How to pretrain an efficient cross-disciplinary language model: The ScilitBERT use case

Jean-Baptiste de la Broise
MDPI
Basel, Switzerland
jeanbaptiste.delabroise@mdpi.com

Nolwenn Bernard
MDPI
Basel, Switzerland

Jean-Philippe Dubuc
MDPI
Basel, Switzerland

Andrea Perlato
MDPI
Basel, Switzerland

Bastien Latard
MDPI
Basel, Switzerland

2021 6th International Conference on Information Technology Research (ICITR) | 978-1-6654-2000-6/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICITR54349.2021.9657164

***Abstract*—Transformer based models are widely used in various text processing tasks, such as classification, named entity recognition. The representation of scientific texts is a complicated task, and the utilization of general English BERT models for this task is suboptimal. We observe the lack of models for multi-disciplinary academic texts representation, and on a broader scale, a lack of specialized models pretrained on specific domains, for which general English BERT models are suboptimal.**

This paper introduces ScilitBERT, a BERT model pretrained on an inclusive cross-disciplinary academic corpus. ScilitBERT is half as deep as RoBERTa, and has a much lower pretraining computation cost. ScilitBERT obtains at least 96% of RoBERTa's accuracy on two academic domain downstream tasks. The presented cross-disciplinary academic model has been publicly released[1]. The results obtained show that for domains that use a technolect and have a sizeable amount of raw text data; the pretraining of dedicated models should be considered and favored.

***Index Terms*—language models; clustering, classification, and association rules; benchmarking; text analysis**

I. Introduction

Several domains use precise English technolect and writing conventions. These domains tend to produce plenty of raw text data, *e.g.*, journalistic writing. Many actors in these domains tend to apply natural language understanding techniques, and are therefore interested in the progress of domain-specific natural language understanding. We argue that the academic domain belongs to such domains. Indeed, several papers showed evidence that academic writing varies from general English in several ways, such as the type of words being used, and the abstract nature of the discourse [1]. Academic writing follows precise conventions and vocabulary. Moreover, the academic domain produces a lot of raw text data quickly. The rate of academic publication release is constantly increasing [2]. The will to process and analyze this flow of data calls for the use of automated methods. In the field of academic editing, natural language processing (hereinafter NLP) tools can: generate metadata, categorize documents, help authors find institutions to publish with [3], guide readers with reading recommendations, etc. Consequently, the academic domain is suitable to conduct a study on domain-specific model training.

[1]https://github.com/JeanBaptiste-dlb/ScilitBERT

Transformer-based models [4] such as BERT [5] are promising for many NLP applications. Indeed, BERT-based models can be pretrained on raw text and then fine-tuned on supervised tasks. Pretraining gives the model an understanding of the underlying writing rules used in the pretraining corpus domain (*e.g.*, grammar, style). This baseline knowledge allows the model to achieve outstanding performances on many types of supervised downstream tasks. A paradigm shift came with the release of large, capable, and available pretrained models for general English understanding by some companies, *e.g.*, FacebookAI RoBERTa [6], and OpenAI GPT-2 [7]. The availability of those highly pretrained models raises a question: is it still necessary to train specialized models for domain-specific tasks?

The first noticeable advantage of domain-specific models comes from the limited number of words that a model can learn to represent. Intuitively, the model cannot learn and store an infinite number of embeddings representing that many words. This limitation is the reason that a Transformer-based model has a limited vocabulary generated using a subword segmentation algorithm. This type of algorithm dedicates a token to words and subwords that are important in the corpus. The model will learn a tailored embedding representation for every token in its vocabulary. The representation of a word that does not have a dedicated token is a combination of its subword embeddings. For example, a representation of the word "likelihood" could be a combination of "like" and "lihood" subword tokens' embedding. Given enough data to learn an accurate embedding representation for the word "likelihood", the word embedding for "likelihood" will be a better overall representation of this word than any linear combination of the embeddings for the subwords "like" and "lihood".

