Domain-Specific Language Model Pretraining for Biomedical Natural Language Processing

YU GU, ROBERT TINN, HAO CHENG, MICHAEL LUCAS, NAOTO USUYAMA, XIAODONG LIU, TRISTAN NAUMANN, JIANFENG GAO, and HOIFUNG POON, Microsoft Research

Pretraining large neural language models, such as BERT, has led to impressive gains on many natural language processing (NLP) tasks. However, most pretraining efforts focus on general domain corpora, such as newswire and Web. A prevailing assumption is that even domain-specific pretraining can benefit by starting from general-domain language models. In this article, we challenge this assumption by showing that for domains with abundant unlabeled text, such as biomedicine, pretraining language models from scratch results in substantial gains over continual pretraining of general-domain language models. To facilitate this investigation, we compile a comprehensive biomedical NLP benchmark from publicly available datasets. Our experiments show that domain-specific pretraining serves as a solid foundation for a wide range of biomedical NLP tasks, leading to new state-of-the-art results across the board. Further, in conducting a thorough evaluation of modeling choices, both for pretraining and task-specific fine-tuning, we discover that some common practices are unnecessary with BERT models, such as using complex tagging schemes in named entity recognition. To help accelerate research in biomedical NLP, we have released our state-of-the-art pretrained and task-specific models for the community, and created a leaderboard featuring our BLURB benchmark (short for Biomedical Language Understanding & Reasoning Benchmark) at https://aka.ms/BLURB.

1 INTRODUCTION

In **natural language processing (NLP)**, pretraining large neural language models on unlabeled text has proven to be a successful strategy for transfer learning. A prime example is **Bidirectional Encoder Representations from Transformers (BERT)** [16], which has become a standard building block for training task-specific NLP models. Existing pretraining work typically focuses on the newswire and Web domains. For example, the original
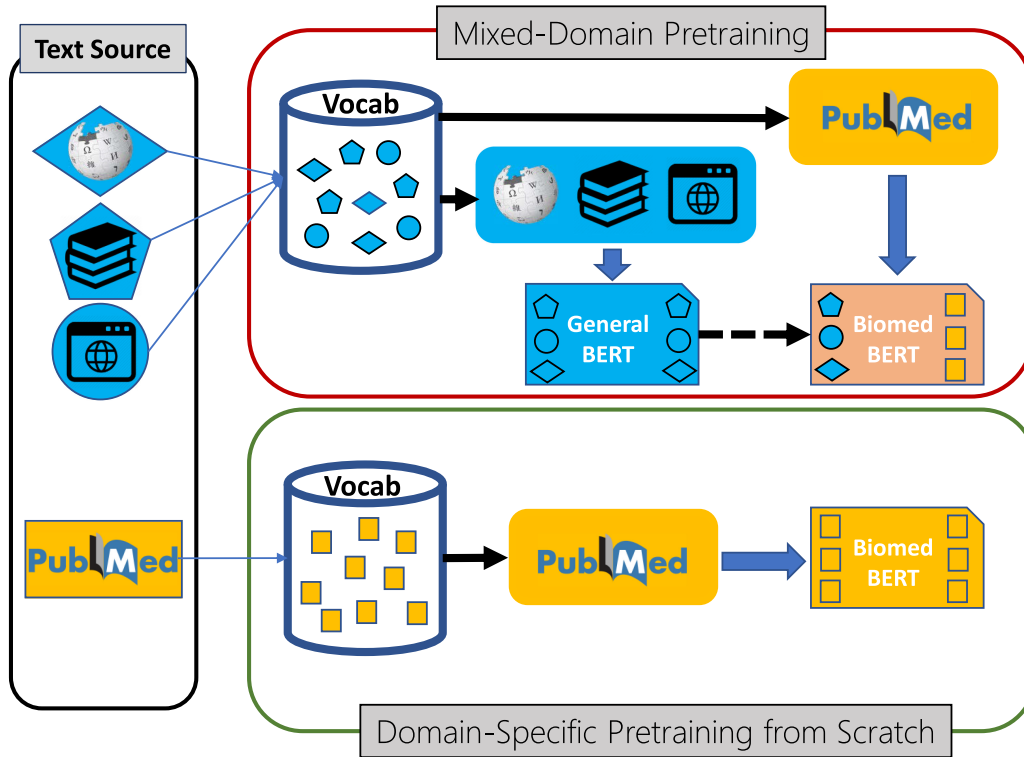
Fig. 1. Two paradigms for neural language model pretraining. Top: The prevailing mixed-domain paradigm assumes that out-domain text is still helpful and typically initializes domain-specific pretraining with a general-domain language model and inherits its vocabulary. Bottom: Domain-specific pretraining from scratch derives the vocabulary and conducts pretraining using solely in-domain text. In this article, we show that for domains with abundant text such as biomedicine, domain-specific pretraining from scratch can substantially outperform the conventional mixed-domain approach.

BERT model was trained on Wikipedia[1] and BookCorpus [62], and subsequent efforts have focused on crawling additional text from the Web to power even larger-scale pretraining [39, 50].

In specialized domains like biomedicine, past work has shown that using in-domain text can provide additional gains over general-domain language models [8, 34, 45]. However, a prevailing assumption is that out-domain text is still helpful and previous work typically adopts a mixed-domain approach, such as by starting domain-specific pretraining from an existing general-domain language model (Figure 1, top). In this article, we question this assumption. We observe that mixed-domain pretraining such as continual pretraining can be viewed as a form of transfer learning in itself, where the source domain is general text, such as newswire and the Web, and the target domain is specialized text, such as biomedical papers. Based on the rich literature of multi-task learning and transfer learning [4, 13, 38, 59], successful transfer learning occurs when the target data is scarce and the source domain is highly relevant to the target one. For domains with abundant unlabeled text such as biomedicine, it is unclear that domain-specific pretraining can benefit by transfer from general domains. In fact, the majority of general domain text is substantively different from biomedical text, raising the prospect of negative transfer that actually hinders the target performance.

---

[1]http://wikipedia.org.

We thus set out to conduct a rigorous study on domain-specific pretraining and its impact on downstream applications, using biomedicine as a running example. *We show that domain-specific pretraining from scratch substantially outperforms continual pretraining of generic language models, thus demonstrating that the prevailing assumption in support of mixed-domain pretraining is not always applicable* (Figure 1).

To facilitate this study, we compile a comprehensive biomedical NLP benchmark from publicly-available datasets, and conduct in-depth comparisons of modeling choices for pretraining and task-specific fine-tuning by their impact on domain-specific applications. Our experiments show that domain-specific pretraining from scratch can provide a solid foundation for biomedical NLP, leading to new state-of-the-art performance across a wide range of tasks. Additionally, we discover that the use of transformer-based models, like BERT, necessitates rethinking several common practices. For example, BIO tags and more complex variants are the standard label representation for **Named Entity Recognition (NER)**. However, we find that simply using IO (in or out of entity mentions) suffices with BERT models, leading to comparable or better performance.

To help accelerate research in biomedical NLP, we have released our state-of-the-art pretrained and task-specific models for the community, and created a leaderboard featuring our comprehensive benchmark at https://aka.ms/BLURB.

2 METHODS

2.1 Language Model Pretraining

In this section, we provide a brief overview of neural language model pretraining, using BERT [16] as a running example.

2.1.1 Vocabulary. We assume that the input consists of text spans, such as sentences separated by special tokens [SEP]. To address the problem of out-of-vocabulary words, neural language models generate a vocabulary from subword units, using **Byte-Pair Encoding (BPE)** [51] or variants such as WordPiece [32]. Essentially, the BPE algorithm tries to greedily identify a small set of subwords that can compactly form all words in the given corpus. It does this by first shattering all words in the corpus and initializing the vocabulary with characters and delimiters. It then iteratively augments the vocabulary with a new subword that is most frequent in the corpus and can be formed by concatenating two existing subwords, until the vocabulary reaches the pre-specified size (e.g., 30,000 in standard BERT models or 50,000 in RoBERTa [39]). In this article, we use the WordPiece algorithm, which is a BPE variant that uses likelihood based on the unigram language model rather than frequency in choosing which subwords to concatenate. The text corpus and vocabulary may preserve the case (cased) or convert all characters to lowercase (uncased).

2.1.2 Model Architecture. State-of-the-art neural language models are generally based on transformer architectures [55], following the recent success of BERT [16, 39]. The transformer model introduces a multi-layer, multi-head self-attention mechanism, which has demonstrated superiority in leveraging GPU-based parallel computation and modeling long-range dependencies in texts, compared to recurrent neural networks, such as **Long Short-Term Memory (LSTM)** [22]. The input token sequence is first processed by a lexical encoder, which combines a token embedding, a (token) position embedding and a segment embedding (i.e., which text span the token belongs to) by element-wise summation. This embedding layer is then passed to multiple layers of transformer modules [55]. In each transformer layer, a contextual representation is generated for each token by summing a non-linear transformation of the representations of all tokens in the prior layer, weighted by the attentions computed using the given token's representation in the prior layer as the query. The final layer outputs contextual representations for all tokens, which combine information from the whole text span.

