filling (SOFC-Slot) dataset by Friedrich et al. (2020)[45]: This is the same as the above dataset except that entity types are more fine-grained. There are 16 different entity types, namely Anode Material, Cathode Material, Conductivity, Current Density, Degradation Rate, Device, Electrolyte Material, Fuel Used, Interlayer Material, Open Circuit Voltage, Power Density, Resistance, Support Material, Time of Operation, Voltage, and Working Temperature. Two additional entity types: Experiment Evoking Word and Thickness, are used for training the models.

For relation classification, we use the Materials Synthesis Procedures dataset by Mysore et al. (2019)[46]. This dataset consists of 230 synthesis procedures annotated as graphs where nodes represent the participants of synthesis steps, and edges specify the relationships between the nodes. The average length of a synthesis procedure is nine sentences, and 26 tokens are present in each sentence on average. The dataset consists of 16 relation labels. The relation labels have been divided into three categories by the authors:

a. Operation-Argument relations: Recipe target, Solvent material, Atmospheric material, Recipe precursor, Participant material, Apparatus of, Condition of
b. Non-Operation Entity relations: Descriptor of, Number of, Amount of, Apparatus-attr-of, Brand of, Core of, Property of, Type of
c. Operation-Operation relations: Next operation

The train, validation, and test set consist of 150, 30, and 50 annotated material synthesis procedures, respectively.

The dataset for classifying research papers related to glass science or not on the basis of their abstracts has been taken from Venugopal et al. (2021)[10]. The authors have manually labeled 1500 abstracts as glass and non-glass. These abstracts belong to different fields of glass science like bioactive glasses, rare-earth glasses, glass ceramics, thin-film studies, and optical, dielectric, and thermal properties of glasses, to name a few. We divide the abstracts into a train-validation-test split of 3:1:1.

Modeling

For NER task, we use the BERT contextual output embedding of the first wordpiece of every token to classify the tokens among $|L|$ classes. We model the NER task using three architectures: LM-Linear, LM-CRF, and LM-BiLSTM-CRF. Here, LM can be replaced by any BERT-based transformer model. We take LM to be BERT, SciBERT and MatSciBERT in this work.

1. LM-Linear: The output embedding of the wordpieces are passed through a linear layer with softmax activation. We use the BERT Token Classifier implementation of transformers library[57].
2. LM-CRF: We replace the final softmax activation of the LM-Linear architecture with a CRF layer[61] so that the model can learn to label the tokens belonging to the same entity mentioned and also learn the transition scores between different entity types. We use the CRF implementation of PyTorch-CRF library[62].
3. LM-BiLSTM-CRF: Bidirectional Long Short-Term Memory[63] is added in between the LM and CRF layer. BERT embeddings of all the wordpieces are passed through a stacked BiLSTM. The output of BiLSTM is finally fed to the CRF layer to make predictions.

In case of Relation Classification task, we use the Entity Markers-Entity Start architecture[60] proposed by Soares et al. (2019) for modeling. Here, we surround the entity spans within the sentence with some special wordpieces. We wrap the first and second entities with [E1], [\E1] and [E2], [\E2] respectively. We concatenate the output embeddings of [E1] and [E2] and then pass it through a linear layer with softmax activation. We use the standard cross-entropy loss function for the training of the linear layer and finetuning of the language model.

For the baseline, we use two recent models, MaxPool and MaxAtt, proposed by Maini et al. (2020)[50]. In this approach too, the pair of entities are wrapped with the same special tokens. Then glove embeddings[18] of words in the input sentence are passed through a BiLSTM, an aggregation mechanism (different for MaxPool and MaxAtt) over words, and a linear layer with softmax activation.

In Paper Abstract Classification task, we use the output embedding of the CLS token to encode the entire text/abstract. We pass this embedding through a simple classifier to make predictions. We use the BERT Sentence Classifier implementation of the transformers library[57]. For the baseline, we use a similar approach as relation classification except that there is no pair of input entities.