In this context, a theoretically ideal model would be the combination of the following: (1) a vocabulary with one token per word, (2) enough training data to learn an accurate representation for each of those tokens, (3) enough computing power to train the said model, (4) enough memory to store the embeddings for every token, and a large number of weights. As no one can create such a model, some domains require a dedicated model to reach their best performances.

Other arguments in favor of domain-specific pretraining are, (1) the control over the pretraining data: general English

models tend to train on abundant low-quality data, leading to biases in the resulting models. (2) The control over the architecture: Many general English models use deep architectures; one may want to use a less deep architecture, to train with limited resources, or to quickly generate an embedding for every document in a database. (3) To get a better memory load/performance ratio: When using a general English model on a task that involves domain-specific data, most of the potency of the model remains unused. Each unused weight or embedding has a cost, as it makes the model slower and heavier than it should be.

Considering those arguments, general English models still rival domain-specific ones on their domain, because the pretraining resources of models such as RoBERTa tend to be unattainable by narrow domains' communities. Nonetheless, the intuition that the academic domain is specific enough to benefit from a dedicated model is tested as this work introduces ScilitBERT, an academic-specific language representation model. ScilitBERT is based on a BERT [5] architecture and applies some of the best practices enlightened by RoBERTa [6]; ScilitBERT is pretrained on a broad corpus of academic articles abstracts. It is then compared to RoBERTa and SciBERT [8] using two academic language understanding benchmarks, namely "Journal Finder" and "Web of Science (Hereinafter WoS) topic classification".

II. Related Work

Text representation is the task of obtaining the most semantically accurate representation of a text document in a vector space. Two heavily used word representation methods are Glove [9] and word2vec [10]. These techniques are based, respectively, on matrix factorization and representation learning. These techniques can represent the semantics of words (this representation is fixed and context-independent). A mean of document words' vectors can be a decent representation for the document. Other works [11], [12] have already investigated the benefits of domain-specific training for these models.

The Transformer architecture [4] improves greatly upon previous representation methods, even if it was initially intended as a sequence-to-sequence model. The Transformer, using a self-attention mechanism, generates a contextualized embedding for each word in a document. A Transformer target word's embedding encapsulates the absolute embedding of the target word, and the meaning of the document's words that are relevant to understand the target word in its context. A neural network evaluates the "relevance" of each word in the document toward understanding the target word. As a sequence-to-sequence model, the Transformer is composed of an encoder and a decoder. A property of the encoder is that it learns to represent an input document [13] and each word it contains, thus making it useful for tasks outside the sequence-to-sequence spectrum.

BERT [5] leverages the representation properties of the Transformer's encoder. It stacks encoder layers to build a powerful language representation model. BERT also leverages self-supervised bidirectional training objectives, such as masked language modeling (MLM) and next sentence prediction (NSP), allowing for the efficient self-supervised pretraining of the model on raw text data. BERT architecture was further studied in RoBERTa's paper [6]. This paper lists, analyzes, and shows some best practices that should be used to pretrain a BERT model. Some of these best practices are used to pretrain ScilitBERT; see Section IV-B.

The proposed work is related to numerous attempts at pretraining a BERT model for academic text representation [8], [14]. Those models focus on the biomedical sub-domain of the academic domain. Both pretrained a BERT model from scratch [14] instead of taking the mixed-domain training route, as advised for low-resource training [15]. This work also takes the training from scratch path, as it allows the model to leverage the benefits of using a domain-specific vocabulary. Both SciBERT and BiomedBERT demonstrated state-of-the-art performances in many biomedical domain-specific tasks.

Another work on academic language representation is OAG-BERT [16]. OAG stands for open academic graph. The idea is to contextualize and enrich the training samples using heterogeneous metadata, *e.g.*, the author affiliations, their field of studies, etc. The approach leverages metadata to increase its performances on several tasks but, for many applications, such as academic writing enhancement, the added metadata are irrelevant; sometimes, the metadata are also unavailable.

