Tri-Train: Automatic Pre-Fine Tuning between Pre-Training and Fine-Tuning for SciNER

Qingkai Zeng[†], **Wenhao Yu**[†], **Mengxia Yu**[†], **Tianwen Jiang**[‡],
Tim Weninger[†], **Meng Jiang**[†]
†University of Notre Dame, Notre Dame, IN, USA
‡Harbin Institute of Technology, Harbin, Heilongjiang, China
†{qzeng, wyu1, myu2, tweninge, mjiang2}@nd.edu
‡{twjiang}@ir.hit.edu.cn

Abstract

The training process of scientific NER models is commonly performed in two steps: *i)* Pre-training a language model by self-supervised tasks on huge data and *ii)* fine-tune training with small labelled data. The success of the strategy depends on the relevance between the data domains and between the tasks. However, gaps are found in practice when the target domains are specific and small. We propose a novel framework to introduce a "pre-fine tuning" step between pre-training and fine-tuning. It constructs a corpus by selecting sentences from unlabeled documents that are the most relevant with the labelled training data. Instead of predicting tokens in random spans, the pre-fine tuning task is to predict tokens in entity candidates identified by text mining methods. Pre-fine tuning is automatic and light-weight because the corpus size can be much smaller than pre-training data to achieve a better performance. Experiments on seven benchmarks demonstrate the effectiveness.

1 Introduction

In many scientific domains such as biomedicine and computer science (CS), named entity recognition (NER) is a fundamental information extraction task (Nédellec et al., 2013; Luan et al., 2018; Zeng et al., 2019; Jiang et al., 2020). Like many other natural language processing (NLP) tasks, language modeling plays an essential role in learning to perform like a domain expert (Jiang et al., 2019; Yu et al., 2020; Zhang et al., 2020). Two-step training process has been widely used in NLP research, especially for domain-specific NER. *Step 1:* Pre-train a language model by self-supervised task(s) such as masked token prediction and next sentence prediction on large-scale datasets of billions of tokens. *Step 2:* Fine-tune the neural model on a target task with a carefully labelled domain-specific dataset. Here, the premises of the strategy's success are that *i)* the pre-training corpora and fine-tuning labelled data are domain-relevant; *ii)* the pre-training task(s) and fine-tuning task are also relevant.

Data relevance becomes an issue for scientific NER (SciNER). Language models like BERT (Devlin et al., 2019) were pre-trained on general corpora such as English Wikipedia and Books Corpus. Therefore, SciBERT (Beltagy et al., 2019) was developed on paper corpus (3.2B tokens) in CS and bio domains from Semantic Scholar. BioBERT (Lee et al., 2020) added the PubMed Central (PMC) full-text corpus (13.5B tokens). They both applied the same pre-training tasks as the regular BERT. The pre-training cost week(s) on 8 TPUs or V100 GPUs to win over the general-domain BERT.

However, the data relevance gap was not fully fixed because the target data were often collected in a very specific and small research field such as BRCA1-related breast cancer and vision AI, while the pre-training corpora were too broad to adapt the model effectively to the target task. Moreover, the pre-training aimed at predicting *random* masked tokens or tokens in *random* masked spans (Joshi et al., 2020), which could be too weakly associated with the target task of entity recognition and typing. It is important to bridge the gaps of data and tasks for accurate SciNER.

In this work, we propose a novel framework called Tri-Train that introduces a training step between heavy pre-training (PT) and small-data fine tuning (FT). We name it "pre-fine tuning" (PFT). The framework is illustrated in Figure 1. First, it constructs a corpus by selecting a set of sentences relevant with the labelled training data (the specific research field) from unlabeled auxiliary corpora (like those used for pre-training). The PFT corpus is of a medium size compared to those in PT and FT (300K tokens). Second, we optimize the pre-trained model parameters on two tasks. Instead of random masked tokens or tokens in random masked

Findings of the Association for Computational Linguistics: EMNLP 2020, pages 4778–4787
November 16 - 20, 2020.

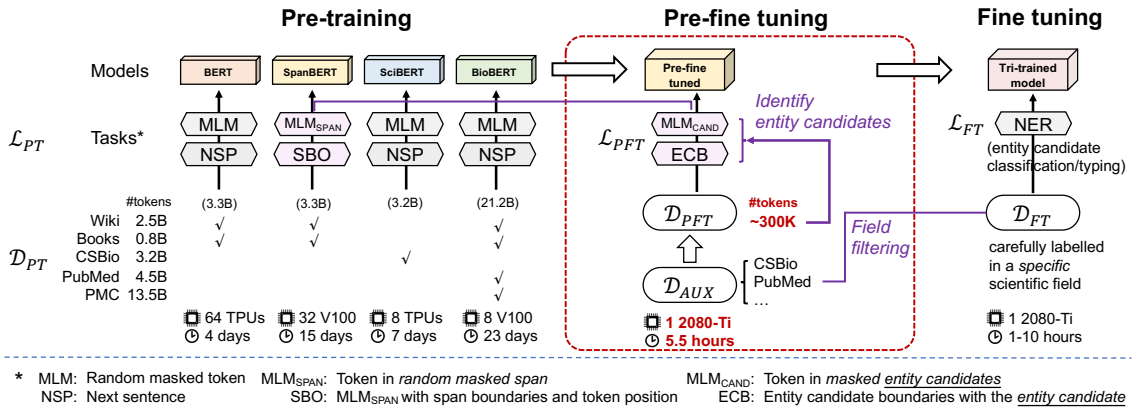


Figure 1: In the proposed Tri-Train framework, we introduce a training step called *pre-fine tuning* between pre-training and fine tuning. It includes *i)* corpus construction $\mathcal{D}_{\text{PFT}}$ by filtering with target data from a specific field and *ii)* two novel tasks, MLM_{CAND} and ECB, supported by automatic entity candidate recognition methods. The pre-fine tuning is light-weight compared to pre-training and improves the performance in many SciNER datasets.

