SciBert: A Pretrained Language Model for Scientific Text

Iz Beltagy **Kyle Lo** **Arman Cohan**
Allen Institute for Artificial Intelligence, Seattle, WA, USA
{beltagy,kylel,armanc}@allenai.org

Abstract

Obtaining large-scale annotated data for NLP tasks in the scientific domain is challenging and expensive. We release SciBert, a pretrained language model based on Bert (Devlin et al., 2019) to address the lack of high-quality, large-scale labeled scientific data. SciBert leverages unsupervised pretraining on a large multi-domain corpus of scientific publications to improve performance on downstream scientific NLP tasks. We evaluate on a suite of tasks including sequence tagging, sentence classification and dependency parsing, with datasets from a variety of scientific domains. We demonstrate statistically significant improvements over Bert and achieve new state-of-the-art results on several of these tasks. The code and pretrained models are available at https://github.com/allenai/scibert/.

1 Introduction

The exponential increase in the volume of scientific publications in the past decades has made NLP an essential tool for large-scale knowledge extraction and machine reading of these documents. Recent progress in NLP has been driven by the adoption of deep neural models, but training such models often requires large amounts of labeled data. In general domains, large-scale training data is often possible to obtain through crowdsourcing, but in scientific domains, annotated data is difficult and expensive to collect due to the expertise required for quality annotation.

As shown through ELMo (Peters et al., 2018), GPT (Radford et al., 2018) and Bert (Devlin et al., 2019), unsupervised pretraining of language models on large corpora significantly improves performance on many NLP tasks. These models return contextualized embeddings for each token which can be passed into minimal task-specific neural architectures. Leveraging the success of unsupervised pretraining has become especially important especially when task-specific annotations are difficult to obtain, like in scientific NLP. Yet while both Bert and ELMo have released pretrained models, they are still trained on general domain corpora such as news articles and Wikipedia.

In this work, we make the following contributions:

(*i*) We release SciBert, a new resource demonstrated to improve performance on a range of NLP tasks in the scientific domain. SciBert is a pretrained language model based on Bert but trained on a large corpus of scientific text.

(*ii*) We perform extensive experimentation to investigate the performance of finetuning versus task-specific architectures atop frozen embeddings, and the effect of having an in-domain vocabulary.

(*iii*) We evaluate SciBert on a suite of tasks in the scientific domain, and achieve new state-of-the-art (SOTA) results on many of these tasks.

2 Methods

Background The Bert model architecture (Devlin et al., 2019) is based on a multilayer bidirectional Transformer (Vaswani et al., 2017). Instead of the traditional left-to-right language modeling objective, Bert is trained on two tasks: predicting randomly masked tokens and predicting whether two sentences follow each other. SciBert follows the same architecture as Bert but is instead pretrained on scientific text.

Vocabulary Bert uses WordPiece (Wu et al., 2016) for unsupervised tokenization of the input text. The vocabulary is built such that it contains the most frequently used words or subword units. We refer to the original vocabulary released with Bert as BaseVocab.

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 3615–3620,
Hong Kong, China, November 3–7, 2019. ©2019 Association for Computational Linguistics

We construct SciVocab, a new WordPiece vocabulary on our scientific corpus using the SentencePiece[1] library. We produce both cased and uncased vocabularies and set the vocabulary size to 30K to match the size of BaseVocab. The resulting token overlap between BaseVocab and SciVocab is 42%, illustrating a substantial difference in frequently used words between scientific and general domain texts.

Corpus We train SciBert on a random sample of 1.14M papers from Semantic Scholar (Ammar et al., 2018). This corpus consists of 18% papers from the computer science domain and 82% from the broad biomedical domain. We use the full text of the papers, not just the abstracts. The average paper length is 154 sentences (2,769 tokens) resulting in a corpus size of 3.17B tokens, similar to the 3.3B tokens on which Bert was trained. We split sentences using ScispaCy (Neumann et al., 2019),[2] which is optimized for scientific text.

3 Experimental Setup

3.1 Tasks

We experiment on the following core NLP tasks:

1. Named Entity Recognition (NER)
2. PICO Extraction (PICO)
3. Text Classification (CLS)
4. Relation Classification (REL)
5. Dependency Parsing (DEP)

PICO, like NER, is a sequence labeling task where the model extracts spans describing the Participants, Interventions, Comparisons, and Outcomes in a clinical trial paper (Kim et al., 2011). REL is a special case of text classification where the model predicts the type of relation expressed between two entities, which are encapsulated in the sentence by inserted special tokens.