III. Academic domain corpus

To pretrain a BERT model using a self-supervised objective, a large corpus is required. For this purpose, a pretraining corpus containing abstracts found on Scilit was created. Scilit API is used to fetch articles published between January 2017 and March 2021. Scilit contains articles from various fields, including, but not limited to, chemistry, computer science, psychology, and geopolitics. This diversity does not allow for a clear understanding of the distribution of each field in the corpus. This problem is addressed in Section VI-B1.

To improve the overall quality of the corpus, some text preprocessing is applied on the raw corpus fetched from Scilit. Text preprocessing is an important step when building a model to obtain better results. In this work, the raw corpus is processed to: (1) remove empty abstracts and duplicates, (2) remove non-English characters, (3) exclude abstracts that contain less than four sentences, (4) convert HTML entities representing characters to UTF-8.

This data cleaning process removes approximately 50% of the fetched documents. The resulting corpus is 12 GB in total, and it contains 9.1 million abstracts, with a total of 1.8 billion words.

IV. ScilitBERT's Pretraining

This section describes the method and tools used to pretrain ScilitBERT on the pretraining corpus.

A. ScilitBERT's Tokenizer

To pretrain ScilitBERT, a tokenizer is required. Indeed, as a BERT-based model, ScilitBERT uses a finite vocabulary,

which is necessary to train on the MLM self-supervised objective. Moreover, learning to represent words that do not occur enough in the corpus is useless. ScilitBERT's vocabulary is generated using the byte-pair encoding algorithm.

1) Byte-Pair Encoding Algorithm: The tokenizer construction follows the byte-level byte-pair encoding method, hereafter BBPE [17], a more compact version of the byte-pair encoding, hereafter BPE [18].

BPE selects the most appropriate tokens to build a vocabulary that can represent the corpus. It starts by making a token out of each character found in the training input alphabet. It then follows a rule based on token pair frequency, to fuse two tokens and form a new one. After generating the character-level tokens, the first subword will be the most frequent combination of two character tokens found in the corpus. The token fusion step is repeated until the vocabulary has reached the requested size. A subword tokenization method is chosen as it allows a good control of the vocabulary size, and a complete representation of the pretraining corpus. As ScilitBERT follows a RoBERTa approach, the choice was made to use BBPE instead of WordPiece. These methods are similar, and therefore, this choice is inconsequential.

2) Choices for ScilitBERT's tokenizer: Our tokenizer has some characteristics, that are detailed below:

- ScilitBERT's tokenizer is case-sensitive, and so are the ones of the models selected for comparison. The main benefit is disambiguation, *e.g.*, "STAR", meaning "Satellite for Telecommunication Applications and Research" and the object "a star".
- ScilitBERT's tokenizer does not consider sequences of digits, *e.g.*, "42" cannot be a token in its vocabulary. This choice is made because, in other models' vocabulary, the digits take up a sizeable amount of space, *e.g.*, 1769 of 52,000 tokens for RoBERTa's vocabulary.

3) Tokenizers' Comparison: ScilitBERT's tokenizer is compared to a general language model's tokenizer (RoBERTa) and a scientific domain model's tokenizer (SciBERT).

a) ScilitBERT's and RoBERTa's Tokenizer Comparison: A comparison between the vocabularies of ScilitBERT and RoBERTa-large (hereinafter RoBERTa's vocabulary) shows that only 41% of ScilitBERT's tokens are also present in RoBERTa's vocabulary; *e.g.*, "proteases", "catalysts", and "autoregressive" are in ScilitBERT's vocabulary, but not in RoBERTa's vocabulary. With enough data and training, our model will better represent each token that is not present in RoBERTa's vocabulary.