2.1.3 Self-Supervision. A key innovation in BERT [16] is the use of a **Masked Language Model (MLM)** for self-supervised pretraining. Traditional language models are typically generative models that predict the next token based on the preceding tokens—for example, n-gram models represent the conditional probability

of the next token by a multinomial of the preceding n-gram, with various smoothing strategies to handle rare occurrences [43]. The MLM instead randomly replaces a subset of tokens by a special token (e.g., [MASK]) and asks the language model to predict them. The training objective is the cross-entropy loss between the original tokens and the predicted ones. In BERT and RoBERTa, 15% of the input tokens are chosen, among which a random 80% are replaced by [MASK], 10% are left unchanged, and 10% are randomly replaced by a token from the vocabulary. Instead of using a constant masking rate of 15%, a standard approach is to gradually increase it from 5% to 25% with 5% increment for every 20% of training epochs, which makes pretraining more stable [37]. The original BERT algorithm also uses **Next Sentence Prediction (NSP)**, which determines for a given sentence pair whether one sentence follows the other in the original text. The utility of NSP has been called into question [39], but we include it in our pretraining experiments to enable a head-to-head comparison with prior BERT models.

2.1.4 Advanced Pretraining Techniques. In the original formulation of BERT [16], the MLM simply selects random subwords to mask. When a word is only partially masked, it is relatively easy to predict the masked portion given the observed ones. In contrast, **Whole-Word Masking (WWM)** enforces that the whole word must be masked if one of its subwords is chosen. This has been adopted as the standard approach because it forces the language model to capture more contextual semantic dependencies.

In this article, we also explore adversarial pretraining and its impact on downstream applications. Motivated by successes in countering adversarial attacks in computer vision, adversarial pretraining introduces perturbations in the input embedding layer that maximize the adversarial loss, thus forcing the model to not only optimize the standard training objective (MLM) but also minimize adversarial loss [37].

2.2 Biomedical Language Model Pretraining

In this article, we will use biomedicine as a running example in our study of domain-specific pretraining. In other words, biomedical text is considered in-domain, whereas others are regarded as out-domain. Intuitively, using in-domain text in pretraining should help with domain-specific applications. Indeed, prior work has shown that pretraining with PubMed text leads to better performance in biomedical NLP tasks [8, 34, 45]. The main question is whether pretraining should include text from other domains. The prevailing assumption is that pretraining can always benefit from more text, including out-domain text. In fact, none of the prior biomedical-related BERT models have been pretrained using purely biomedical text [8, 34, 45]. Here, we challenge this assumption and show that *domain-specific pretraining from scratch* can be superior to *mixed-domain pretraining* for downstream applications.

2.2.1 Mixed-Domain Pretraining. The standard approach to pretraining a biomedical BERT model conducts *continual pretraining* of a general-domain pretrained model, as exemplified by BioBERT [34]. Specifically, this approach would initialize with the standard BERT model [16], pretrained using Wikipedia and BookCorpus. It then continues the pretraining process with MLM and NSP using biomedical text. In the case of BioBERT, continual pretraining is conducted using PubMed abstracts and **PubMed Central (PMC)** full-text articles. BlueBERT [45] uses both PubMed text and de-identified clinical notes from MIMIC-III [26].

Note that in the continual pretraining approach, the vocabulary is the same as the original BERT model, in this case the one generated from Wikipedia and BookCorpus. Although convenient, this is a major disadvantage for this approach, as the vocabulary is not representative of the target biomedical domain.

Compared to the other biomedical-related pretraining efforts, SciBERT [8] is a notable exception as it generates the vocabulary and pretrains from scratch, using biomedicine and computer science as representatives for scientific literature. However, from the perspective of biomedical applications, SciBERT still adopts the mixed-domain pretraining approach, as computer science text is clearly out-domain.

2.2.2 Domain-Specific Pretraining from Scratch. The mixed-domain pretraining approach makes sense if the target application domain has little text of its own and can thereby benefit from pretraining using related domains.

Table 1. Comparison of Common Biomedical Terms in Vocabularies Used by the Standard BERT, SciBERT, and PubMedBERT (Ours)

Biomedical Term	Category	BERT	SciBERT	PubMedBERT (Ours)
diabetes	disease	✓	✓	✓
leukemia	disease	✓	✓	✓
lithium	drug	✓	✓	✓
insulin	drug	✓	✓	✓
DNA	gene	✓	✓	✓
promoter	gene	✓	✓	✓
hypertension	disease	hyper-tension	✓	✓
nephropathy	disease	ne-ph-rop-athy	✓	✓
lymphoma	disease	l-ym-ph-oma	✓	✓
lidocaine	drug	lid-oca-ine]	✓	✓
oropharyngeal	organ	oro-pha-ryn-ge-al	or-opharyngeal	✓
cardiomyocyte	cell	card-iom-yo-cy-te	cardiomy-ocyte	✓
chloramphenicol	drug	ch-lor-amp-hen-ico-l	chlor-amp-hen-icol	✓
RecA	gene	Rec-A	Rec-A	✓
acetyltransferase	gene	ace-ty-lt-ran-sf-eras-e	acetyl-transferase	✓
clonidine	drug	cl-oni-dine	clon-idine	✓
naloxone	drug	na-lo-xon-e	nal-oxo-ne	✓

A ✓ indicates the biomedical term appears in the corresponding vocabulary; otherwise, the term will be broken into word pieces separated by a hyphen. These word pieces often have no biomedical relevance and may hinder learning in downstream tasks.

However, this is not the case for biomedicine, which has more than 30 million abstracts in PubMed, and adds more than a million each year. We thus hypothesize that domain-specific pretraining from scratch is a better strategy for biomedical language model pretraining.

A major advantage of domain-specific pretraining from scratch stems from having an in-domain vocabulary. Table 1 compares the vocabularies used in various pretraining strategies. BERT models using continual pretraining are stuck with the original vocabulary from the general-domain corpora, which does not contain many common biomedical terms. Even for SciBERT, which generates its vocabulary partially from biomedical text, the deficiency compared to a purely biomedical vocabulary is substantial. As a result, standard BERT models are forced to divert parameterization capacity and training bandwidth to model biomedical terms using fragmented subwords. For example, *naloxone*, a common medical term, is divided into four pieces ([na, ##lo, ##xon, ##e]) by BERT, and *acetyltransferase* is shattered into seven pieces ([ace, ##ty, ##lt, ##ran, ##sf, ##eras, ##e]) by BERT.[2] Both terms appear in the vocabulary of PubMedBERT.

Another advantage of domain-specific pretraining from scratch is that the language model is trained using purely in-domain data. For example, SciBERT pretraining has to balance optimizing for biomedical text and computer science text, the latter of which is unlikely to be beneficial for biomedical applications. Continual pretraining, however, may potentially recover from out-domain modeling, although not completely. Aside from the vocabulary issue mentioned earlier, neural network training uses non-convex optimization, which means that continual pretraining may not be able to completely undo suboptimal initialization from the general-domain language model.

---

[2]Prior work also observed similar shattering for clinical words [52].

Table 2. Comparison of the Biomedical Datasets in Prior Language Model Pretraining Studies and BLURB

	BioBERT [34]	SciBERT [8]	BLUE [45]	BLURB
BC5-chem [35]	✓	✓	✓	✓
BC5-disease [35]	✓	✓	✓	✓
NCBI-disease [18]	✓	✓	—	✓
BC2GM [53]	✓	—	—	✓
JNLPBA [27]	✓	—	—	✓
EBM PICO [44]	—	✓	—	✓
ChemProt [31]	✓	✓	✓	✓
DDI [21]	✓	—	✓	✓
GAD [11]	✓	—	—	✓
BIOSSES [54]	—	—	✓	✓
HoC [20]	—	—	✓	✓
PubMedQA [25]	—	—	—	✓
BioASQ [42]	✓	—	—	✓

In our experiments, we show that domain-specific pretraining with in-domain vocabulary confers clear advantages over mixed-domain pretraining, be it continual pretraining of general-domain language models or pretraining on mixed-domain text.

2.3 BLURB: A Comprehensive Benchmark for Biomedical NLP

The ultimate goal of language model pretraining is to improve performance on a wide range of downstream applications. In general-domain NLP, the creation of comprehensive benchmarks, such as GLUE [56, 57], greatly accelerates advances in language model pretraining by enabling head-to-head comparisons among pretrained language models. In contrast, prior work on biomedical pretraining tends to use different tasks and datasets for downstream evaluation, as shown in Table 2. This makes it hard to assess the impact of pretrained language models on the downstream tasks we care about. To the best of our knowledge, BLUE [45] is the first attempt to create an NLP benchmark in the biomedical domain. We aim to improve on its design by addressing some of its limitations. First, BLUE has limited coverage of biomedical applications used in other recent work on biomedical language models, as shown in Table 2. For example, it does not include any question-answering task. More importantly, BLUE mixes PubMed-based biomedical applications (six datasets such as BC5, ChemProt, and HoC) with MIMIC-based clinical applications (four datasets such as i2b2 and MedNLI). Clinical notes differ substantially from biomedical literature, to the extent that we observe BERT models pretrained on clinical notes perform poorly on biomedical tasks, similar to the standard BERT. Consequently, it is advantageous to create separate benchmarks for these two domains.

To facilitate investigations of biomedical language model pretraining and help accelerate progress in biomedical NLP, we create a new benchmark, **BLURB (Biomedical Language Understanding & Reasoning Benchmark)**. We focus on PubMed-based biomedical applications and leave the exploration of the clinical domain, and other high-value verticals, to future work. To make our effort tractable and facilitate head-to-head comparison with prior work, we prioritize the selection of datasets used in recent work on biomedical language models and will explore the addition of other datasets in future work.