spans, the first task is to predict tokens in masked entity candidates. The entity candidates in the PFT corpus were automatically identified by text mining methods (Adar and Datta, 2015; Shang et al., 2018). The second task is to predict the candidate's boundary tokens using the entity candidates themselves. These two tasks are designed based on regular pre-training tasks and bringing the knowledge that SciNER in the target domain needs into consideration. When practitioners found the performance of two-step framework was unsatisfactory and they were not able to re-train the heavy BERTs, our framework would be an effective solution.

We evaluate our framework based on **four** kinds of BERT models: regular BERT, SpanBERT (Joshi et al., 2020), BioBERT, and SciBERT. Experiments on **seven** SciNER benchmarks demonstrate the effectiveness of the proposed framework.

2 Preliminaries

2.1 Problem Definition

Input Given a SciNER corpus $\mathcal{D}_{\text{FT}}$, we derive a set of word sequence spans (up to length L) in the corpus $\mathcal{S}$. $\mathcal{D}_{\text{FT}}$ was labelled by a schema of entity types $\mathcal{Y}$. And suppose we have an auxiliary corpus $\mathcal{D}_{\text{AUX}}$ which is much bigger than $\mathcal{D}_{\text{FT}}$.

Output Predict whether a span $S \in \mathcal{S}$ is an entity, and if it is, predict the entity type $y \in \mathcal{Y}$.

2.2 Pre-training: SpanBERT

SpanBERT (Joshi et al., 2020) is a self-supervised pre-training language model inspired by BERT (Devlin et al., 2019). It extends BERT by (1) masking contiguous random spans instead of random tokens, and (2) training span boundary representations to predict the entire masked span, without relying on the individual token representations. SpanBERT has two objectives: span mask prediction (MLM_{SPAN}) and span boundary objective (SBO).

2.2.1 Span mask prediction (MLM_{SPAN})

Pre-training models require large pre-training corpora $\mathcal{D}_{\text{PT}}$. Span masking iteratively samples spans of text. In each iteration, it randomly selects a starting point for a span to be masked and the length of span is determined by a geometric distribution $l \sim Geo(p)$. Given a sentence $X = (x_1, \ldots, x_n) \in \mathcal{D}_{\text{PT}}$, a masked span of tokens can be represented as $(x_s, ..., x_e)$, where (s, e) indicates its start and end positions. For each token $x_i \in (x_s, ..., x_e)$, the embedding can be designated as $\mathbf{x}_i$. The loss of span mask prediction can be seen as a standard masked language model loss on a continuous span, which can be represented as:

$$\mathcal{L}_{\text{MLM\_SPAN}} = -\log\left(\prod_{i=s}^{e} P(x_i|\mathbf{x}_i)\right).$$

2.2.2 Span boundary objective (SBO)

In span selection models, boundary tokens play a crucial role in span representation. SpanBERT introduces an objective that involves predicting each token of a masked span using only the representations of the observed tokens at the boundaries. For each token x_i, it is encoded by the external boundary tokens' embedding $\mathbf{x}_{s-1}, \mathbf{x}_{e+1}$ and the position embedding of target token $\mathbf{p}_{i-s+1}$:

$$\mathbf{y}_i = f(\mathbf{x}_{s-1}, \mathbf{x}_{e+1}, \mathbf{p}_{i-s+1}),$$

where $f(\cdot)$ is a two-layer feed forward neural network. The SBO loss is

$$\mathcal{L}_{\text{SBO}} = -\log\left(\prod_{i=s}^{e} P(x_i|\mathbf{y}_i)\right).$$

SpanBERT sums the loss from both MLM_{SPAN} and SBO for each token x_i:

$$\mathcal{L}_{\text{PT}} = \mathcal{L}_{\text{MLM\_SPAN}}(x_i, \mathbf{x}_i) + \mathcal{L}_{\text{SBO}}(x_i, \mathbf{y}_i).$$

2.3 Fine-tuning: SciNER

Pre-trained models can be fine tuned for SciNER. They first encode a span's tokens to contextualized representations, then reduce into a single vector through a non-parameterized function, and finally put it into a reader layer to predict the entity type.

2.3.1 Span embedding encoder

Token embedding layer The encoder of BERT based models generates contextualized embeddings for each token x in sentence X:

$$\mathbf{x} = \text{BERT}_{\Theta}(X)[x] \in \mathbb{R}^d,$$

where Θ are model parameters and $d = 768$.