3.2 Datasets

For brevity, we only describe the newer datasets here, and refer the reader to the references in Table 1 for the older datasets. EBM-NLP (Nye et al., 2018) annotates PICO spans in clinical trial abstracts. SciERC (Luan et al., 2018) annotates entities and relations from computer science abstracts. ACL-ARC (Jurgens et al., 2018) and SciCite (Cohan et al., 2019) assign intent labels (e.g. Comparison, Extension, etc.) to sentences from scientific papers that cite other papers. The Paper Field dataset is built from the Microsoft Academic Graph (Sinha et al., 2015)[3] and maps paper titles to one of 7 fields of study. Each field of study (i.e. geography, politics, economics, business, sociology, medicine, and psychology) has approximately 12K training examples.

3.3 Pretrained Bert Variants

Bert-Base We use the pretrained weights for Bert-Base (Devlin et al., 2019) released with the original Bert code.[4] The vocabulary is BaseVocab. We evaluate both cased and uncased versions of this model.

SciBert We use the original Bert code to train SciBert on our corpus with the same configuration and size as Bert-Base. We train 4 different versions of SciBert: (*i*) cased or uncased and (*ii*) BaseVocab or SciVocab. The two models that use BaseVocab are finetuned from the corresponding Bert-Base models. The other two models that use the new SciVocab are trained from scratch.

Pretraining Bert for long sentences can be slow. Following the original Bert code, we set a maximum sentence length of 128 tokens, and train the model until the training loss stops decreasing. We then continue training the model allowing sentence lengths up to 512 tokens.

We use a single TPU v3 with 8 cores. Training the SciVocab models from scratch on our corpus takes 1 week[5] (5 days with max length 128, then 2 days with max length 512). The BaseVocab models take 2 fewer days of training because they aren't trained from scratch.

All pretrained Bert models are converted to be compatible with PyTorch using the pytorch-transformers library.[6] All our models (Sections 3.4 and 3.5) are implemented in PyTorch using AllenNLP (Gardner et al., 2017).

[1] https://github.com/google/sentencepiece

[2] https://github.com/allenai/SciSpaCy

[3] https://academic.microsoft.com/

[4] https://github.com/google-research/bert

[5] Bert's largest model was trained on 16 Cloud TPUs for 4 days. Expected 40-70 days (Dettmers, 2019) on an 8-GPU machine.

[6] https://github.com/huggingface/pytorch-transformers

Casing We follow Devlin et al. (2019) in using the cased models for NER and the uncased models for all other tasks. We also use the cased models for parsing. Some light experimentation showed that the uncased models perform slightly better (even sometimes on NER) than cased models.

3.4 Finetuning BERT

We mostly follow the same architecture, optimization, and hyperparameter choices used in Devlin et al. (2019). For text classification (i.e. CLS and REL), we feed the final BERT vector for the `[CLS]` token into a linear classification layer. For sequence labeling (i.e. NER and PICO), we feed the final BERT vector for each token into a linear classification layer with softmax output. We differ slightly in using an additional conditional random field, which made evaluation easier by guaranteeing well-formed entities. For DEP, we use the model from Dozat and Manning (2017) with dependency tag and arc embeddings of size 100 and biaffine matrix attention over BERT vectors instead of stacked BiLSTMs.

In all settings, we apply a dropout of 0.1 and optimize cross entropy loss using Adam (Kingma and Ba, 2015). We finetune for 2 to 5 epochs using a batch size of 32 and a learning rate of 5e-6, 1e-5, 2e-5, or 5e-5 with a slanted triangular schedule (Howard and Ruder, 2018) which is equivalent to the linear warmup followed by linear decay (Devlin et al., 2019). For each dataset and BERT variant, we pick the best learning rate and number of epochs on the development set and report the corresponding test results.

We found the setting that works best across most datasets and models is 2 or 4 epochs and a learning rate of 2e-5. While task-dependent, optimal hyperparameters for each task are often the same across BERT variants.

3.5 Frozen BERT Embeddings

We also explore the usage of BERT as pretrained contextualized word embeddings, like ELMo (Peters et al., 2018), by training simple task-specific models atop frozen BERT embeddings.