BLURB is comprised of a comprehensive set of biomedical NLP tasks from publicly available datasets, including NER, evidence-based medical information extraction (PICO), relation extraction, sentence similarity, document classification, and question answering. Table 3 provides an overview of the BLURB datasets. For question answering, prior work has considered both classification tasks (e.g., whether a reference text contains the answer

Table 3. Datasets Used in the BLURB Biomedical NLP Benchmark

Dataset	Task	Train	Dev	Test	Evaluation Metrics
BC5-chem	NER	5,203	5,347	5,385	F1 entity-level
BC5-disease	NER	4,182	4,244	4,424	F1 entity-level
NCBI-disease	NER	5,134	787	960	F1 entity-level
BC2GM	NER	15,197	3,061	6,325	F1 entity-level
JNLPBA	NER	46,750	4,551	8,662	F1 entity-level
EBM PICO	PICO	339,167	85,321	16,364	Macro F1 word-level
ChemProt	Relation Extraction	18,035	11,268	15,745	Micro F1
DDI	Relation Extraction	25,296	2,496	5,716	Micro F1
GAD	Relation Extraction	4,261	535	534	Micro F1
BIOSSES	Sentence Similarity	64	16	20	Pearson
HoC	Document Classification	1,295	186	371	Micro F1
PubMedQA	Question Answering	450	50	500	Accuracy
BioASQ	Question Answering	670	75	140	Accuracy

Note: We list the numbers of instances in train, dev, and test (e.g., entity mentions in NER and PICO elements in evidence-based medical information extraction).

to a given question) and more complex tasks such as list and summary [42]. The latter types often require additional engineering efforts that are not relevant to evaluating neural language models. For simplicity, we focus on the classification tasks such as yes/no question answering in BLURB and leave the inclusion of more complex question answering to future work.

To compute a summary score for BLURB, the simplest way is to report the average score among all tasks. However, this may place undue emphasis on simpler tasks such as NER for which there are many existing datasets. Therefore, we group the datasets by their task types, compute the average score for each task type, and report the macro average among the task types. To help accelerate research in biomedical NLP, we release the BLURB benchmark as well as a leaderboard at http://aka.ms/BLURB.

In the following are detailed descriptions for each task and corresponding datasets.

2.3.1 Named Entity Recognition.

BC5-Chemical & BC5-Disease. The BioCreative V Chemical-Disease Relation corpus [35] was created for evaluating relation extraction of drug-disease interactions, but it is frequently used as a NER corpus for detecting chemical (drug) and disease entities. The dataset consists of 1,500 PubMed abstracts broken into three even splits for training, development, and test. We use a pre-processed version of this dataset generated by Crichton et al. [14], discard the relation labels, and train NER models for chemical (*BC5-Chemical*) and disease (*BC5-Disease*) separately.

NCBI-Disease. The Natural Center for Biotechnology Information Disease corpus [18] contains 793 PubMed abstracts with 6,892 annotated disease mentions linked to 790 distinct disease entities. We use a pre-processed set of train, development, and test splits generated by Crichton et al. [14].

BC2GM. The Biocreative II Gene Mention corpus [53] consists of sentences from PubMed abstracts with manually labeled gene and alternative gene entities. Following prior work, we focus on the gene entity annotation. In its original form, BC2GM contains 15,000 train and 5,000 test sentences. We use a pre-processed version of the dataset generated by Crichton et al. [14], which carves out 2,500 sentences from the training data for development.

JNLPBA. The Joint Workshop on Natural Language Processing in Biomedicine and Its Applications shared task [27] is a NER corpus on PubMed abstracts. The entity types are chosen for molecular biology applications: protein, DNA, RNA, cell line, and cell type. Some of the entity type distinctions are not very meaningful. For example, a gene mention often refers to both the DNA and gene products such as the RNA and protein. Following prior work that evaluates on this dataset [34], we ignore the type distinction and focus on detecting the entity mentions. We use the same train, development, and test splits as in Crichton et al. [14].

2.3.2 Evidence-Based Medical Information Extraction (PICO).

EBM PICO. The Evidence-Based Medicine corpus [44] contains PubMed abstracts on clinical trials, where each abstract is annotated with P, I, and O in PICO: Participants (e.g., `diabetic patients`), Intervention (e.g., `insulin`), Comparator (e.g., `placebo`), and Outcome (e.g., `blood glucose levels`). Comparator (C) labels are omitted as they are standard in clinical trials: placebo for passive control and standard of care for active control. There are 4,300, 500, and 200 abstracts in training, development, and test, respectively. The training and development sets were labeled by Amazon Mechanical Turkers, whereas the test set was labeled by Upwork contributors with prior medical training. EBM PICO provides labels at the word level for each PIO element. For each of the PIO elements in an abstract, we tally the F1 score at the word level, then compute the final score as the average among PIO elements in the dataset. Occasionally, two PICO elements might overlap with each other (e.g., a participant span might contain within it an intervention span). In EBM PICO, about 3% of the PIO words are in the overlap. Note that the dataset released along with SciBERT appears to remove the overlapping words from the larger span (e.g., the participant span as mentioned earlier). We instead use the original dataset [44] and their scripts for preprocessing and evaluation.

2.3.3 Relation Extraction.

ChemProt. The Chemical Protein Interaction corpus [31] consists of PubMed abstracts annotated with chemical-protein interactions between chemical and protein entities. There are 23 interactions organized in a hierarchy, with 10 high-level interactions (including NONE). Most relation instances in ChemProt are within single sentences. Following prior work [8, 34], we only consider sentence-level instances. We follow the ChemProt authors' suggestions and focus on classifying five high-level interactions—UPREGULATOR (CPR : 3), DOWNREGULATOR (CPR : 4), AGONIST (CPR : 5), ANTAGONIST (CPR : 6), SUBSTRATE (CPR : 9)—as well as everything else (`false`). The ChemProt annotation is not exhaustive for all chemical-protein pairs. Following previous work [34, 45], we expand the training and development sets by assigning a `false` label for all chemical-protein pairs that occur in a training or development sentence, but we do not have an explicit label in the ChemProt corpus. Note that prior work uses slightly different label expansion of the test data. To facilitate head-to-head comparison, we will provide instructions for reproducing the test set in BLURB from the original dataset.

DDI. The Drug-Drug Interaction corpus [21] was created to facilitate research on pharmaceutical information extraction, with a particular focus on pharmacovigilance. It contains sentence-level annotation of drug-drug interactions on PubMed abstracts. Note that some prior work [45, 61] discarded 90 training files that the authors considered not conducive to learning drug-drug interactions. We instead use the original dataset and produce our train/dev/test split of 624/90/191 files.

GAD. The Genetic Association Database corpus [11] was created semi-automatically using the Genetic Association Archive.[3] Specifically, the archive contains a list of gene-disease associations, with the corresponding sentences in the PubMed abstracts reporting the association studies. Bravo et al. [11] used a biomedical NER tool to identify gene and disease mentions, and create the positive examples from the annotated sentences in the

---

[3]http://geneticassociationdb.nih.gov/.

archive, and negative examples from gene-disease co-occurrences that were not annotated in the archive. We use an existing preprocessed version of GAD and its corresponding train/dev/test split created by Lee et al. [34].

2.3.4 Sentence Similarity.

BIOSSES. The Sentence Similarity Estimation System for the Biomedical Domain [54] contains 100 pairs of PubMed sentences each of which is annotated by five expert-level annotators with an estimated similarity score in the range from 0 (no relation) to 4 (equivalent meanings). It is a regression task, with the average score as the final annotation. We use the same train/dev/test split as that in the work of Peng et al. [45] and use Pearson correlation for evaluation.

2.3.5 Document Classification.

HoC. The Hallmarks of Cancer corpus was motivated by the pioneering work on cancer hallmarks [20]. It contains annotation on PubMed abstracts with binary labels, each of which signifies the discussion of a specific cancer hallmark. The authors use 37 fine-grained hallmarks that are grouped into 10 top-level ones. We focus on predicting the top-level labels. The dataset was released with 1,499 PubMed abstracts [6] and has since been expanded to 1,852 abstracts [5]. Note that Peng et al. [45] discarded a control subset of 272 abstracts that do not discuss any cancer hallmark (i.e., all binary labels are false). We instead adopt the original dataset and report micro F1 across the 10 cancer hallmarks. Although the original dataset provided sentence-level annotation, we follow the common practice and evaluate on the abstract level [19, 60]. We create the train/dev/test split, as they were not available previously.[4]

2.3.6 Question Answering (QA).

PubMedQA. The PubMedQA dataset [25] contains a set of research questions, each with a reference text from a PubMed abstract as well as an annotated label of whether the text contains the answer to the research question (yes/maybe/no). We use the original train/dev/test split with 450, 50, and 500 questions, respectively.

BioASQ. The BioASQ corpus [42] contains multiple question answering tasks annotated by biomedical experts, including yes/no, factoid, list, and summary questions. Pertaining to our objective of comparing neural language models, we focus on the yes/no questions (Task 7b) and leave the inclusion of other tasks to future work. Each question is paired with a reference text containing multiple sentences from a PubMed abstract and a yes/no answer. We use the official train/dev/test split of 670/75/140 questions.

2.4 Task-Specific Fine-Tuning

Pretrained neural language models provide a unifying foundation for learning task-specific models. Given an input token sequence, the language model produces a sequence of vectors in the contextual representation. A task-specific prediction model is then layered on top to generate the final output for a task-specific application. Given task-specific training data, we can learn the task-specific model parameters and refine the BERT model parameters by gradient descent using backpropagation.