Hyperparameters

We use a linear decay schedule for the learning rate with a warmup ratio of 0.1. To ensure sufficient training of randomly initialized non-BERT layers, we set different learning rates for the BERT part and non-BERT part. We set the peak learning rate of the non-BERT part to 3e-4 and choose the peak learning rate of the BERT part from [$2e^{-5}$, $3e^{-5}$, $5e^{-5}$], whichever results in a maximum validation score averaged across three seeds. We use a batch size of 16 and an AdamW optimizer for all the architectures. For LM-BiLSTM-CRF architecture, we use a 2-layer stacked BiLSTM with a hidden dimension of 300 and dropout of 0.2 in between the layers. We perform finetuning for 15, 20, and 40 epochs for Matscholar, SOFC, and SOFC Slot datasets, respectively, as initial experiments exhibited little or no improvement after the specified number of epochs. All the weights of any given architecture are updated during finetuning, i.e., we do not freeze any of the weights. We make the code for finetuning and different architectures publicly available. We refer readers to the code for further details about the hyperparameters.

Evaluation metrics

We evaluate the NER task based on entity-level exact matches. We use the CoNLL evaluation script (https://github.com/spyysalo/conlleval.py). For NER and Relation Classification tasks, we use Micro-F1 and Macro-F1 as the primary evaluation metrics. We use accuracy to evaluate the performance of the paper abstract classification task.

DATA AVAILABILITY

Any data used for the work are available from the corresponding authors upon reasonable request. The PIIs and DOIs of the research papers used in this work are available at https://github.com/M3RG-IITD/MatSciBERT/blob/main/pretraining/piis_dois.csv.

CODE AVAILABILITY

All the codes used in the present work are available at https://github.com/M3RG-IITD/MatSciBERT. Also, the codes with finetuned models for the downstream tasks are available at https://doi.org/10.5281/zenodo.6413296.

Received: 29 October 2021; Accepted: 12 April 2022;
Published online: 03 May 2022