Span representation layer The embedding of each span S is a concatenation of the max-pooling results of token embeddings and span width features. The width feature vector $\mathbf{s}_{\text{width}}$ is learned by back propagation. The embedding of span S is:

$$\mathbf{s} = \text{MaxPooling}_{x \in S}(\mathbf{x}) \oplus \mathbf{s}_{\text{width}}.$$

2.3.2 Prediction layer

We use the span embeddings to predict the entity type for each span S using a softmax classifier:

$$\hat{y} = \text{argmax}(\text{softmax}(\mathbf{W} \cdot \mathbf{s} + \mathbf{b})),$$

where $\mathbf{W}$ and $\mathbf{b}$ are learnable parameters.

2.3.3 Objective function

We define the loss function of the SciNER model as the negative log-likelihood loss:

$$\mathcal{L}_{\text{FT}} = -\log\left(\prod_{S \in \mathcal{S}} P(y|\hat{y})\right),$$

where $y \in \mathcal{Y}$ is the ground-truth entity type of span candidate S.

3 Proposed Tri-Train Framework

3.1 Overview

The framework has three steps (see Figure 1):

- Pre-training: Train transformer models by self-supervised tasks on large corpora;
- Pre-fine tuning: Use the pre-trained models as initialization. Construct a medium-size corpus relevant with the target domain. Optimize model parameters by two new tasks.
- Fine tuning: Use the pre-fine tuned models as initialization. Optimize model parameters with the labelled data on the target task.

In later sections, we focus on the second step. We will first introduce the two tasks for pre-fine tuning. And then we discuss the task settings and corpus construction.

3.2 Pre-Fine Tuning

It has two objectives. The first is to predict the tokens in masked entity candidates using span boundaries and token position as contexts of entity. The second is to predict the span boundaries using the entity candidates as contexts of boundaries. These two tasks are designed based on BERT and SpanBERT (i.e., predicting masked tokens). And they focus on learning the relationship between entity candidates (not random spans) and their boundaries, which is important for SciNER.

3.2.1 Entity-candidate mask prediction (MLM_{CAND})

Suppose we have a corpus $\mathcal{D}_{\text{PFT}}$ of unlabelled documents. (The construction will be introduced in Section 3.3.) We apply text mining methods (in Section 3.2.3), most of which are phrase mining and concept discovery algorithms, to find a set of entity candidates $\mathcal{C}$. An entity candidate is denoted by $c = (x_s, \ldots, x_e) \in \mathcal{C}$. Instead of randomly masking spans in the corpus, we mask spans that can be matched with the entity candidates. The model's encoder generates token-level representations $\mathbf{x}_i$ for token $x_i \in c$. The loss of entity-candidate mask prediction is written as:

$$\mathcal{L}_{\text{MLM\_CAND}} = -\log\left(\prod_{i=s}^{e} P(x_i|\mathbf{x}_i)\right).$$

3.2.2 Entity-candidate boundary prediction (ECB)

Besides predicting entity candidate spans with their boundary words, another task for learning the rela-

tionship between entity candidates and their boundaries is predicting the boundary words with the entity candidate as a context. The left and right boundary words are denoted as x_{s-1} and x_{e+1}. The model generates contextualized token embeddings of the two boundary words: $\mathbf{x}_{s-1}$ and $\mathbf{x}_{e+1}$ if available. The loss function is defined as:

$$\mathcal{L}_{\text{MLM\_ECB}} = -\log\left(P(x_{s-1}|\mathbf{x}_{s-1})\right) - \log\left(P(x_{e+1}|\mathbf{x}_{e+1})\right).$$

This training step sums losses from MLM_{CAND} and ECB for each entity candidate:

$$\mathcal{L}_{\text{PFT}} = \mathcal{L}_{\text{MLM\_CAND}} + \mathcal{L}_{\text{ECB}}.$$

3.2.3 Identifying entity candidates

There are three text mining methods to automatically identify entity candidates. We merge the set of entity candidates produced by each method. We use them as "labels" to match the new corpus $\mathcal{D}_{\text{PFT}}$ and support the MLM_{CAND} and ECB tasks.

Existing dictionaries We build a dictionary of entity candidates. First, it has all the labelled entities in the training data. Second, we add the entities on the MeSH and UMLS ontologies for BioNER. We use the dictionary to match with spans in the PFT corpus. To avoid noise, only when the span and its boundary words are all matched, we consider it as an entity candidate in $\mathcal{D}_{\text{PFT}}$. The dictionary is denoted by $\mathcal{C}_{\text{exist}}$.

Syntactic patterns Pattern-based methods such as SCHBase (Adar and Datta, 2015) aimed at discovering scientific entities (and their acronyms) with no need of human annotation. SCHBase utilized writing habits of scientific papers, such as using parentheses to link the acronyms of the scientific entities and their full name.

> "... In this work, we use Support Vector Machine (SVM) as a classifier ..."

Results of pattern-based methods are reliable. However, like most pattern-based methods, SCHBase makes low coverage: A great number of entities do not follow the patterns. The set of entity candidates discovered by patterns is denoted as $\mathcal{C}_{\text{pattern}}$.