Prior work on biomedical NLP often adopts different task-specific models and fine-tuning methods, which makes it difficult to understand the impact of an underlying pretrained language model on task performance. In this section, we review standard methods and common variants used for each task. In our primary investigation comparing pretraining strategies, we fix the task-specific model architecture using the standard method identifed here, to facilitate a head-to-head comparison among the pretrained neural language models. Subsequently, we start with the same pretrained BERT model and conduct additional investigation on the impact for the

---

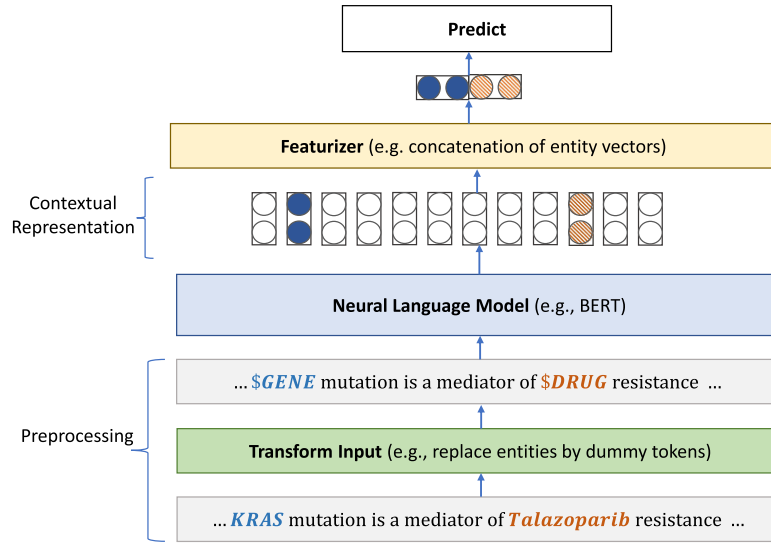[4]The original authors used cross validation for their evaluation.

Fig. 2. A general architecture for task-specific fine-tuning of neural language models, with a relation-extraction example. Note that the input goes through additional processing such as word-piece tokenization in the neural language model module.

various choices in the task-specific models. For prior biomedical BERT models, our standard task-specific methods generally lead to comparable or better performance when compared to their published results.

2.4.1 A General Architecture for Fine-Tuning Neural Language Models. Figure 2 shows a general architecture of fine-tuning neural language models for downstream applications. An input instance is first processed by a TransformInput module that performs task-specific transformations such as appending special instance marker (e.g., [CLS]) or dummifying entity mentions for relation extraction. The transformed input is then tokenized using the neural language model's vocabulary and fed into the neural language model. Next, the contextual representation at the top layer is processed by a Featurizer module and then fed into the Predict module to generate the final output for a given task.

To facilitate a head-to-head comparison, we apply the same fine-tuning procedure for all BERT models and tasks. Specifically, we use cross-entropy loss for classification tasks and mean square error for regression tasks. We conduct hyperparameter search using the development set based on task-specific metrics. Similar to previous work, we jointly fine-tune the parameters of the task-specific prediction layer as well as the underlying neural language model.

2.4.2 Task-Specific Problem Formulation and Modeling Choices. Many NLP applications can be formulated as a classification or regression task, wherein either individual tokens or sequences are the prediction target. Modeling choices usually vary in two aspects: the instance representation and the prediction layer. Table 4 presents an overview of the problem formulation and modeling choices for tasks we consider, and detailed descriptions are provided in the following. For each task, we highlight the standard modeling choices with an asterisk (*).

NER. Given an input text span (usually a sentence), the NER task seeks to recognize mentions of entities of interest. It is typically formulated as a sequential labeling task, where each token is assigned a tag to signify whether it is in an entity mention or not. The modeling choices primarily vary on the tagging scheme and classification method. BIO is the standard tagging scheme that classifies each token as the beginning of an entity (B),

Table 4. Standard NLP Tasks and Their Problem Formulations and Modeling Choices

Task	Problem Formulation	Modeling Choices
NER	Token Classification	Tagging Scheme, Classification Layer
PICO	Token Classification	Tagging Scheme, Classification Layer
Relation Extraction	Sequence Classification	Entity/Relation Representation, Classification Layer
Sentence Similarity	Sequence Regression	Sentence Representation, Regression Loss
Document Classification	Sequence Classification	Document Representation, Classification Layer
Question Answering	Sequence Classification	Question/Text Representation, Classification Layer

inside an entity (I), or outside (O). The NER tasks in BLURB are only concerned about one entity type (in JNLPBA, all the types are merged into one). In the case when there are multiple entity types, the BI tags would be further divided into fine-grained tags for specific types. Prior work has also considered more complex tagging schemes such as BIOUL, where U stands for the last word of an entity and L stands for a single-word entity. We also consider the simpler IO scheme that only differentiates between in and out of an entity. Classification is done using a simple linear layer or more sophisticated sequential labeling methods such as LSTM or **conditional random field (CRF)** [33].

- TransformInput: Returns the input sequence as is.
- Featurizer: Returns the BERT encoding of a given token.
- *Tagging scheme*: BIO*; BIOUL; IO.
- *Classification layer*: Linear layer*; LSTM; CRF.

PICO. Conceptually, evidence-based medical information extraction is akin to slot filling, as it tries to identify the PIO elements in an abstract describing a clinical trial. However, it can be formulated as a sequential tagging task like NER, by classifying tokens belonging to each element. A token may belong to more than one element (e.g., participant (P) and intervention (I)).

- TransformInput: Returns the input sequence as is.
- Featurizer: Returns the BERT encoding of a given token.
- *Tagging scheme*: BIO*; BIOUL; IO.
- *Classification layer*: Linear layer*; LSTM; CRF.

Relation extraction. Existing work on relation extraction tends to focus on binary relations. Given a pair of entity mentions in a text span (typically a sentence), the goal is to determine if the text indicates a relation for the mention pair. There are significant variations in the entity and relation representations. To prevent overfitting by memorizing the entity pairs, the entity tokens are often augmented with start/end markers or replaced by a dummy token. For featurization, the relation instance is either represented by a special [CLS] token or by concatenating the mention representations. In the latter case, if an entity mention contains multiple tokens, its representation is usually produced by pooling those of individual tokens (max or average). For computational efficiency, we use padding or truncation to set the input length to 128 tokens for GAD and 256 tokens for ChemProt and DDI that contain longer input sequences.

- TransformInput: Entity (dummification*; start/end marker; original); relation ([CLS]*; original).
- Featurizer: Entity (dummy token*; pooling); relation ([CLS] BERT encoding*; concatenation of the mention BERT encoding).
- *Classification layer*: Linear layer*; more sophisticated classifiers (e.g., MLP).

Sentence similarity. The similarity task can be formulated as a regression problem to generate a normalized score for a sentence pair. By default, a special [SEP] token is inserted to separate the two sentences, and a special

Table 5. Summary of Pretraining Details for the Various BERT Models Used in Our Experiments

	Vocabulary	Pretraining	Corpus	Text Size
BERT	Wiki + Books	—	Wiki + Books	3.3B words/16 GB
RoBERTa	Web crawl	—	Web crawl	160 GB
BioBERT	Wiki + Books	continual pretraining	PubMed	4.5B words
SciBERT	PMC + CS	from scratch	PMC + CS	3.2B words
ClinicalBERT	Wiki + Books	continual pretraining	MIMIC	0.5B words/3.7 GB
BlueBERT	Wiki + Books	continual pretraining	PubMed + MIMIC	4.5B words
PubMedBERT	PubMed	from scratch	PubMed	3.1B words/21 GB

Statistics for prior BERT models are taken from their publications when available. The size of a text corpus such as PubMed may vary a bit, depending on downloading time and preprocessing (e.g., filtering out empty or very short abstracts). Both BioBERT and PubMedBERT also have a version pretrained with additional PMC full text; here we list the standard version pretrained using PubMed only.

[CLS] token is prepended to the beginning to represent the pair. The BERT encoding of [CLS] is used to compute the regression score.

- TransformInput: [CLS] S_1 [SEP] S_2 [SEP], for sentence pair S_1, S_2.
- Featurizer: [CLS] BERT encoding.
- *Regression layer*: Linear regression.

Document classification. For each text span and category (an abstract and a cancer hallmark in HoC), the goal is to classify whether the text belongs to the category. By default, a [CLS] token is appended to the beginning of the text, and its BERT encoding is passed on by the Featurizer for the final classification, which typically uses a simple linear layer.

- TransformInput: [CLS] D [SEP], for document D.
- Featurizer: Returns [CLS] BERT encoding.
- *Classification layer*: Linear layer.

Question answering. For the two-way (yes/no) or three-way (yes/maybe/no) question-answering task, the encoding is similar to the sentence similarity task. Namely, a [CLS] token is prepended to the beginning, followed by the question and reference text, with a [SEP] token to separate the two text spans. The [CLS] BERT encoding is then used for the final classification. For computational efficiency, we use padding or truncation to set the input length to 512 tokens.

- TransformInput: [CLS] Q [SEP] T [SEP], for question Q and reference text T.
- Featurizer: Returns [CLS] BERT encoding.
- *Classification layer*: Linear layer.

2.5 Experimental Settings

For biomedical domain-specific pretraining, we generate the vocabulary and conduct pretraining using the latest collection of PubMed[5] abstracts: 14 million abstracts, 3.2 billion words, 21 GB. (The original collection contains more than 4 billion words; we filter out any abstracts with less than 128 words to reduce noise.)

We follow the standard pretraining procedure based on the TensorFlow implementation released by NVIDIA.[6] We use Adam [30] for the optimizer using a standard slanted triangular learning rate schedule with warm-up in 10% of steps and cool-down in 90% of steps. Specifically, the learning rate increases linearly from zero to the peak

---

[5]https://pubmed.ncbi.nlm.nih.gov/ (downloaded in February 2020).
[6]https://github.com/NVIDIA/DeepLearningExamples.

rate of 6×10^{-4} in the first 10% of steps, then decays linearly to zero in the remaining 90% of steps. Training is done for 62,500 steps with batch size of 8,192, which is comparable to the computation used in previous biomedical pretraining.[7] The training takes about 5 days on one DGX-2 machine with 16 V100 GPUs. We find that the cased version has similar performance to the uncased version in preliminary experiments; thus, we focus on uncased models in this study. We use WWM, with a masking rate of 15%. We denote the resulting BERT model *PubMedBERT*.