REFERENCES

1. Science, N. & (US), T. C. Materials genome initiative for global competitiveness. (Executive Office of the President, National Science and Technology Council, https://www.mgi.gov/sites/default/files/documents/materials_genome_initiative-final.pdf, 2011).
2. Jain, A. et al. Commentary: The Materials Project: A materials genome approach to accelerating materials innovation. *APL Mater.* **1**, 011002 (2013).
3. Zunger, A. Inverse design in search of materials with target functionalities. *Nat. Rev. Chem.* **2**, 1–16 (2018).
4. Chen, C. et al. A critical review of machine learning of energy materials. *Adv. Energy Mater.* **10**, 1903242 (2020).
5. de Pablo, J. J. et al. New frontiers for the materials genome initiative. *Npj Comput. Mater.* **5**, 1–23 (2019).
6. Greenaway, R. L. & Jelfs, K. E. Integrating computational and experimental workflows for accelerated organic materials discovery. *Adv. Mater.* **33**, 2004831 (2021).
7. Ravinder et al. Artificial intelligence and machine learning in glass science and technology: 21 challenges for the 21st century. *Int. J. Appl. Glass Sci.* **12**, 277–292 (2021).
8. Zanotto, E. D. & Coutinho, F. A. B. How many non-crystalline solids can be made from all the elements of the periodic table? *J. Non-Cryst. Solids* **347**, 285–288 (2004).
9. Weston, L. et al. Named entity recognition and normalization applied to large-scale information extraction from the materials science literature. *J. Chem. Inf. Model.* **59**, 3692–3702 (2019).
10. Venugopal, V. et al. Looking through glass: Knowledge discovery from materials science literature using natural language processing. *Patterns* **2**, 100290 (2021).
11. Zaki, M., Jayadeva & Krishnan, N. M. A. Extracting processing and testing parameters from materials science literature for improved property prediction of glasses. *Chem. Eng. Process. - Process Intensif.* 108607 (2021). https://doi.org/10.1016/j.cep.2021.108607.
12. El-Bousiydy, H. et al. What can text mining tell us about lithium-ion battery researchers' habits? *Batter. Supercaps* **4**, 758–766 (2021).
13. Brown, P. F., Della Pietra, V. J., deSouza, P. V., Lai, J. C. & Mercer, R. L. Class-based *n*-gram models of natural language. *Comput. Linguist.* **18**, 467–480 (1992).
14. Ando, R. K. & Zhang, T. A framework for learning predictive structures from multiple tasks and unlabeled data. *J. Mach. Learn. Res.* **6**, 1817–1853 (2005).
15. Blitzer, J., McDonald, R. & Pereira, F. Domain adaptation with structural correspondence learning. in *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing* 120–128 (Association for Computational Linguistics, 2006).
16. Mikolov, T., Chen, K., Corrado, G. & Dean, J. Efficient estimation of word representations in vector space. in *1st international conference on learning representations, ICLR 2013, scottsdale, arizona, USA, may 2-4, 2013, workshop track proceedings* (eds. Bengio, Y. & LeCun, Y.) (2013).
17. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S. & Dean, J. Distributed representations of words and phrases and their compositionality. in *Advances in Neural Information Processing Systems* vol. 26 (Curran Associates, Inc., 2013).
18. Pennington, J., Socher, R. & Manning, C. GloVe: Global vectors for word representation. in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* 1532–1543 (Association for Computational Linguistics, 2014).
19. Lee, J. et al. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics* **36**, 1234–1240 (2020).
20. Gururangan, S. et al. Don't stop pretraining: Adapt language models to domains and tasks. in *Proceedings of the 58th annual meeting of the association for computational linguistics* 8342–8360 (Association for Computational Linguistics, 2020).
21. Beltagy, I., Lo, K. & Cohan, A. 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 2019, hong kong, china*, november 3-7, 2019 (eds. Inui, K., Jiang, J., Ng, V. & Wan, X.) 3613–3618 (Association for Computational Linguistics, 2019).
22. Araci, D. FinBERT: Financial sentiment analysis with pre-trained language models. Preprint at https://arxiv.org/abs/1908.10063 (2019).
23. Lee, J.-S. & Hsiang, J. Patent classification by fine-tuning BERT language model. *World Pat. Inf.* **61**, 101965 (2020).
24. Manning, C. & Schutze, H. *Foundations of Statistical Natural Language Processing*. (MIT Press, 1999).
25. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. in *NAACL-HLT (1)* 4171–4186 (Association for Computational Linguistics, 2019).
26. Zhu, Y. et al. Aligning books and movies: towards story-like visual explanations by watching movies and reading books. in *2015 IEEE International Conference on Computer Vision (ICCV)* 19–27 (2015).
27. Swain, M. C. & Cole, J. M. ChemDataExtractor: A toolkit for automated extraction of chemical information from the scientific literature. *J. Chem. Inf. Model.* **56**, 1894–1904 (2016).
28. Huang, S. & Cole, J. M. A database of battery materials auto-generated using ChemDataExtractor. *Sci. Data* **7**, 260 (2020).
29. Court, C. J. & Cole, J. M. Auto-generated materials database of Curie and Néel temperatures via semi-supervised relationship extraction. *Sci. Data* **5**, 180111 (2018).
30. Kononova, O. et al. Text-mined dataset of inorganic materials synthesis recipes. *Sci. Data* **6**, 203 (2019).
31. Uvegi, H. et al. Literature mining for alternative cementitious precursors and dissolution rate modeling of glassy phases. *J. Am. Ceram. Soc.* **104**, 3042–3057 (2020).
32. Jensen, Z. et al. A machine learning approach to zeolite synthesis enabled by automatic literature data extraction. *ACS Cent. Sci.* **5**, 892–899 (2019).
33. Guha, S. et al. MatScIE: An automated tool for the generation of databases of methods and parameters used in the computational materials science literature. *Comput. Mater. Sci.* **192**, 110325 (2021).