For text classification, we feed each sentence of BERT vectors into a 2-layer BiLSTM of size 200 and apply a multilayer perceptron (with hidden size 200) on the concatenated first and last BiLSTM vectors. For sequence labeling, we use the same BiLSTM layers and use a conditional random field to guarantee well-formed predictions. For DEP, we use the full model from Dozat and Manning (2017) with dependency tag and arc embeddings of size 100 and the same BiLSTM setup as other tasks. We did not find changing the depth or size of the BiLSTMs to significantly impact results (Reimers and Gurevych, 2017).

We optimize cross entropy loss using Adam, but holding BERT weights frozen and applying a dropout of 0.5. We train with early stopping on the development set (patience of 10) using a batch size of 32 and a learning rate of 0.001.

We did not perform extensive hyperparameter search, but while optimal hyperparameters are going to be task-dependent, some light experimentation showed these settings work fairly well across most tasks and BERT variants.

4 Results

Table 1 summarizes the experimental results. We observe that SCIBERT outperforms BERT-Base on scientific tasks (+2.11 F1 with finetuning and +2.43 F1 without)[8]. We also achieve new SOTA results on many of these tasks using SCIBERT.

4.1 Biomedical Domain

We observe that SCIBERT outperforms BERT-Base on biomedical tasks (+1.92 F1 with finetuning and +3.59 F1 without). In addition, SCIBERT achieves new SOTA results on BC5CDR and ChemProt (Lee et al., 2019), and EBM-NLP (Nye et al., 2018).

SCIBERT performs slightly worse than SOTA on 3 datasets. The SOTA model for JNLPBA is a BiLSTM-CRF ensemble trained on multiple NER datasets not just JNLPBA (Yoon et al., 2018). The SOTA model for NCBI-disease is BIOBERT (Lee et al., 2019), which is BERT-Base finetuned on 18B tokens from biomedical papers. The SOTA result for GENIA is in Nguyen and Verspoor (2019) which uses the model from Dozat and Manning (2017) with part-of-speech (POS) features, which we do not use.

In Table 2, we compare SCIBERT results with reported BIOBERT results on the subset of datasets included in (Lee et al., 2019). Interesting, SCIBERT outperforms BIOBERT results on

[7] The SOTA paper did not report a single score. We compute the average of the reported results for each class weighted by number of examples in each class.

[8] For rest of this paper, all results reported in this manner are averaged over datasets excluding UAS for DEP since we already include LAS.

Field	Task	Dataset	SOTA	BERT-Base		SciBERT	
				Frozen	Finetune	Frozen	Finetune
Bio	NER	BC5CDR (Li et al., 2016)	88.85[7]	85.08	86.72	88.73	**90.01**
		JNLPBA (Collier and Kim, 2004)	**78.58**	74.05	76.09	75.77	77.28
		NCBI-disease (Dogan et al., 2014)	**89.36**	84.06	86.88	86.39	88.57
	PICO	EBM-NLP (Nye et al., 2018)	66.30	61.44	71.53	68.30	**72.28**
	DEP	GENIA (Kim et al., 2003) - LAS	**91.92**	90.22	90.33	90.36	90.43
		GENIA (Kim et al., 2003) - UAS	**92.84**	91.84	91.89	92.00	91.99
	REL	ChemProt (Kringelum et al., 2016)	76.68	68.21	79.14	75.03	**83.64**
CS	NER	SciERC (Luan et al., 2018)	64.20	63.58	65.24	65.77	**67.57**
	REL	SciERC (Luan et al., 2018)	n/a	72.74	78.71	75.25	**79.97**
	CLS	ACL-ARC (Jurgens et al., 2018)	67.9	62.04	63.91	60.74	**70.98**
Multi	CLS	Paper Field	n/a	63.64	65.37	64.38	**65.71**
		SciCite (Cohan et al., 2019)	84.0	84.31	84.85	**85.42**	**85.49**
Average				73.58	77.16	76.01	79.27

Table 1: Test performances of all BERT variants on all tasks and datasets. **Bold** indicates the SOTA result (multiple results bolded if difference within 95% bootstrap confidence interval). Keeping with past work, we report macro F1 scores for NER (span-level), macro F1 scores for REL and CLS (sentence-level), and macro F1 for PICO (token-level), and micro F1 for ChemProt specifically. For DEP, we report labeled (LAS) and unlabeled (UAS) attachment scores (excluding punctuation) for the same model with hyperparameters tuned for LAS. All results are the average of multiple runs with different random seeds.