Phrase mining We use AutoPhrase (Shang et al., 2018), a statistical learning-based phrase mining method to extract interesting phrases as entity candidates from the PFT corpus. This method calculates the features of phrase candidates such as frequency, concordance, completeness, and informativeness to evaluate their quality. The set of entity candidates discovered by phrase mining is denoted as $\mathcal{C}_{\text{phrase}}$.

Eventually, we have the set of entity candidates from $\mathcal{D}_{\text{PFT}}$: $\mathcal{C} = \mathcal{C}_{\text{exist}} \cup \mathcal{C}_{\text{pattern}} \cup \mathcal{C}_{\text{phrase}}$.

Table 1: Statistics of seven SciNER benchmarks.

	# Sentences			# Types
	train	dev	test	
BioNLP09	7,462	1,448	2,446	2
BioNLP11EPI	5,698	1,955	4,122	2
BioNLP11ID	2,496	721	1,961	4
BioNLP13CG	3,033	1,003	1,906	16
BioNLP13GE	2,499	2,737	3,391	2
BioNLP13PC	2,499	857	1,695	4
SCIERC	1,861	455	551	6

3.3 PFT corpus construction

When applied the proposed framework to scientific NER, as described in Figure 1 and the introduction section, we are interested in the questions such as what, where, and which. We have various options of using pre-fine-tuning for the task.

Sentence content selection Intuitively, if the PFT corpus is strongly related to the SciNER dataset, the domain relevance can support a good performance. For example, PubMed corpus may have more contextual information than Amazon Review corpus for the BioNER tasks; sentences about breast cancer research, a subfield of cancer research of biology and biomedicine, should be collected for pre-fine tuning when the target domain is on BRCA1, a gene relevant with breast cancer.

Sentence quality selection If a sentence had too few entity candidates, it would not support the learning step significantly. So when we collect the sentence to construct the corpus, we sort the sentences by the number of entity candidates. Then we take the top 30,000 sentences as $\mathcal{D}_{\text{PFT}}$.

4 Experiment

4.1 Data Sets

We describe seven SciNER benchmarks and three auxiliary corpora for pre-fine tuning.

4.1.1 Seven SciNER benchmarks

We conduct experiments on SCIERC (AI domain) and six BioNER benchmarks. Descriptions can be found in Appendix. Statistics are in Table 1.

Table 2: Our framework outperforms the state-of-the-art SciNER model. PFT stands for pre-fine tuning.

Dataset	SOTA (SpERT)			Best Pre-Training	PFT on MLM_{SPAN}			PFT on MLM_{CAND}		
	P	R	F1		P	R	F1	P	R	F1
BioNLP09	90.66	87.15	88.87	SciBERT	90.44	86.82	88.61	91.52	88.05	**89.77**
BioNLP11EPI	84.58	83.99	84.28	BioBERT	83.94	84.52	84.23	85.70	86.11	**85.90**
BioNLP11ID	84.40	83.30	83.35	SciBERT	85.20	83.18	84.18	85.91	85.06	**85.48**
BioNLP13CG	86.46	85.38	85.92	BioBERT	85.79	86.03	85.91	88.33	86.42	**87.37**
BioNLP13GE	73.49	78.94	76.12	BioBERT	73.15	78.82	76.12	77.95	83.37	**80.57**
BioNLP13PC	90.09	91.30	90.69	BioBERT	88.37	89.14	88.75	91.08	91.43	**91.25**
SciERC	67.36	67.72	67.53	SciBERT	68.36	67.77	68.08	69.91	68.25	**69.07**

4.1.2 Three auxiliary corpora for $\mathcal{D}_{\text{PFT}}$

Machine learning (ML) corpus. It is collected by FTS (Zha et al., 2018) which includes the title and abstract of 1.2 million computer science papers downloaded from DBLP and Semantic Scholar.

PubMed corpus. It has 140.9 million sentences from the abstracts of 15.5 million articles on MEDLINE (a life science database) (Lee et al., 2020).

Amazon review corpus. It has 233.1 million product reviews (ratings, text, helpfulness votes) ranging from 05/1996 to 10/2018 (Ni et al., 2019).

4.2 Competitive Models

SpERT (Eberts and Ulges, 2019). It is a span-based joint entity and relation extraction method. It leads the board using multi-task fine-tuning on the labelled data with *entity relation* information which is *NOT* used in our Tri-Train models. So it is not easy to win over this baseline. Our models perform pre-fine tuning and *single-task* fine-tuning.

PFT on MLM. Based on our framework, we implement two models. One applies random span mask prediction. The other uses our proposed task of predicting tokens in masked entity-candidates.

4.3 Implementation Details

All language models we use have a maximum 512 input token sequence and consist of a 12-layer transformer network with 12 attention heads and 768 word dimensions. For model fine tuning, we use Adam optimizer (Kingma and Ba, 2014) with learning rate of 5e-5. The maximum length of span candidates is 8. All these experiments are trained on one RTX 2080ti GPU whose memory is 11GB.

4.4 Experimental Results

In this section, we examine multiple aspects of the proposed Tri-Train framework. We first compare it with SOTA and see whether it is adaptive to multiple kinds of BERT models. Then we investigate the strategies of choosing auxiliary corpus, choosing sentences, identifying entity candidates, and choosing dictionary size.