For comparison, we use the public releases of BERT [16], RoBERTa [39], BioBERT [34], SciBERT [8], ClinicalBERT [1], and BlueBERT [45]. Table 5 presents an overview. BioBERT and BlueBERT conduct continual pretraining from BERT, whereas ClinicalBERT conducts continual pretraining from BioBERT; thus, they all share the same vocabulary as BERT. BioBERT comes with two versions. We use BioBERT++ (v1.1), which was trained for a longer time and performed better. ClinicalBERT also comes with two versions. We use Bio+Clinical BERT.

Prior pretraining work has explored two settings: BERT-BASE with 12 transformer layers and 100 million parameters; BERT-LARGE with 24 transformer layers and 300 million parameters. Prior work in biomedical pretraining uses BERT-BASE only. For head-to-head comparison, we also use BERT-BASE in pretraining PubMedBERT. BERT-LARGE appears to yield improved performance in some preliminary experiments. We leave an in-depth exploration to future work.

For task-specific fine-tuning, we use Adam [30] with the standard slanted triangular learning rate schedule (warm-up in the first 10% of steps and cool-down in the remaining 90% of steps) and a dropout probability of 0.1. Due to random initialization of the task-specific model and drop out, the performance may vary for different random seeds, especially for small datasets like BIOSSES, BioASQ, and PubMedQA. We report the average scores from 10 runs for BIOSSES, BioASQ, and PubMedQA, and 5 runs for the others.

For all datasets, we use the development set for tuning the hyperparameters with the same range: learning rate (1e-5, 3e-5, 5e-5), batch size (16, 32), and epoch number (2–60). Ideally, we would conduct separate hyperparameter tuning for each model on each dataset. However, this would incur a prohibitive amount of computation, as we have to enumerate all combinations of models, datasets, and hyperparameters, each of which requires averaging over multiple runs with different randomization. In practice, we observe that the development performance is not very sensitive to hyperparameter selection, as long as they are in a ballpark range. Consequently, we focus on hyperparameter tuning using a subset of representative models such as BERT and BioBERT, and use a common set of hyperparameters for each dataset that work well for both out-domain and in-domain language models.

3 RESULTS

In this section, we conduct a thorough evaluation to assess the impact of domain-specific pretraining in biomedical NLP applications. First, we fix the standard task-specific model for each task in BLURB, and conduct a head-to-head comparison of domain-specific pretraining and mixed-domain pretraining. Next, we evaluate the impact of various pretraining options such as vocabulary, WWM, and adversarial pretraining. Finally, we fix a pretrained BERT model and compare various modeling choices for task-specific fine-tuning.

3.1 Domain-Specific Pretraining vs Mixed-Domain Pretraining

We compare BERT models by applying them to the downstream NLP applications in BLURB. For each task, we conduct the same fine-tuning process using the standard task-specific model as specified in Section 2.4. Table 6 shows the results.

By conducting domain-specific pretraining from scratch, PubMedBERT consistently outperforms all the other BERT models in most biomedical NLP tasks, often by a significant margin. The gains are most substantial against

---

[7]For example, BioBERT started with the standard BERT, which was pretrained using 1M steps with batch size of 256 and ran another 1M steps in continual pretraining.

Table 6. Comparison of Pretrained Language Models on the BLURB Biomedical NLP Benchmark

	BERT		RoBERTa	BioBERT	SciBERT		ClinicalBERT	BlueBERT	PubMedBERT
	Uncased	Cased	Cased	Cased	Uncased	Cased	Cased	Cased	Uncased
BC5-chem	89.25	89.99	89.43	92.85	92.49	92.51	90.80	91.19	**93.33**
BC5-disease	81.44	79.92	80.65	84.70	84.54	84.70	83.04	83.69	**85.62**
NCBI-disease	85.67	85.87	86.62	**89.13**	88.10	88.25	86.32	88.04	87.82
BC2GM	80.90	81.23	80.90	83.82	83.36	83.36	81.71	81.87	**84.52**
JNLPBA	77.69	77.51	77.86	78.55	78.68	78.51	78.07	77.71	**79.10**
EBM PICO	72.34	71.70	73.02	73.18	73.12	73.06	72.06	72.54	**73.38**
ChemProt	71.86	71.54	72.98	76.14	75.24	75.00	72.04	71.46	**77.24**
DDI	80.04	79.34	79.52	80.88	81.06	81.22	78.20	77.78	**82.36**
GAD	80.41	79.61	80.63	82.36	82.38	81.34	80.48	79.15	**83.96**
BIOSSES	82.68	81.40	81.25	89.52	86.25	87.15	91.23	85.38	**92.30**
HoC	80.20	80.12	79.66	81.54	80.66	81.16	80.74	80.48	**82.32**
PubMedQA	51.62	49.96	52.84	**60.24**	57.38	51.40	49.08	48.44	55.84
BioASQ	70.36	74.44	75.20	84.14	78.86	74.22	68.50	68.71	**87.56**
BLURB score	76.11	75.86	76.46	80.34	78.86	78.14	77.29	76.27	**81.16**

The standard task-specific models are used in the same fine-tuning process for all BERT models. The BLURB score is the macro average of average test results for each of the six tasks (NER, PICO, relation extraction, sentence similarity, document classification, question answering). See Table 3 for the evaluation metric used in each task.

BERT models trained using out-domain text. Notably, although the pretraining corpus is the largest for RoBERTa, its performance on biomedical NLP tasks is among the worst, similar to the original BERT model. Models using biomedical text in pretraining generally perform better. However, mixing out-domain data in pretraining generally leads to worse performance. In particular, even though clinical notes are more relevant to the biomedical domain than general-domain text, adding them does not confer any advantage, as is evident by the results of ClinicalBERT and BlueBERT. Not surprisingly, BioBERT is the closest to PubMedBERT, as it also uses PubMed text for pretraining. However, by conducting domain-specific pretraining from scratch, including using the PubMed vocabulary, PubMedBERT is able to obtain consistent gains over BioBERT in most tasks. A notable exception is PubMedQA, but this dataset is small, and there are relatively high variances among runs with different random seeds.

Compared to the published results for BioBERT, SciBERT, and BlueBERT in their original papers, our results are generally comparable or better for the tasks they have been evaluated on. The ClinicalBERT paper does not report any results on these biomedical applications [1].

3.2 Ablation Study on Pretraining Techniques

To assess the impact of pretraining options on downstream applications, we conduct several ablation studies using PubMedBERT as a running example. Table 7 shows results assessing the effect of vocabulary and WWM. Using the original BERT vocabulary derived from Wikipedia & BookCorpus (by continual pretraining from the original BERT), the results are significantly worse than using an in-domain vocabulary from PubMed. Additionally, WWM leads to consistent improvement across the board, regardless of the vocabulary in use. A significant advantage in using an in-domain vocabulary is that the input will be shorter in downstream tasks, as shown in Table 8, which makes learning easier. Figure 3 shows examples of how domain-specific pretraining with in-domain vocabulary helps correct errors from mixed-domain pretraining.

Furthermore, we found that pretraining on general-domain text provides no benefit even if we use the in-domain vocabulary (Table 9). The first column in Table 9 corresponds to BioBERT, which conducted pretraining first on the general domain and then on PubMed. The second column adopts the same continual pretraining

Table 7. Evaluation of the Impact of Vocabulary and WWM on the
Performance of PubMedBERT on BLURB

	Wiki + Books		PubMed	
	Word Piece	Whole Word	Word Piece	Whole Word
BC5-chem	93.20	93.31	92.96	**93.33**
BC5-disease	85.00	85.28	84.72	**85.62**
NCBI-disease	88.39	**88.53**	87.26	87.82
BC2GM	83.65	83.93	83.19	**84.52**
JNLPBA	78.83	78.77	78.63	**79.10**
EBM PICO	73.30	**73.52**	73.44	73.38
ChemProt	75.04	76.70	75.72	**77.24**
DDI	81.30	**82.60**	80.84	82.36
GAD	83.02	82.42	81.74	**83.96**
BIOSSES	91.36	91.79	**92.45**	92.30
HoC	81.76	81.74	80.38	**82.32**
PubMedQA	52.20	**55.92**	54.76	55.84
BioASQ	73.69	76.41	78.51	**87.56**
BLURB score	79.16	79.96	79.62	**81.16**

Table 8. Comparison of the Average Input
Length in Word Pieces Using General-Domain
vs In-Domain Vocabulary

Vocab	Wiki + Books	PubMed
BC5-chem	35.9	28.0
BC5-disease	35.9	28.0
NCBI-disease	34.2	27.4
BC2GM	38.5	30.5
JNLPBA	33.7	26.0
EBM PICO	30.7	25.1
ChemProt	75.4	55.5
DDI	106.0	75.9
GAD	47.0	35.7
BIOSSES	80.7	61.6
HoC	40.6	31.0
PubMedQA	343.1	293.0
BioASQ	702.4	541.4

strategy, except that the in-domain vocabulary (from PubMed) was used, which actually led to slight degradation in performance. However, by conducting pretraining from scratch on PubMed, we attained similar performance even with half of the compute (third column) and attained significant gain with the same amount of compute (fourth column; PubMedBERT). In sum, general-domain pretraining confers no advantage here in domain-specific pretraining.