Indeed, the ability for RoBERTa to learn the meaning of a word, such as "prophylactic" tokenized with the subwords "pro", "ph", "yl", "actic" by its tokenizer, is questionable at best. Those subwords are encountered in many different words that share nothing with the word "prophylactic". Consequently, it is highly unlikely for RoBERTa to extract any meaningful information when encountering said word. There are many other words that RoBERTa cannot meaningfully tokenize, *e.g.*, "isotherms" $\rightarrow$ "is", "other", "ms". Both "prophylactic" and "isotherms" have a dedicated token in ScilitBERT's vocabulary, meaning that these words are frequently encountered in academic literature. Consequently, it is important for a model that works on academia-related tasks to learn a dedicated representation of these words.

b) ScilitBERT's and SciBERT's Tokenizer Comparison: In addition to the BBPE tokenizer, a WordPiece tokenizer [19], [20] is trained on our corpus. This new tokenizer will allow for a better comparison with SciBERT's tokenizer, as SciBERT's tokenizer is a WordPiece tokenizer. Both vocabularies contain approx. 32,000 tokens. ScilitBERT and SciBERT share 50% of their vocabulary.

Frequent tokens found in ScilitBERT, but not in SciBERT, include coronavirus-related terms, *e.g.*, "COVID", "pandemic", and geographical/nationality tokens, *e.g.*, "Chinese", and "American". These geographical tokens come mainly from Scilit geopolitical papers. Frequent tokens found in SciBERT, but not in ScilitBERT, include biomedical vocabulary, *e.g.*, "nanocompos", "thrombocytop", "arthro", and computer science-related vocabulary, *e.g.*, "telecomm", "neurode", and "simul" ("simulation" and "Telecommunications" are in ScilitBERT's vocabulary).
Words found in SciBERT's vocabulary, but not in ScilitBERT's, are precise biomedical domain terms or subwords for which ScilitBERT's vocabulary contains the sur-words.

B. Model Architecture

The model is based on a BERT architecture [5], it follows some of RoBERTa's best practices. The core differences between the original BERT and RoBERTa are as follows: (1) the removal of the next-sentence prediction objective. (2) Dynamic masking, which allows for the pretraining of more epochs with less over fitting. (3) The removal of the input_type_ids field in the inputs. This removal allows more sample to be fit into the memory and, consequently, allows for an increase in batch size, which is beneficial for the training [6].

ScilitBERT is tweaked to adapt to limited GPU resources. It has half as many encoder layers as RoBERTa and SciBERT, and it is consequently twice as fast during training and inference. This difference in depth does not greatly change the overall weight of the model, as the embedding table represents 80% of ScilitBERT's memory load, i.e., ScilitBERT, while being half as deep as RoBERTa, is only 10% lighter in memory.

C. Pretraining

1) Hyper-Parameters: ScilitBERT's pretraining was completed using the parameters in Table I.

2) Training GPU time: An estimate of the processing time for RoBERTa and ScilitBERT:

- RoBERTa: 123,000 cumulated V100 TPU hours
- ScilitBERT: 160 RTX Titan hours

The perplexity of ScilitBERT decreased during the whole training, and was not decreasing enough during the last hours of pretraining to justify further pretraining. The results should be viewed through the scope of those pretraining

TABLE I
PRETRAINING CONFIGURATION FOR SCILITBERT AND ROBERTA. PARAMETERS THAT SHARE VALUES ARE NOT DISPLAYED AND CAN BE FOUND IN ROBERTA'S PAPER APPENDIX.

parameter	ScilitBERT	RoBERTa-base
dataset size	12 Gb	160 Gb
hardware	1 Titan RTX	1024 v100 TPU
Adam beta 2	0.999	0.98
Adam epsilon	1e-8	1e-6
batch size	20	8000
peak learning rate	5e-5	6e-4
epoch	2	UNKNOWN
steps	910,000	500,000

durations. In its current state, ScilitBERT is under-trained. However, RoBERTa's improvement on downstream tasks after 100,000 steps is very little— $+2\%$ at most on four different benchmarks for five times as much pretraining[2]— whereby, ScilitBERT can potentially achieve good results with a fraction of RoBERTa's pretraining time.