4.4.1 Comparing with SOTA

Table 2 presents the performance comparisons. Compared with the state-of-the-art SpERT, PFT on MLM_{SPAN} does not improve much. So predicting random masked spans would not be able to bridge the pre-training model and the target NER task. Our proposed PFT on MLM_{CAND} outperforms SpERT on all the seven benchmarks. It improves precision, recall and F1 by +1.80%, +1.73%, and +1.85% (average) on the six BioNER benchmarks. It improves precision, recall and F1 by +1.55%, +0.48%, and +0.99% on SciERC. So, masking entity candidates can effectively extract useful information from the auxiliary corpus.

4.4.2 Adapting PFT on four BERTs

We applied the proposed framework (with the pre-fine tuning middle step) on four kinds of pre-trained BERTS. BERT and SpanBERT are pre-trained with general domain corpus (English Wikipedia and Books Corpus). SciBERT is pre-trained with CS and Bio corpora from Semantic Scholar. BioBERT is pre-trained with the general corpora and PubMed/PMC data. As shown in Table 3, we observe that all kinds of BERTs can benefit from the pre-fine tuning process. Compared with their original performance, BERT, SpanBERT, BioBERT, and SciBERT improve F1 score by +1.42%, +1.78%, +1.52%, and +1.02% (average) on six BioNER benchmarks. They also improve F1 by +1.0%, +0.6%, +0.6%, and +1.6% on SciERC, respectively.

Since SciBERT and BioBERT were pre-trained on scientific corpora, they demonstrate superior im-

Table 3: PFT stands for pre-fine tuning. It can consistently improves the performance on 7 SciNER benchmarks. The best performances were achieved on BioBERT and SciBERT with the proposed PFT.

	BERT							**SpanBERT**						
	No PFT			**With PFT**				**No PFT**			**With PFT**			
	P	R	F1	P	R	F1	Δ F1	P	R	F1	P	R	F1	Δ F1
BioNLP09	88.5	83.5	85.9	89.6	84.5	87.1	+1.2	87.6	85.2	86.4	88.6	86.0	**87.3**	+0.9
BioNLP11EPI	81.1	78.0	79.5	80.9	79.7	80.3	+0.8	78.6	81.0	79.6	81.8	80.6	**81.2**	+1.6
BioNLP11ID	86.6	80.6	83.5	88.2	80.4	**84.0**	+0.6	84.0	80.7	82.3	85.4	81.9	83.6	+1.3
BioNLP13CG	83.3	81.1	82.1	84.6	82.4	83.4	+1.3	82.3	80.8	81.6	83.9	83.3	**83.6**	+2.0
BioNLP13GE	70.0	70.7	70.4	73.3	74.6	73.9	+3.5	71.2	75.8	73.4	76.34	77.3	**76.8**	+3.4
BioNLP13PC	86.3	87.1	86.7	87.6	88.0	87.8	+1.1	86.7	87.6	87.1	88.3	88.8	**88.6**	+1.5
SciERC	67.8	65.2	66.5	68.7	66.3	**67.5**	+1.0	66.1	67.8	66.8	67.2	67.6	67.4	+0.6
	BioBERT							**SciBERT**						
	No PFT			**With PFT**				**No PFT**			**With PFT**			
	P	R	F1	P	R	F1	Δ F1	P	R	F1	P	R	F1	Δ F1
BioNLP09	89.1	88.3	88.7	92.0	86.9	89.4	+0.7	90.1	87.1	88.9	91.5	88.1	**89.8**	+0.9
BioNLP11EPI	85.6	84.6	85.1	85.7	86.1	**85.9**	+0.8	84.6	84.0	84.3	86.3	83.3	84.8	+0.5
BioNLP11ID	84.7	83.6	84.2	85.6	83.7	84.6	+0.4	84.4	83.3	83.4	85.9	85.1	**85.5**	+2.1
BioNLP13CG	87.8	85.6	86.7	88.3	86.4	**87.4**	+0.7	86.5	85.4	85.9	86.4	87.0	86.7	+0.8
BioNLP13GE	72.6	76.8	74.7	78.0	83.4	**80.6**	+5.9	73.5	78.9	76.1	76.0	79.6	77.7	+1.6
BioNLP13PC	90.4	91.1	90.7	91.1	91.4	**91.3**	+0.6	90.1	91.3	90.7	90.5	91.3	90.9	+0.2
SciERC	68.4	67.8	68.1	70.0	67.5	68.7	+0.6	67.4	67.7	67.5	69.9	68.3	**69.1**	+1.6

Table 4: The relevance of domains of auxiliary corpus with training data matters in pre-fine tuning.

	Amazon (top 30K)			**PubMed (top 30K)**			**ML (top 30K)**			**bottom 30K**		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
BioNLP09	86.7	81.2	83.9	91.5	88.1	**89.8**	88.9	86.7	87.8	90.8	87.1	88.9
BioNLP11EPI	84.6	84.9	84.7	85.7	86.1	**85.9**	86.4	83.5	84.9	85.5	85.2	85.4
BioNLP11ID	83.9	83.4	83.7	85.9	85.1	**85.5**	85.4	85.1	85.2	85.4	84.7	85.1
BioNLP13CG	86.2	85.3	85.8	88.3	86.4	**87.4**	88.2	84.9	86.5	88.2	86.3	87.2
BioNLP13GE	71.2	71.2	71.2	78.0	83.4	**80.6**	73.2	77.0	75.1	76.8	81.8	79.3
BioNLP13PC	86.0	86.1	86.1	91.1	91.4	**91.3**	90.7	89.8	90.3	90.6	90.0	90.3
SciERC	66.5	67.4	67.0	68.5	67.1	67.8	69.9	68.3	**69.1**	68.7	68.3	68.5

provements than BERT and SpanBERT. Enhancing data relevance is an effective way to achieve good performance. Besides, SciBERT presents the best performance on two of the BioNER benchmarks, which indicates that SciBERT obtained knowledge in the biological domain.