In our standard PubMedBERT pretraining, we used PubMed abstracts only. We also tried adding full-text articles from PMC,[8] with the total pretraining text increased substantially to 16.8 billion words (107 GB). Surprisingly,

---

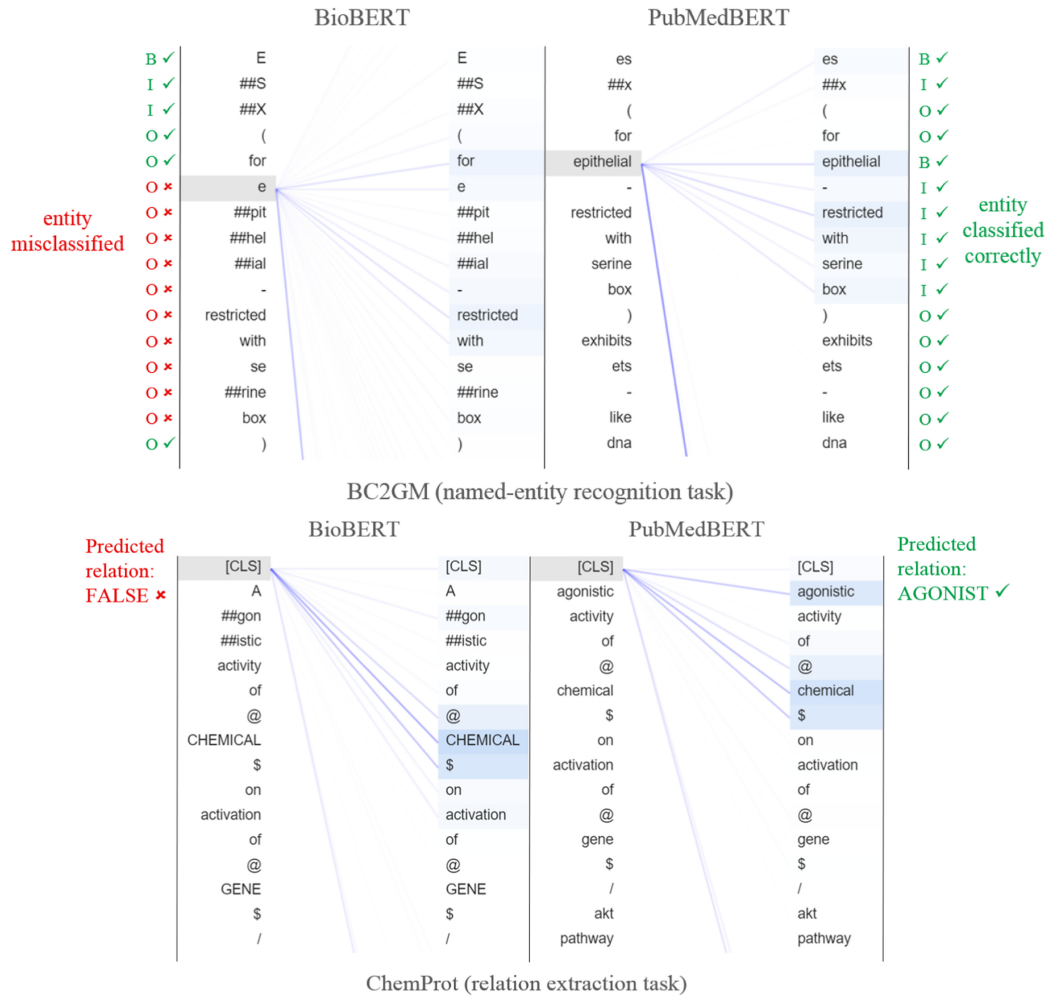[8]https://www.ncbi.nlm.nih.gov/pmc/.

Fig. 3. Examples of how domain-specific pretraining helps correct errors from mixed-domain pretraining. Top: Attention for the leading word piece of the gene mention "epithelial-restricted with serine box" (abbreviation "ESX") in the BC2GM dataset. Bottom: Attention for the [CLS] token in an instance of AGONIST relation between a pair of dummified chemical and protein. In both cases, we show the aggregate attention from the penultimate layer to the preceding layer, which tends to be most informative about the final classification. Note how BioBERT tends to shatter the relevant words by inheriting the general-domain vocabulary. The domain-specific vocabulary enables PubMedBERT to learn better attention patterns and make correct predictions.

this generally leads to a slight degradation in performance across the board. However, by extending pretraining for 60% longer (100K steps in total), the overall results improve and slightly outperform the standard PubMed-BERT using only abstracts. The improvement is somewhat mixed across the tasks, with some gaining and others losing. We hypothesize that the reason for this behavior is twofold. First, PMC inclusion is influenced by funding policy and differs from general PubMed distribution, and full texts generally contain more noise than abstracts. As most existing biomedical NLP tasks are based on abstracts, full texts may be slightly out-domain compared to abstracts. Moreover, even if full texts are potentially helpful, their inclusion requires additional pretraining cycles to make use of the extra information.

Table 9. Evaluation of the Impact of Pretraining Corpora and Time on the
Performance on BLURB

Pretraining	Wiki + Books → PubMed		PubMed (half time)	PubMed
Vocab	Wiki + Books	PubMed	PubMed	PubMed
BC5-chem	92.85	**93.41**	93.05	93.33
BC5-disease	84.70	85.43	85.02	**85.62**
NCBI-disease	**89.13**	87.60	87.77	87.82
BC2GM	83.82	84.03	84.11	**84.52**
JNLPBA	78.55	79.01	78.98	**79.10**
EBM PICO	73.18	**73.80**	73.74	73.38
ChemProt	76.14	77.05	76.69	**77.24**
DDI	80.88	81.21	81.96	**82.36**
GAD	82.36	82.47	82.80	**83.96**
BIOSSES	89.52	89.93	92.12	**92.30**
HoC	81.54	**83.14**	82.13	82.32
PubMedQA	**60.24**	54.84	55.28	55.84
BioASQ	84.14	79.00	79.43	**87.56**
BLURB score	80.34	80.03	80.23	**81.16**

In the first two columns, pretraining was first conducted on Wiki & Books, then on PubMed
abstracts. All use the same amount of compute (twice as long as original BERT pretraining),
except for the third column, which only uses half (same as original BERT pretraining).

Table 10. Evaluation of the Impact of Pretraining Text on the Performance
of PubMedBERT on BLURB

	PubMed	PubMed + PMC	PubMed + PMC (longer training)
BC5-chem	93.33	**93.36**	93.34
BC5-disease	85.62	85.62	**85.76**
NCBI-disease	87.82	**88.34**	88.04
BC2GM	**84.52**	84.39	84.37
JNLPBA	79.10	78.90	**79.16**
EBM PICO	73.38	73.64	**73.72**
ChemProt	**77.24**	76.96	76.80
DDI	82.36	**83.56**	82.06
GAD	**82.34**	82.24	81.58
BIOSSES	92.30	90.39	**92.31**
HoC	82.32	82.16	**82.62**
PubMedQA	55.84	**61.02**	60.02
BioASQ	**87.56**	83.43	87.20
BLURB score	81.07	80.91	**81.42**

The first result column corresponds to the standard PubMedBERT pretrained using PubMed
abstracts ("PubMed"). The second one corresponds to PubMedBERT trained using both PubMed
abstracts and PMC full text ("PubMed+PMC"). The last one corresponds to PubMedBERT trained
using both PubMed abstracts and PMC full text, for 60% longer ("PubMed+PMC (longer training)").

Adversarial pretraining has been shown to be highly effective in boosting performance in general-domain
applications [37]. We thus conducted adversarial pretraining in PubMedBERT and compared its performance
with standard pretraining (Table 11). Surprisingly, adversarial pretraining generally leads to a slight degradation

Table 11. Comparison of PubMedBERT Performance on
BLURB Using Standard and Adversarial Pretraining

	PubMedBERT	+ Adversarial
BC5-chem	**93.33**	93.17
BC5-disease	**85.62**	85.48
NCBI-disease	87.82	**87.99**
BC2GM	**84.52**	84.07
JNLPBA	79.10	**79.18**
EBM PICO	**73.38**	72.92
ChemProt	**77.24**	77.04
DDI	82.36	**83.62**
GAD	**83.96**	83.54
BIOSSES	92.30	**94.11**
HoC	**82.32**	82.20
PubMedQA	**55.84**	53.30
BioASQ	**87.56**	82.71
BLURB score	**81.16**	80.77

in performance, with some exceptions such as sentence similarity (BIOSSES). We hypothesize that the reason may be similar to what we observe in pretraining with full texts. Namely, adversarial training is most useful if the pretraining corpus is more diverse and relatively out-domain compared to the application tasks. We leave a more thorough evaluation of adversarial pretraining to future work.

3.3 Ablation Study on Fine-Tuning Methods

In the preceding studies on pretraining methods, we fix the fine-tuning methods to the standard methods described in Section 2.4. Next, we will study the effect of modeling choices in task-specific fine-tuning, by fixing the underlying pretrained language model to our standard PubMedBERT (WWM, PubMed vocabulary, pretrained using PubMed abstracts).

Prior to the current success of pretraining neural language models, standard NLP approaches were often dominated by sequential labeling methods, such as CRF and more recently recurrent neural networks such as LSTM. Such methods were particularly popular for NER and relation extraction.

With the advent of BERT models and the self-attention mechanism, the utility of explicit sequential modeling becomes questionable. The top layer in the BERT model already captures many non-linear dependencies across the entire text span. Therefore, it is conceivable that even a linear layer on top can perform competitively. We find that this is indeed the case for NER and relation extraction, as shown in Table 12. The use of a bidirectional LSTM (Bi-LSTM) does not lead to any substantial gain compared to linear layer.