D. Evaluation Tasks

As the model is specialized in a domain, the usual benchmarks are not usable for the evaluation (*e.g.*, GLUE [21], SQUAD [22]), as these benchmarks are too general to evaluate the academic language understanding. We develop two academic language understanding benchmarks named "Journal Finder" and "WoS topic classification". These benchmarks are used to test ScilitBERT against the following models. (1) RoBERTa: a model achieving great performances in many general English language understanding tasks. (2) SciBERT: a BERT language model, trained on the full-text of 1.14 million papers in computer science and biomedical topics.

1) Journal Finder: Given an article title and abstract, the task is to find the journal in which it is published. The dataset is an aggregation of articles, the composition of a dataset entry is an article title, its abstract, and a label. The label identifies the journal in which the paper is published. The same cleaning process as for the pretraining corpus is applied; see Section III. The articles are selected among the ones released by MDPI before February 2021. A journal is taken into account if 200 usable articles are found within it. This process isolates 167 journals.

The dataset contains approx. 410,000 training samples, 45,000 testing samples and 41,000 validation samples; the split is stratified because the dataset is highly unbalanced. The task is difficult, as some journals' topics overlap. These overlaps are perceptible in the UMAP [23] representation of the fine-tuned model articles' embeddings. The embeddings are studied to highlight intersections between some journal topics.

Let J be a journal composed by a set of articles, each represented by Cartesian coordinates in 768 dimensions. The coordinates for an article are the components of its embedding generated by ScilitBERT fine-tuned on the journal finder task. An embedding is, in this context, the output of the last hidden layer of the model. The UMAP algorithm is used to acquire insights into the distribution of articles belonging to J. An article's embedding dimension is reduced from 768 to 32 using UMAP. A journal, such as J, is represented by the centroid of its articles' embeddings. An analysis is conducted using the Euclidean distance between journal centroids (hereafter $dist$):

- *International Journal of Molecular Science*, *Nutrients* and *Molecules*: for each of these journal pairs, $dist \leq 3.3$. Meanwhile, the mean of the distance between two distinct journals is 6.9. The three journals intersect.
- Social sciences articles (*Religions* and *Humanities*) are very well separated from the rest, $dist \geq 10.7$. These journals are also close to one another, with $dist = 1.6$; however, they do not tend to overlap, as they are compact. Indeed, for these journals, the median distance of an article to the journal centroid is inferior to 0.12, whereas the mean of this measure is 1.18 across the set of journals.
- The journal *Applied Sciences* is confusing, as some of its articles can be found in each scientific journal's area. The median distance of this journal's articles to its centroid is 2.7; this median is inferior to 1.4 for every other journal.

2) Web of Science (WoS) Topic Classification: Another classification task is used to evaluate ScilitBERT; the goal is to predict the category to which an article belongs, using its title and abstract. This task is not only a benchmark, as it is also used in practice to determine the underlying distribution of data in the pretraining dataset; see Section VI-B1. Some characteristics of the dataset include: (1) the representation of 243 topics, (2) a relatively good balance with approx. 2000 articles per topic, (3) a good overall separation of the topics. This dataset cannot be released, as it contains some papers that are not open access. The dataset contains approx. 448,300 articles in the train set, 11,500 articles in the test set and 11,100 in the validation set. Data are extracted using the WoS API.

V. FINE-TUNING

TABLE III
HYPER-PARAMETERS FOR FINE-TUNING ON BOTH TASKS

h-param	value
learning rate	3e-5
beta 1	0.9
beta 2	0.999
epochs	3
weight decay	0.01
fp16	True
batch size	10[a]

[a] 16 for ScilitBERT and RoBERTa on the Journal Finder task.