4.4.3 Choosing auxiliary corpus

Intuitively, highly relevant auxiliary corpus with the labelled training data would help the model obtain useful knowledge during the pre-fine tuning (PFT) process. We compare model performances on using three different domain corpora for PFT. They are Machine Learning (AI domain), PubMed (biology domain), and Amazon (shopping domain). Table 4 presents the performance of using different auxiliary corpora on the benchmarks. We observe that model performances are correlated with the domain relevance between auxiliary corpus and target corpus. First, using PubMed as auxiliary corpus performs better than using other domain corpus on all BioNER benchmarks. Similarly, using ML corpus achieves the best performance on SciERC, which improves precision, recall, and F1 by +1.4%, +1.2%, and +1.3%, respectively. Using Amazon corpus cannot provide satisfactory performances on all seven benchmarks. It indicates that irrelevant domain corpus may hurt the model training.

4.4.4 Choosing sentences

In Table 4, we know top 30K sentences in PubMed helps BioNER via pre-fine tuning; top 30K sentences in ML helps on SciERC. We also show the performance when choose the bottom 30K

Table 5: The merged set of entity candidates (combining existing dictionaries, pattern mining, and phrase mining results) perform the best to support the MLM_{CAND} and ECB tasks.

	$\mathcal{C}_{\text{exist}}$ **only**			$\mathcal{C}_{\text{pattern}}$ **only**			$\mathcal{C}_{\text{phrase}}$ **only**			$\mathcal{C}$ **(merged)**		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
BioNLP09	90.5	86.2	88.3	91.9	87.3	89.6	90.2	88.9	89.6	91.5	88.1	**89.8**
BioNLP11EPI	84.9	84.1	84.5	85.7	84.5	85.1	85.3	85.2	85.2	85.7	86.1	**85.9**
BioNLP11ID	85.2	83.4	84.3	86.3	84.4	85.4	85.1	84.8	85.0	85.9	85.1	**85.5**
BioNLP13CG	87.4	85.7	86.5	88.6	84.3	86.4	87.4	87.0	87.2	88.3	86.4	**87.4**
BioNLP13GE	74.5	77.7	76.1	79.2	80.0	79.6	77.3	81.5	79.4	78.0	83.4	**80.6**
BioNLP13PC	90.3	90.8	90.6	90.9	90.0	90.9	90.5	91.9	91.2	91.1	91.4	**91.3**
SciERC	67.4	67.9	67.7	69.7	67.6	68.6	68.6	68.3	68.4	69.9	68.3	**69.1**

sentences for the best matched auxiliary corpus. The performances are still better than using irrelevant domain corpus; however, they are consistently worse than using the top 30K sentences. This observation demonstrate that choosing highly relevant sentences is as important as choosing a corpus.

4.4.5 Choosing methods to identify entity candidates

Table 5 presents interesting results. If we identify only the entity candidates that were in some existing dictionaries ($\mathcal{C}_{\text{exist}}$), our models make tiny improvements on four of the seven benchmarks. This is because the size of dictionaries is limited. Few entity candidates were identified in auxiliary corpus. This leads to insufficient pre-fine tuning.

First, both pattern-based method (SCHBase) and phrase mining method (AutoPhrase) demonstrate their effectiveness on labelling entity candidates and using them to predict tokens in masked spans for pre-fine tuning. The effectiveness can be observed on all the BioNER and SciNER benchmarks. Compared with the phrase mining method, the pattern-based method makes a higher precision because entity candidates identified by textual patterns are more reliable than noun or verbal phrases. The pattern-based method improves precision by +1.31% (average) on all benchmarks. In contrast, the phrase mining method improves recall by +1.35% compared with the pattern-based method. The pattern-based method and phrase mining method are complementary for masking entity candidates. By merging the set of identified entity candidates and matching the auxiliary corpus with their names, we try to maximize the labelling accuracy. Table 5 shows that our merged entity candidate dictionaries can support PFT to achieve the state-of-the-art performance in terms of F1 score on all benchmarks.