Task	Dataset	BioBERT	SciBERT
NER	BC5CDR	88.85	90.01
	JNLPBA	77.59	77.28
	NCBI-disease	89.36	88.57
REL	ChemProt	76.68	83.64

Table 2: Comparing SciBERT with the reported BioBERT results on biomedical datasets.

BC5CDR and ChemProt, and performs similarly on JNLPBA despite being trained on a substantially smaller biomedical corpus.

4.2 Computer Science Domain

We observe that SciBERT outperforms BERT-Base on computer science tasks (+3.55 F1 with finetuning and +1.13 F1 without). In addition, SciBERT achieves new SOTA results on ACL-ARC (Cohan et al., 2019), and the NER part of SciERC (Luan et al., 2018). For relations in SciERC, our results are not comparable with those in Luan et al. (2018) because we are performing relation classification given gold entities, while they perform joint entity and relation extraction.

4.3 Multiple Domains

We observe that SciBERT outperforms BERT-Base on the multidomain tasks (+0.49 F1 with finetuning and +0.93 F1 without). In addition, SciBERT outperforms the SOTA on SciCite (Cohan et al., 2019). No prior published SOTA results exist for the Paper Field dataset.

5 Discussion

5.1 Effect of Finetuning

We observe improved results via BERT finetuning rather than task-specific architectures atop frozen embeddings (+3.25 F1 with SciBERT and +3.58 with BERT-Base, on average). For each scientific domain, we observe the largest effects of finetuning on the computer science (+5.59 F1 with SciBERT and +3.17 F1 with BERT-Base) and biomedical tasks (+2.94 F1 with SciBERT and +4.61 F1 with BERT-Base), and the smallest effect on multidomain tasks (+0.7 F1 with SciBERT and +1.14 F1 with BERT-Base). On every dataset except BC5CDR and SciCite, BERT-Base with finetuning outperforms (or performs similarly to) a model using frozen SciBERT embeddings.

5.2 Effect of SciVocab

We assess the importance of an in-domain scientific vocabulary by repeating the finetuning experiments for SciBERT with BaseVocab. We find the optimal hyperparameters for SciBERT-BaseVocab often coincide with those of SciBERT-SciVocab.

Averaged across datasets, we observe +0.60 F1 when using SciVocab. For each scientific domain, we observe +0.76 F1 for biomedical tasks, +0.61 F1 for computer science tasks, and +0.11 F1 for multidomain tasks.

Given the disjoint vocabularies (Section 2) and the magnitude of improvement over BERT-Base (Section 4), we suspect that while an in-domain vocabulary is helpful, SCIBERT benefits most from the scientific corpus pretraining.

6 Related Work

Recent work on domain adaptation of BERT includes BIOBERT (Lee et al., 2019) and CLINICALBERT (Alsentzer et al., 2019; Huang et al., 2019). BIOBERT is trained on PubMed abstracts and PMC full text articles, and CLINICALBERT is trained on clinical text from the MIMIC-III database (Johnson et al., 2016). In contrast, SCIBERT is trained on the full text of 1.14M biomedical and computer science papers from the Semantic Scholar corpus (Ammar et al., 2018). Furthermore, SCIBERT uses an in-domain vocabulary (SCIVOCAB) while the other above-mentioned models use the original BERT vocabulary (BASEVOCAB).

7 Conclusion and Future Work

We released SCIBERT, a pretrained language model for scientific text based on BERT. We evaluated SCIBERT on a suite of tasks and datasets from scientific domains. SCIBERT significantly outperformed BERT-Base and achieves new SOTA results on several of these tasks, even compared to some reported BIOBERT (Lee et al., 2019) results on biomedical tasks.

For future work, we will release a version of SCIBERT analogous to BERT-Large, as well as experiment with different proportions of papers from each domain. Because these language models are costly to train, we aim to build a single resource that's useful across multiple domains.

Acknowledgment

We thank the anonymous reviewers for their comments and suggestions. We also thank Waleed Ammar, Noah Smith, Yoav Goldberg, Daniel King, Doug Downey, and Dan Weld for their helpful discussions and feedback. All experiments were performed on beaker.org and supported in part by credits from Google Cloud.

References

Emily Alsentzer, John R. Murphy, Willie Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, and Matthew B. A. McDermott. 2019. Publicly available clinical bert embeddings. In *ClinicalNLP workshop at NAACL*.