34. Olivetti, E. A. et al. Data-driven materials research enabled by natural language processing and information extraction. *Appl. Phys. Rev.* **7**, 041317 (2020).
35. Jessop, D. M., Adams, S. E., Willighagen, E. L., Hawizy, L. & Murray-Rust, P. OSCAR4: a flexible architecture for chemical text-mining. *J. Cheminformatics* **3**, 41 (2011).
36. Epps, R. W. et al. Artificial chemist: an autonomous quantum dot synthesis bot. *Adv. Mater.* **32**, 2001626 (2020).
37. MacLeod, B. P. et al. Self-driving laboratory for accelerated discovery of thin-film materials. *Sci. Adv.* **6**, eaaz8867 (2020).
38. Zhang, Y., Chen, Q., Yang, Z., Lin, H. & Lu, Z. BioWordVec, improving biomedical word embeddings with subword information and MeSH. *Sci. Data* **6**, 52 (2019).
39. Ammar, W. et al. Construction of the Literature Graph in Semantic Scholar. in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Volume 3 *(Industry Papers)* 84–91 (Association for Computational Linguistics, 2018).
40. Alsentzer, E. et al. Publicly available clinical BERT embeddings. in *Proceedings of the 2nd Clinical Natural Language Processing Workshop* 72–78 (Association for Computational Linguistics, 2019).
41. Johnson, A. E. W. et al. MIMIC-III, a freely accessible critical care database. *Sci. Data* **3**, 160035 (2016).
42. Libovický, J., Rosa, R. & Fraser, A. On the language neutrality of pre-trained multilingual representations. in Findings of the association for computational linguistics: EMNLP 2020 1663–1674 (Association for Computational Linguistics, 2020)
43. Gupta, T., Zaki, M., Krishnan, N. M. A., & Mausam. MatSciBERT: A materials domain language model for text mining and information extraction. Preprint at https://arxiv.org/abs/2109.15290. (2021).
44. Walker, N. et al. The impact of domain-specific pre-training on named entity recognition tasks in materials science. *Available SSRN 3950755* (2021).
45. Friedrich, A. et al. 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 1255–1268 (Association for Computational Linguistics, 2020).
46. Mysore, S. et al. The materials science procedural text corpus: Annotating materials synthesis procedures with shallow semantic structures. in *Proceedings of the 13th linguistic annotation workshop* 56–64 (Association for Computational Linguistics, 2019). https://doi.org/10.18653/v1/W19-4007.
47. Wu, Y. et al. Google's neural machine translation system: Bridging the gap between human and machine translation. Preprint at https://arxiv.org/abs/1609.08144. (2016).
48. Tokenizer. https://huggingface.co/transformers/main_classes/main_classes/tokenizer.html.
49. Liu, Y. et al. RoBERTa: A robustly optimized BERT pretraining approach. Preprint at https://arxiv.org/abs/1907.11692. (2019).
50. Maini, P., Kolluru, K., Pruthi, D., & Mausam. Why and when should you pool? analyzing pooling in recurrent architectures. in *Findings of the association for computational linguistics*: *EMNLP 2020* 4568–4586 (Association for Computational Linguistics, 2020).
51. Sainburg, T., McInnes, L. & Gentner, T. Q. Parametric UMAP embeddings for representation and semisupervised learning. *Neural Comput.* **33**, 2881–2907 (2021).
52. Goodfellow, I., Bengio, Y. & Courville, A. *Deep learning*. (MIT Press, 2016).
53. allenai/scibert_scivocab_uncased · Hugging Face. https://huggingface.co/allenai/scibert_scivocab_uncased.
54. Hugging Face. *GitHub* https://github.com/huggingface.
55. Wolf, T. et al. Transformers: State-of-the-art natural language processing. in *Proceedings of the 2020 conference on empirical methods in natural language processing: System demonstrations* 38–45 (Association for Computational Linguistics, 2020).
56. bert/run_pretraining.py at master · google-research/bert. *GitHub* https://github.com/google-research/bert.
57. Paszke, A. et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. 12.
58. Crossref Metadata Search. https://search.crossref.org/.
59. Elsevier Developer Portal. https://dev.elsevier.com/.
60. Baldini Soares, L., FitzGerald, N., Ling, J. & Kwiatkowski, T. Matching the blanks: Distributional similarity for relation learning. in *Proceedings of the 57th annual meeting of the association for computational linguistics* 2895–2905 (Association for Computational Linguistics, 2019).
61. Lafferty, J. D., McCallum, A. & Pereira, F. C. N. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. in *ICML* 282–289 (Morgan Kaufmann, 2001).
62. pytorch-crf — pytorch-crf 0.7.2 documentation. https://pytorch-crf.readthedocs.io/en/stable/.
63. Huang, Z., Xu, W. & Yu, K. Bidirectional LSTM-CRF models for sequence tagging. Preprint at https://arxiv.org/abs/1508.01991 (2015).

ACKNOWLEDGEMENTS

N.M.A.K. acknowledges the funding support received from SERB (ECR/2018/002228), DST (DST/INSPIRE/04/2016/002774), BRNS YSRA (53/20/01/2021-BRNS), ISRO RESPOND as part of the STC at IIT Delhi. M.Z. acknowledges the funding received from the PMRF award by Government of India. M. acknowledges grants by Google, IBM, Bloomberg, and a Jai Gupta chair fellowship. The authors thank the High Performance Computing (HPC) facility at IIT Delhi for computational and storage resources.

AUTHOR CONTRIBUTIONS

M. and N.M.A.K. supervised the work. T.G. developed the codes and trained the models. M.Z. along with T.G. performed data collection and processing. All the authors analyzed the results and wrote the manuscript.

COMPETING INTERESTS

The authors declare no competing interests.

ADDITIONAL INFORMATION

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41524-022-00784-w.

Correspondence and requests for materials should be addressed to N. M. Anoop Krishnan or Mausam.

Reprints and permission information is available at http://www.nature.com/reprints

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.