4.4.6 All entity candidates or frequent only

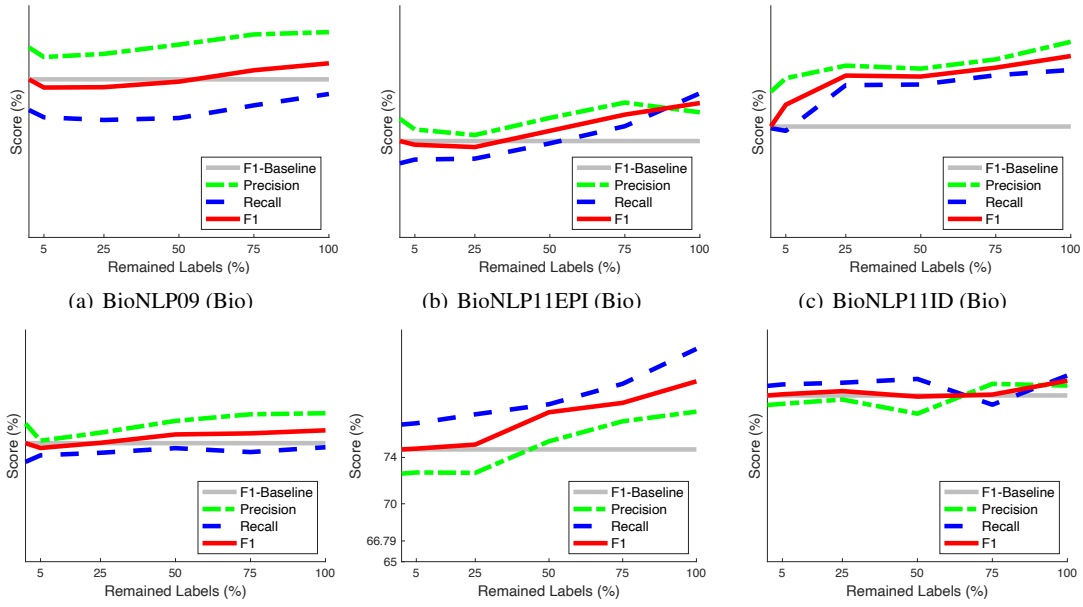
In proposed method of building dictionary, the size of dictionary is usually proportional to the scale of auxiliary text data. To obtain a big dictionary, we expect to have a large auxiliary corpus. However, the frequency of entities in the dictionary forms a long tail – very few entities have very high frequency while most entities are infrequent. Does the long tail, or only the frequent head, improve or hurt the pre-fine tuning? Figures 2 and 3 present results to answer this question. Briefly, the answer is the long tail matters – it is useful to have a large dictionary from a large-scale auxiliary corpus.

We sort the entities in the merged dictionary from the highest density to the lowest. We use the top $l\%$ frequent entities (called "remained labels" on the horizontal axis of the figures) in the dictionaries for pre-fine-tuning (l is from 5 to 100). We observed that: (1) In all benchmarks, 100% achieve the best F1-score. (2) In some benchmarks, when the dictionary size is smaller than 50%, pre-fine tuning hurts the performance on F1 score. The reason is that even though compared with corpus used in pre-training, auxiliary corpus is a small, it still needs amount of diverse information to ensure the ability of generalization.

5 Related Work

5.1 SciNER: Scientific Entity Recognition

NER is typically cast as a sequence labeling problem by integrating LSTMs, CRF, and language models (Lample et al., 2016; Ma and Hovy, 2016; Liu et al., 2018). Another idea is to generate span candidates and predict their type. Span-based models have been proposed with multi-task learning strategies (Luan et al., 2018, 2019). The multiple tasks include concept recognition, relation extraction, and co-reference resolution. With the popular-



(a) BioNLP09 (Bio) (b) BioNLP11EPI (Bio) (c) BioNLP11ID (Bio)

(d) BioNLP13CG (Bio) (e) BioNLP13GE (Bio) (f) BioNLP13PC (Bio)

Figure 2: Overall, a bigger size of dictionaries consistently improves the performance on six BioNER benchmarks.

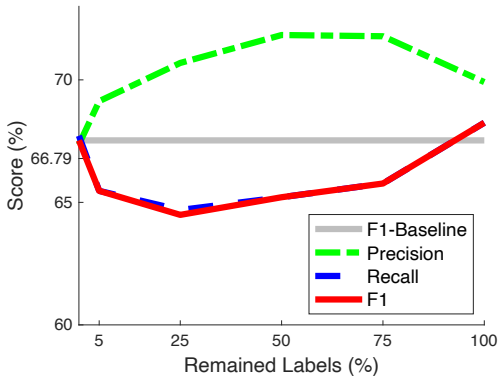


Figure 3: On SCIERC, leaving half of the entity candidates un-masked in pre-fine tuning may cause a more desired precision but a much lower recall and F1 score. We still recommend to fully use the dictionaries.

ity of self-supervised learning, pre-trained models are widely used in NER (Peters et al., 2018; Devlin et al., 2019). They improved the performance with contextual information from massive data.

5.2 Contextualized Language Representation

ELMo (Peters et al., 2018) proposed bi-directional LSTMs based language models. OpenAI proposed GPT (Radford et al.) used a multi-layer transformer as decoder to predict text sequence one-by-one.

BERT (Devlin et al., 2019) employs a bidirectional Transformer encoder (Vaswani et al., 2017) to fuse both the left and the right contexts. It includes two novel pre-training tasks: masked language model (MLM) and next sentence prediction (NSP). To augment the semantic information in corpus, pre-training with some variants of a masked language model objective is used in many BERT-based models. These models improve the performance by span prediction (Joshi et al., 2020), including entity embeddings (Zhang et al., 2019; Sun et al., 2019), autoregressive pretraining (Yang et al., 2019; Dai et al., 2018), and sentence ordering objective (Wang et al., 2019). Influenced by BERT's great success on multiple natural language processing (NLU) tasks, researchers proposed BERT-like models in scientific domain (Beltagy et al., 2019; Lee et al., 2020). They still need related corpora and computing resources for training.