We also investigate the tagging scheme used in NER. The standard tagging scheme distinguishes words by their positions within an entity. For sequential tagging methods such as CRF and LSTM, distinguishing the position within an entity is potentially advantageous compared to the minimal IO scheme that only distinguishes between the inside and outside of entities. But for BERT models, once again, the utility of more complex tagging schemes is diminished. We thus conducted a head-to-head comparison of the tagging schemes using three biomedical NER tasks in BLURB. As we can see in Table 13, the difference is minuscule, suggesting that with self-attention, the sequential nature of the tags is less essential in NER modeling.

The use of neural methods also has subtle, but significant, implications for relation extraction. Previously, relation extraction was generally framed as a classification problem with manually crafted feature templates. To prevent overfitting and enhance generalization, the feature templates would typically avoid using the entities

Table 12. Comparison of Linear Layers vs Recurrent
Neural Networks for Task-Specific Fine Tuning in NER
(Entity-level F1) and Relation Extraction (Micro F1), All
Using the Standard PubMedBERT

Task-Specific Model	Linear Layer	Bi-LSTM
BC5-chem	**93.33**	93.12
BC5-disease	85.62	**85.64**
JNLPBA	**79.10**	**79.10**
ChemProt	**77.24**	75.40
DDI	**82.36**	81.70
GAD	**83.96**	**83.42**

Table 13. Comparison of Entity-Level F1 for
Biomedical NER Using Different Tagging
Schemes and the Standard PubMedBERT

Tagging Scheme	BIO	BIOUL	IO
BC5-chem	93.33	**93.37**	93.11
BC5-disease	85.62	85.59	**85.63**
JNLPBA	**79.10**	79.02	79.05

in question. Neural methods do not need handcrafted features but rather use the neural encoding of the given text span, including the entities themselves. This introduces a potential risk that the neural network may simply memorize the entity combination. This problem is particularly pronounced in self-supervision settings, such as distant supervision, because the positive instances are derived from entity tuples with known relations. As a result, it is a common practice to "dummify" entities (i.e., replace an entity with a generic tag such as $DRUG or $GENE) [24, 58].

This risk remains in the standard supervised setting, such as in the tasks that comprise BLURB. We thus conducted a systematic evaluation of entity dummification and relation encoding, using two relation extraction tasks in BLURB.

For entity marking, we consider three variants: dummify the entities in question; use the original text; add start and end tags to entities in question. For relation encoding, we consider three schemes. In the [CLS] encoding introduced by the original BERT paper, the special token [CLS] is prepended to the beginning of the text span, and its contextual representation at the top layer is used as the input in the final classification. Another standard approach concatenates the BERT encoding of the given entity mentions, each obtained by applying max pooling to the corresponding token representations. Finally, following prior work, we also consider simply concatenating the top contextual representation of the entity start tag, if the entity markers are in use [7].

Table 14 shows the results. Simply using the original text indeed exposes the neural methods to significant overfitting risk. Using [CLS] with the original text is the worst choice, as the relation encoding has a hard time distinguishing which entities in the text span are in question. Dummification remains the most reliable method, which works for either relation encoding method. Interestingly, using entity markers leads to slightly better results in both datasets, as it appears to prevent overfitting while preserving useful entity information. We leave it to future work to study whether this would generalize to all relation extraction tasks.

4 DISCUSSION

Standard supervised learning requires labeled examples, which are expensive and time consuming to annotate. Self-supervision using unlabeled text is thus a long-standing direction for alleviating the annotation bottleneck

Table 14. Evaluation of the Impact of Entity Dummification and Relation Encoding
in Relation Extraction, All Using PubMedBERT

Input Text	Classification Encoding	ChemProt	DDI
ENTITY DUMMIFICATION	[CLS]	77.24	82.36
ENTITY DUMMIFICATION	MENTION	77.22	82.08
ORIGINAL	[CLS]	50.52	37.00
ORIGINAL	MENTION	75.48	79.42
ENTITY MARKERS	[CLS]	**77.72**	82.22
ENTITY MARKERS	MENTION	77.22	**82.42**
ENTITY MARKERS	ENTITY START	77.58	82.18

With entity dummification, the entity mentions in question are anonymized using entity type
tags such as $DRUG or $GENE. With entity marker, special tags marking the start and end of an
entity are appended to the entity mentions in question. Relation encoding is derived from the
special [CLS] token appended to the beginning of the text or the special entity start token, or by
concatenating the contextual representation of the entity mentions in question.

using transfer learning. Early methods focused on clustering related words using distributed similarity, such as
Brown Clusters [12, 36]. With the revival of neural approaches, neural embedding has become the new staple for
transfer learning from unlabeled text. This starts with simple stand-alone word embeddings [41, 46] and evolves
into more sophisticated pretrained language models, from LSTM in ULMFiT [23] and ELMo [47] to transformer-
based models in GPT [48, 49] and BERT [16, 39]. Their success is fueled by access to large text corpora, advanced
hardware such as GPUs, and a culmination of advances in optimization methods, such as Adam [30] and slanted
triangular learning rate [23]. Here, transfer learning goes from the pretrained language models to fine-tuning
task-specific models for downstream applications.

As the community ventures beyond the standard newswire and Web domains, and begins to explore high-
value verticals such as biomedicine, a different kind of transfer learning is brought into play by combining text
from various domains in pretraining language models. The prevailing assumption is that such mixed-domain
pretraining is advantageous. In this article, we show that this type of transfer learning may not be applicable
when there is a sufficient amount of in-domain text, as is the case in biomedicine. In fact, our experiments
comparing clinical BERTs with PubMedBERT on biomedical NLP tasks show that even related text such as clinical
notes may not be helpful, since we already have abundant biomedical text from PubMed. Our results show that
we should distinguish different types of transfer learning and separately assess their utility in various situations.

There are a plethora of biomedical NLP datasets, especially from various shared tasks such as BioCre-
ative [3, 29, 40, 53], BioNLP [15, 28], SemEval [2, 9, 10, 17], and BioASQ [42]. The focus has evolved from simple
tasks, such as NER, to more sophisticated tasks, such as relation extraction and question answering, and new
tasks have been proposed for emerging application scenarios such as evidence-based medical information extrac-
tion [44]. However, although comprehensive benchmarks and leaderboards are available for the general domains
(e.g., GLUE [57] and SuperGLUE [56]), they are still a rarity in biomedical NLP. In this article, inspired by prior
effort toward this direction [45], we create the first leaderboard for biomedical NLP, BLURB—a comprehensive
benchmark containing 13 datasets for six tasks.

5 CONCLUSION

In this article, we challenge a prevailing assumption in pretraining neural language models and show that domain-
specific pretraining from scratch can significantly outperform mixed-domain pretraining such as continual pre-
training from a general-domain language model, leading to new state-of-the-art results for a wide range of
biomedical NLP applications. To facilitate this study, we create BLURB, a comprehensive benchmark for biomed-
ical NLP featuring a diverse set of tasks such as NER, relation extraction, document classification, and question

answering. To accelerate research in biomedical NLP, we release our state-of-the-art biomedical BERT models and set up a leaderboard based on BLURB.

Future directions include further exploration of domain-specific pretraining strategies, incorporating more tasks in biomedical NLP, and extension of the BLURB benchmark to clinical and other high-value domains.

REFERENCES

[1] Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jindi, Tristan Naumann, and Matthew McDermott. 2019. Publicly available clinical BERT embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop.* 72–78. https://doi.org/10.18653/v1/W19-1909

[2] Marianna Apidianaki, Saif M. Mohammad, Jonathan May, Ekaterina Shutova, Steven Bethard, and Marine Carpuat (Eds.). 2018. *Proceedings of the 12th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2018, New Orleans, Louisiana, USA, June 5-6, 2018.* Association for Computational Linguistics. https://www.aclweb.org/anthology/volumes/S18-1/.

[3] Cecilia N. Arighi, Phoebe M. Roberts, Shashank Agarwal, Sanmitra Bhattacharya, Gianni Cesareni, Andrew Chatr-aryamontri, Simon Clematide, et al. 2011. BioCreative III interactive task: An overview. *BMC Bioinformatics* 12, 8 (Oct. 2011), S4. https://doi.org/10.1186/1471-2105-12-S8-S4

[4] Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing.* 355–362. https://www.aclweb.org/anthology/D11-1033.

[5] Simon Baker, Imran Ali, Ilona Silins, Sampo Pyysalo, Yufan Guo, Johan Högberg, Ulla Stenius, and Anna Korhonen. 2017. Cancer hallmarks analytics tool (CHAT): A text mining approach to organize and evaluate scientific literature on cancer. *Bioinformatics* 33, 24 (2017), 3973–3981.

[6] Simon Baker, Ilona Silins, Yufan Guo, Imran Ali, Johan Högberg, Ulla Stenius, and Anna Korhonen. 2015. Automatic semantic classification of scientific literature according to the hallmarks of cancer. *Bioinformatics* 32, 3 (2015), 432–440.

[7] Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics.* 2895–2905. https://doi.org/10.18653/v1/P19-1279

[8] Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. 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'19).* 3615–3620. https://doi.org/10.18653/v1/D19-1371

[9] Steven Bethard, Marine Carpuat, Marianna Apidianaki, Saif M. Mohammad, Daniel M. Cer, and David Jurgens (Eds.). 2017. *Proceedings of the 11th International Workshop on Semantic Evaluation, SemEval@ACL 2017, Vancouver, Canada, August 3-4, 2017.* Association for Computational Linguistics. https://www.aclweb.org/anthology/volumes/S17-2/.