For each competing model, fine-tuning is performed using the same hyper-parameters. Only the batch size increases for ScilitBERT and RoBERTa on the journal finder task, because of the different input sizes caused by the added input_type_ids field in SciBERT model inputs. Using the same parameters gives an advantage to deeper models (RoBERTa, SciBERT) as they train on the same amount of data, but update more weights. The size of those models allows for better information retention at the cost of an increased computation complexity. Indeed, forward path and backpropagation take twice as long

[2]Table 4 in RoBERTa's paper [6]

TABLE II
FINE-TUNED MODEL PERFORMANCES ON THE JOURNAL FINDER AND WEB OF SCIENCE (WOS) TASKS.

Journal Finder	ScilitBERT	RoBERTa	SciBERT	ScilitBERT no pretrain[a]
accuracy/F1[b]	55.6/0.35	56.0/0.36	**56.3/0.37**	52.3/0.27
top-5 accuracy	89.4	89.6	**90.1**	86.1
top-10 accuracy	95.8	96.0	**96.2**	93.6
WoS topic classification				
accuracy/F1	77.7/0.76	**80.9/0.80**	79.1/0.78	59.4/0.56
top-5 accuracy	96.9	**97.6**	97.3	87.3
top-10 accuracy	99.0	**99.2**	99.0	94.2

[a] Uses ScilitBERT's tokenizer and architecture [b] macro-averaged

for those models as for ScilitBERT. The fine-tuning is short; only three epochs. The goal is to retain the value of pretraining to assess its benefit. To conduct this assessment, a RoBERTa model that is not pretrained, but has the same characteristics, as ScilitBERT (vocabulary, depth) is added to the experiments. It is not a competitor and is used as a baseline model to acquire insights regarding the impact of pretraining on the downstream performances.

VI. RESULTS

The results for both tasks are listed in Table II. The top-k accuracy is used to evaluate the models' performances, as top-1 accuracy tends to be low on tasks with that many classes.

A. Journal Finder

For the fine-tuning of the journal finder task, the pretraining accounts for approx. 6% of ScilitBERT's accuracy; this classification task relies less on pretraining compared to the WoS topic classification task VI-B. Some other tasks, requiring deeper language understanding, will better leverage the benefits of pretraining. Indeed, the purpose of pretraining is to give the model a working knowledge of the language. A classification task does not rely on this knowledge as much as some other tasks, such as summarization. The classification of a document does not rely on a deep understanding of grammar. Meanwhile, the summarization task requires the model to understand a text, assess the importance of different elements in it, and generate a text that follows the language rules and transmits the main ideas.

ScilitBERT achieves 99.3% of RoBERTa's accuracy on this task, while being pretrained for 0.1% of the RoBERTa pretraining duration and fine-tuned for half the duration of RoBERTa's fine-tuning. ScilitBERT is also half as deep as RoBERTa; this difference in depth causes the computation of a prediction and the backpropagation to be twice as fast on ScilitBERT than on RoBERTa, i.e., on average, the inference duration for one sample on this task is $5.42ms$ for ScilitBERT and $10.76ms$ for SciBERT and RoBERTa (both experiments ran on the same device in overall similar conditions).

ScilitBERT reaches 98.8% of SciBERT's accuracy on the journal finder task; the top 10 accuracy for both models is close. It means that, for an application such as suggesting 10 journals to an author in which his article could fit, the choice of the model will not matter, and ScilitBERT will still be twice as fast, making it the better choice.

SciBERT's table of embeddings contains only 32,000 embeddings corresponding to that many tokens in the vocabulary; ScilitBERT has 52,000 embeddings. Consequently, ScilitBERT can represent more words and represent documents faster than SciBERT, while maintaining high accuracy on this task.