6 Conclusion

In this work, we proposed a novel framework to introduce a "pre-fine tuning" step between pre-training and fine-tuning. It constructed a corpus by selecting sentences from unlabeled documents that were the most relevant with the labelled training data. Instead of predicting tokens in random spans, the pre-fine tuning task was to predict tokens in entity candidates identified. Pre-fine tuning was automatic and light-weight because the corpus size could be much smaller than pre-training data to achieve a better performance. Experiments on seven benchmarks demonstrate the effectiveness. We further investigated settings of pre-fine tuning.

Acknowledgements

This work is supported in part by NSF IIS-1849816, CCF-1901059, the US Army Research Office (W911NF-17-1-0448) and the US Defense Advanced Research Projects Agency (DARPA W911NF-17-C-0094).

References

Eytan Adar and Srayan Datta. 2015. Building a scientific concept hierarchy database (schbase). In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 606–615.

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)*, pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.

Zihang Dai, Zhilin Yang, Yiming Yang, William W Cohen, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. 2018. Transformer-xl: Language modeling with longer-term dependency.

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)*, pages 4171–4186.

Markus Eberts and Adrian Ulges. 2019. Span-based joint entity and relation extraction with transformer pre-training. *arXiv preprint arXiv:1909.07755*.

Tianwen Jiang, Qingkai Zeng, Tong Zhao, Bing Qin, Ting Liu, Nitesh Chawla, and Meng Jiang. 2020. Biomedical knowledge graphs construction from conditional statements. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*.

Tianwen Jiang, Tong Zhao, Bing Qin, Ting Liu, Nitesh Chawla, and Meng Jiang. 2019. Multi-input multi-output sequence labeling for joint extraction of fact and condition tuples from 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)*, pages 302–312.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Jin-Dong Kim, Tomoko Ohta, Sampo Pyysalo, Yoshinobu Kano, and Jun'ichi Tsujii. 2009. Overview of bionlp'09 shared task on event extraction. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task*, pages 1–9. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Liyuan Liu, Jingbo Shang, Xiang Ren, Frank Fangzheng Xu, Huan Gui, Jian Peng, and Jiawei Han. 2018. Empower sequence labeling with task-aware neural language model. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Yi Luan, Dave Wadden, Luheng He, Amy Shah, Mari Ostendorf, and Hannaneh Hajishirzi. 2019. A general framework for information extraction using dynamic span graphs. *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)*.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. *arXiv preprint arXiv:1603.01354*.

Claire Nédellec, Robert Bossy, Jin-Dong Kim, Jung-Jae Kim, Tomoko Ohta, Sampo Pyysalo, and Pierre Zweigenbaum. 2013. Overview of bionlp shared task 2013. In *Proceedings of the BioNLP Shared Task 2013 Workshop*, pages 1–7.

Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. 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)*, pages 188–197.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Sampo Pyysalo, Tomoko Ohta, Rafal Rak, Dan Sullivan, Chunhong Mao, Chunxia Wang, Bruno Sobral, Jun'ichi Tsujii, and Sophia Ananiadou. 2012. Overview of the id, epi and rel tasks of bionlp shared task 2011. In *BMC bioinformatics*, volume 13, page S2. BioMed Central.

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

Jingbo Shang, Jialu Liu, Meng Jiang, Xiang Ren, Clare R Voss, and Jiawei Han. 2018. Automated phrase mining from massive text corpora. *IEEE Transactions on Knowledge and Data Engineering*, 30(10):1825–1837.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. 2019. Ernie 2.0: A continual pre-training framework for language understanding. *arXiv preprint arXiv:1907.12412*.

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*, pages 5998–6008.

Wei Wang, Bin Bi, Ming Yan, Chen Wu, Zuyi Bao, Liwei Peng, and Luo Si. 2019. Structbert: Incorporating language structures into pre-training for deep language understanding. *arXiv preprint arXiv:1908.04577*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5754–5764.

Wenhao Yu, Wei Peng, Yu Shu, Qingkai Zeng, and Meng Jiang. 2020. Experimental evidence extraction system in data science with hybrid table features and ensemble learning. In *Proceedings of The Web Conference 2020*, pages 951–961.

Qingkai Zeng, Mengxia Yu, Wenhao Yu, Jinjun Xiong, Yiyu Shi, and Meng Jiang. 2019. Faceted hierarchy: A new graph type to organize scientific concepts and a construction method. In *Proceedings of the Thirteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-13)*, pages 140–150.

Hanwen Zha, Jiaming Shen, Keqian Li, Warren Greiff, Michelle T Vanni, Jiawei Han, and Xifeng Yan. 2018. Fts: Faceted taxonomy construction and search for scientific publications.

Yunyi Zhang, Jiaming Shen, Jingbo Shang, and Jiawei Han. 2020. Empower entity set expansion via language model probing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8151–8160, Online. Association for Computational Linguistics.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. Ernie: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451.