[10] Steven Bethard, Daniel M. Cer, Marine Carpuat, David Jurgens, Preslav Nakov, and Torsten Zesch (Eds.). 2016. *Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2016, San Diego, CA, USA, June 16-17, 2016.* Association for Computational Linguistics. https://www.aclweb.org/anthology/volumes/S16-1/.

[11] Àlex Bravo, Janet Piñero, Núria Queralt-Rosinach, Michael Rautschka, and Laura I. Furlong. 2015. Extraction of relations between genes and diseases from text and large-scale data analysis: Implications for translational research. *BMC Bioinformatics* 16, 1 (2015), 55.

[12] Peter F. Brown, Vincent J. Della Pietra, Peter V. Desouza, Jennifer C. Lai, and Robert L. Mercer. 1992. Class-based n-gram models of natural language. *Computational Linguistics* 18, 4 (1992), 467–480.

[13] Rich Caruana. 1997. Multitask learning. *Machine Learning* 28, 1 (1997), 41–75.

[14] Gamal Crichton, Sampo Pyysalo, Billy Chiu, and Anna Korhonen. 2017. A neural network multi-task learning approach to biomedical named entity recognition. *BMC Bioinformatics* 18, 1 (2017), 368.

[15] Dina Demner-Fushman, Kevin Bretonnel Cohen, Sophia Ananiadou, and Junichi Tsujii (Eds.). 2019. *Proceedings of the 18th BioNLP Workshop and Shared Task, BioNLP@ACL 2019, Florence, Italy, August 1, 2019.* Association for Computational Linguistics. https://www.aclweb.org/anthology/volumes/W19-50/.

[16] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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).* 4171–4186.

[17] Mona T. Diab, Timothy Baldwin, and Marco Baroni (Eds.). 2013. *Proceedings of the 7th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2013, Atlanta, Georgia, USA, June 14-15, 2013.* Association for Computational Linguistics. https://www.aclweb.org/anthology/volumes/S13-2/.

[18] Rezarta Islamaj Doğan, Robert Leaman, and Zhiyong Lu. 2014. NCBI disease corpus: A resource for disease name recognition and concept normalization. *Journal of Biomedical Informatics* 47 (2014), 1–10.

[19] Jingcheng Du, Qingyu Chen, Yifan Peng, Yang Xiang, Cui Tao, and Zhiyong Lu. 2019. ML-Net: Multi-label classification of biomedical texts with deep neural networks. *Journal of the American Medical Informatics Association* 26, 11 (2019), 1279–1285. https://doi.org/10.1093/jamia/ocz085

[20] Douglas Hanahan and Robert A. Weinberg. 2000. The hallmarks of cancer. *Cell* 100, 1 (2000), 57–70.

[21] María Herrero-Zazo, Isabel Segura-Bedmar, Paloma Martínez, and Thierry Declerck. 2013. The DDI corpus: An annotated corpus with pharmacological substances and drug–drug interactions. *Journal of Biomedical Informatics* 46, 5 (2013), 914–920.

[22] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9, 8 (1997), 1735–1780.

[23] Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 328–339. https://doi.org/10.18653/v1/P18-1031

[24] Robin Jia, Cliff Wong, and Hoifung Poon. 2019. Document-level N-ary relation extraction with multiscale representation learning. In *Proceedings of the 17th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT'19)*.

[25] Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. 2019. PubMedQA: A dataset for biomedical research question answering. 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'19)*. 2567–2577. https://doi.org/10.18653/v1/D19-1259

[26] Alistair E. W. Johnson, Tom J. Pollard, Lu Shen, Li-Wei 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. *Scientific Data* 3, 1 (May 2016), 160035. https://doi.org/10.1038/sdata.2016.35

[27] Jin-Dong Kim, Tomoko Ohta, Yoshimasa Tsuruoka, Yuka Tateisi, and Nigel Collier. 2004. Introduction to the bio-entity recognition task at JNLPBA. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and Its Applications (NLPBA/BioNLP'04)*. 73–78. https://www.aclweb.org/anthology/W04-1213.

[28] Jin-Dong Kim, Yue Wang, Toshihisa Takagi, and Akinori Yonezawa. 2011. Overview of Genia Event Task in BioNLP Shared Task 2011. In *Proceedings of the BioNLP Shared Task 2011 Workshop (BioNLP Shared Task'11)*. 7–15.

[29] Sun Kim, Rezarta Islamaj Dogan, Andrew Chatr-aryamontri, Mike Tyers, W. John Wilbur, and Donald C. Comeau. 2015. Overview of BioCreative V BioC track. In *Proceedings of the Fifth BioCreative Challenge Evaluation Workshop, Sevilla, Spain*. 1–9.

[30] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR'15)*. http://arxiv.org/abs/1412.6980.

[31] Martin Krallinger, Obdulia Rabal, Saber A. Akhondi, Martın Pérez Pérez, Jesús Santamaría, G. P. Rodríguez, G. Tsatsaronis, et al. 2017. Overview of the BioCreative VI Chemical-Protein Interaction Track. In *Proceedings of the 6th BioCreative Challenge Evaluation Workshop*, Vol. 1. 141–146.

[32] Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. 66–71. https://doi.org/10.18653/v1/D18-2012

[33] John Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning*. 282–289.

[34] 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. *Bioinformatics* 36, 4 (2019), 1234–1240. https://doi.org/10.1093/bioinformatics/btz682

[35] 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*. Online, May 8, 2016.

[36] Percy Liang. 2005. *Semi-Supervised Learning for Natural Language*. Ph.D. Dissertation. Massachusetts Institute of Technology, Cambridge, MA.

[37] Xiaodong Liu, Hao Cheng, Pengcheng He, Weizhu Chen, Yu Wang, Hoifung Poon, and Jianfeng Gao. 2020. Adversarial training for large neural language models. arXiv:2004.08994.

[38] Xiaodong Liu, Jianfeng Gao, Xiaodong He, Li Deng, Kevin Duh, and Ye-Yi Wang. 2015. Representation learning using multi-task deep neural networks for semantic classification and information retrieval. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 912–921.

[39] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. arXiv:1907.11692.

[40] Yuqing Mao, Kimberly Van Auken, Donghui Li, Cecilia N. Arighi, Peter McQuilton, G. Thomas Hayman, Susan Tweedie, et al. 2014. Overview of the gene ontology task at BioCreative IV. *Database*. Online, August 25, 2014.

[41] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. arXiv:1301.3781.

[42] Anastasios Nentidis, Konstantinos Bougiatiotis, Anastasia Krithara, and Georgios Paliouras. 2019. Results of the seventh edition of the BioASQ challenge. In *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. 553–568.

[43] Hermann Ney, Ute Essen, and Reinhard Kneser. 1994. On structuring probabilistic dependences in stochastic language modelling. *Computer Speech & Language* 8, 1 (1994), 1–38. https://doi.org/10.1006/csla.1994.1001

[44] Benjamin Nye, Junyi Jessy Li, Roma Patel, Yinfei Yang, Iain J. 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 *Proceedings of the Conference of the Association for Computational Linguistics*, Vol. 2018. 197.

[45] Yifan Peng, Shankai Yan, and Zhiyong Lu. 2019. Transfer learning in biomedical natural language processing: An evaluation of BERT and ELMo on ten benchmarking datasets. In *Proceedings of the 18th BioNLP Workshop and Shared Task*. 58–65. https://doi.org/10.18653/v1/W19-5006

[46] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP'14)*. 1532–1543.

[47] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*. 2227–2237. https://doi.org/10.18653/v1/N18-1202

[48] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. https://www.cs.ubc.ca/~amuham01/LING530/papers/radford2018improving.pdf.

[49] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog* 1, 8 (2019), 9.

[50] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research* 21, 140 (2020), 1–67. http://jmlr.org/papers/v21/20-074.html.

[51] Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. 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)*. 1715–1725. https://doi.org/10.18653/v1/P16-1162

[52] Yuqi Si, Jingqi Wang, Hua Xu, and Kirk Roberts. 2019. Enhancing clinical concept extraction with contextual embeddings. *Journal of the American Medical Informatics Association* 26, 11 (2019), 1297–1304.

[53] Larry Smith, Lorraine K. Tanabe, Rie Johnson nee Ando, Cheng-Ju Kuo, I.-Fang Chung, Chun-Nan Hsu, Yu-Shi Lin, et al. 2008. Overview of BioCreative II gene mention recognition. *Genome Biology* 9 (2008), S2.

[54] Gizem Soğancıoğlu, Hakime Öztürk, and Arzucan Özgür. 2017. BIOSSES: A semantic sentence similarity estimation system for the biomedical domain. *Bioinformatics* 33, 14 (2017), i49–i58.

[55] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*. 5998–6008.

[56] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems*. 3266–3280.

[57] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2019 International Conference on Learning Representations (ICLR'19)*.

[58] Hai Wang and Hoifung Poon. 2018. Deep probabilistic logic: A unifying framework for indirect supervision. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP'18)*.

[59] Yichong Xu, Xiaodong Liu, Yelong Shen, Jingjing Liu, and Jianfeng Gao. 2019. Multi-task learning with sample re-weighting for machine reading comprehension. 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)*. 2644–2655. https://doi.org/10.18653/v1/N19-1271

[60] M. Zhang and Z. Zhou. 2014. A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering* 26, 8 (2014), 1819–1837. https://doi.org/10.1109/TKDE.2013.39

[61] Yijia Zhang, Wei Zheng, Hongfei Lin, Jian Wang, Zhihao Yang, and Michel Dumontier. 2018. Drug–drug interaction extraction via hierarchical RNNs on sequence and shortest dependency paths. *Bioinformatics* 34, 5 (2018), 828–835.

[62] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV'15)*.