Waleed Ammar, Dirk Groeneveld, Chandra Bhagavatula, Iz Beltagy, Miles Crawford, Doug Downey, Jason Dunkelberger, Ahmed Elgohary, Sergey Feldman, Vu Ha, Rodney Kinney, Sebastian Kohlmeier, Kyle Lo, Tyler Murray, Hsu-Han Ooi, Matthew Peters, Joanna Power, Sam Skjonsberg, Lucy Lu Wang, Chris Wilhelm, Zheng Yuan, Madeleine van Zuylen, and Oren Etzioni. 2018. Construction of the literature graph in semantic scholar. In *NAACL*.

Arman Cohan, Waleed Ammar, Madeleine van Zuylen, and Field Cady. 2019. Structural scaffolds for citation intent classification in scientific publications. In *NAACL-HLT*, pages 3586–3596, Minneapolis, Minnesota. Association for Computational Linguistics.

Nigel Collier and Jin-Dong Kim. 2004. Introduction to the bio-entity recognition task at jnlpba. In *NLP-BA/BioNLP*.

Tim Dettmers. 2019. TPUs vs GPUs for Transformers (BERT). `http://timdettmers.com/2018/10/17/tpus-vs-gpus-for-transformers-bert/`. Accessed: 2019-02-22.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Rezarta Islamaj Dogan, Robert Leaman, and Zhiyong Lu. 2014. NCBI disease corpus: A resource for disease name recognition and concept normalization. *Journal of biomedical informatics*, 47:1–10.

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. *ICLR*.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke S. Zettlemoyer. 2017. Allennlp: A deep semantic natural language processing platform. In *arXiv:1803.07640*.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *ACL*.

Kexin Huang, Jaan Altosaar, and Rajesh Ranganath. 2019. Clinicalbert: Modeling clinical notes and predicting hospital readmission. *arXiv:1904.05342*.

Alistair E. W. Johnson, Tom J. Pollard aand Lu Shen, Liwei H. Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, , and Roger G. Mark. 2016.

Mimic-iii, a freely accessible critical care database. In *Scientific Data, 3:160035*.

David Jurgens, Srijan Kumar, Raine Hoover, Daniel A. McFarland, and Daniel Jurafsky. 2018. Measuring the evolution of a scientific field through citation frames. *TACL*, 06:391–406.

Jin-Dong Kim, Tomoko Ohta, Yuka Tateisi, and Jun'ichi Tsujii. 2003. GENIA corpus - a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19:i180i182.

Su Kim, David Martínez, Lawrence Cavedon, and Lars Yencken. 2011. Automatic classification of sentences to support evidence based medicine. In *BMC Bioinformatics*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *ICLR*.

Jens Kringelum, Sonny Kim Kjærulff, Søren Brunak, Ole Lund, Tudor I. Oprea, and Olivier Taboureau. 2016. ChemProt-3.0: a global chemical biology diseases mapping. In *Database*.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. In *arXiv:1901.08746*.

Jiao Li, Yueping Sun, Robin J. Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J. Mattingly, Thomas C. Wiegers, and Zhiyong Lu. 2016. BioCreative V CDR task corpus: a resource for chemical disease relation extraction. *Database : the journal of biological databases and curation*.

Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In *EMNLP*.

Mark Neumann, Daniel King, Iz Beltagy, and Waleed Ammar. 2019. ScispaCy: Fast and robust models for biomedical natural language processing. In *arXiv:1902.07669*.

Dat Quoc Nguyen and Karin M. Verspoor. 2019. From pos tagging to dependency parsing for biomedical event extraction. *BMC Bioinformatics*, 20:1–13.

Benjamin Nye, Junyi Jessy Li, Roma Patel, Yinfei Yang, Iain James Marshall, Ani Nenkova, and Byron C. Wallace. 2018. A corpus with multi-level annotations of patients, interventions and outcomes to support language processing for medical literature. In *ACL*.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke S. Zettlemoyer. 2018. Deep contextualized word representations. In *NAACL-HLT*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Nils Reimers and Iryna Gurevych. 2017. Optimal hyperparameters for deep lstm-networks for sequence labeling tasks. In *EMNLP*.

Arnab Sinha, Zhihong Shen, Yang Song, Hao Ma, Darrin Eide, Bo-June Paul Hsu, and Kuansan Wang. 2015. An overview of microsoft academic service (MAS) and applications. In *WWW*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Gregory S. Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. abs/1609.08144.

Wonjin Yoon, Chan Ho So, Jinhyuk Lee, and Jaewoo Kang. 2018. CollaboNet: collaboration of deep neural networks for biomedical named entity recognition. In *DTMBio workshop at CIKM*.