B. Web of Science Topic Classification

ScilitBERT achieves qualitative results, considering the pretraining time and the depth of the model. During fine-tuning, ScilitBERT was ahead for the 60,000 first steps (approx. 1.5 epochs), which means that ScilitBERT's pretraining is indeed useful. This also means that ScilitBERT is a potentially good few-shot learner for academic tasks.

The unpretrained RoBERTa model is, for this task much weaker than the pretrained models. It lies 18.3% below ScilitBERT for the top-1 accuracy. This task can better reveal the value of pretraining than the journal finder task. For this task, ScilitBERT reaches 96% of RoBERTa's accuracy and 98.2% of SciBERT's accuracy.

TABLE IV
DISTRIBUTION OF CORPUS DATA FOLLOWING 45 CATEGORIES. THE PROPORTIONS FOR THE TOP 19 CATEGORIES ARE REPRESENTED.

Category	proportion	Category	proportion
UNCERTAIN	12.0%	BODY	3%
MEDICAL	10.9%	VISUALIZATION	2.8%
BIOLOGY	6.6%	MATERIALS	2.7%
PHYSICS	5.6%	HEALTH	2.1%
ENGINEERING	5.5%	PSYCHOLOGY	1.9%
COMPUTER SCIENCE	4.6%	EDUCATION	1.8%
DISEASES	4.2%	DATA SCIENCE	1.7%
FOOD	3.3%	INDUSTRY	1.6%
ENVIRONMENT	3.2%	SOCIAL SCIENCES	1.4%
CHEMISTRY	3.2%	OTHER	21.9%

1) Distribution of Categories in the Pretraining Corpus Using Web of Science Topic Classification: When the pretraining corpus was generated, there were no data on the topics found in it. It is interesting to know the distribution of the data in the pretraining corpus to know the expected strengths and weaknesses of the model. The ScilitBERT model fine-tuned on the Web of Science topic classification task is used to gain insights on the distribution of data in the pretraining corpus. The overall pipeline is described in Fig. 1. The set of 243 topics is mapped to a set of 45 categories to provide an easier understanding of the data. A paper is put in the "UNCERTAIN" category if the maximum and

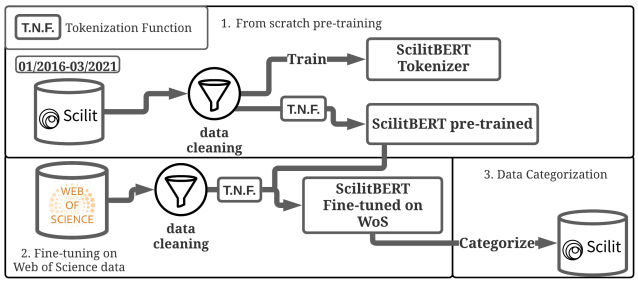

Fig. 1. Pipeline used to categorize the pretraining corpus.

second maximum values for the logit[3] differ by less than the arbitrarily chosen threshold. A simple category distribution of the pretraining corpus data is given in Tab. IV.

VII. Conclusion

This work showed that the computational cost of pretraining an efficient specialized model on a restricted domain is low. This pretraining from scratch of a model has the benefit of providing full control on the model architecture and pretraining data. This approach also allows the use of a domain-specific vocabulary. A specialized model will also make full use of its capabilities, whereas, when using a general model on a narrow domain, some parts of the model are underused.

We release ScilitBERT, which is a BERT model that follows RoBERTa's best practices and is specialized for academic domain tasks. ScilitBERT, with a short pretraining, i.e., 0.1% of RoBERTa's pretraining duration, can compete with highly pretrained language models on academic domain tasks. ScilitBERT is half as deep as RoBERTa and attains at least 98.8% of its competitors accuracy on the proposed academic benchmarks ("Journal-Finder" and "WoS topic classification"). The "Journal-Finder" task dataset is released as a benchmark for academic domain NLP.

We showed that it would be beneficial for any academic subdomain that has access to abundant text data, to invest in the pretraining of language models using in-domain vocabularies. Indeed, many domains use specific language features and vocabularies; for most words in these vocabularies, a subwords' embedding combination is not a decent representation of the said word.

References

[1] W. Nagy and D. Townsend, "Words as tools: Learning academic vocabulary as language acquisition," *Reading Research Quarterly*, vol. 47, no. 1, pp. 91–108, 2012.

[2] L. Bornmann and R. Mutz, "Growth rates of modern science: A bibliometric analysis," *arXiv preprints*, vol. abs/1402.4578, 2014. [Online]. Available: http://arxiv.org/abs/1402.4578

[3] E. Medvet, A. Bartoli, and G. Piccinin, "Publication venue recommendation based on paper abstract," in *IEEE 26th International Conference on Tools with Artificial Intelligence*, 2014, pp. 1004–1010.

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez *et al.*, "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, p. 6000–6010.

[3]The probability distribution given by the model on the set of topics.

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 4171–4186.

[6] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen *et al.*, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprints*, 2019. [Online]. Available: https://arxiv.org/abs/1907.11692

[7] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language Models are Unsupervised Multitask Learners," 2019. [Online]. Available: https://openai.com/blog/better-language-models/

[8] I. Beltagy, K. Lo, and 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*, 2019, pp. 3615–3620.

[9] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Empirical Methods in Natural Language Processing*, 2014, pp. 1532–1543.

[10] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *1st International Conference on Learning Representations, Workshop Track Proceedings*, 2013. [Online]. Available: http://arxiv.org/abs/1301.3781

[11] E. Dusserre and M. Padró, "Bigger does not mean better! we prefer specificity," in *12th International Conference on Computational Semantics — Short papers*, 2017.

[12] A. Roy, Y. Park, and S. Pan, "Incorporating domain knowledge in learning word embedding," in *IEEE 31st International Conference on Tools with Artificial Intelligence*, 2019, pp. 1568–1573.

[13] A. Raganato and J. Tiedemann, "An analysis of encoder representations in transformer-based machine translation," in *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, 2018, pp. 287–297.

[14] Y. Gu, R. Tinn, H. Cheng, M. Lucas, N. Usuyama, X. Liu *et al.*, "Domain-specific language model pretraining for biomedical natural language processing," *arXiv preprints*, 2021. [Online]. Available: https://arxiv.org/abs/2007.15779

[15] S. Gururangan, A. Marasović, S. Swayamdipta, K. Lo, I. Beltagy, D. Downey *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*, 2020, pp. 8342–8360.

[16] X. Liu, D. Yin, X. Zhang, K. Su, K. Wu, H. Yang *et al.*, "OAG-BERT: pre-train heterogeneous entity-augmented academic language models," *arXiv preprints*, 2021. [Online]. Available: https://arxiv.org/abs/2103.02410

[17] C. Wang, K. Cho, and J. Gu, "Neural machine translation with byte-level subwords," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 5, 2020, pp. 9154–9160.

[18] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 2016, pp. 1715–1725.

[19] M. Schuster and K. Nakajima, "Japanese and korean voice search," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2012, pp. 5149–5152.

[20] M. Johnson, M. Schuster, Q. V. Le, M. Krikun, Y. Wu, Z. Chen *et al.*, "Google's multilingual neural machine translation system: Enabling zero-shot translation," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 339–351, 2017.

[21] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman, "GLUE: A multi-task benchmark and analysis platform for natural language understanding," in *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, 2018, pp. 353–355.

[22] P. Rajpurkar, R. Jia, and P. Liang, "Know what you don't know: Unanswerable questions for SQuAD," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2018, pp. 784–789.

[23] L. McInnes, J. Healy, N. Saul, and L. Großberger, "Umap: Uniform manifold approximation and projection," *Journal of Open Source Software*, vol. 3, no. 29, p. 861, 2018.