Low Resource Causal Event Detection from Biomedical Literature

Zhengzhong Liang, Enrique Noriega-Atala, Clayton Morrison and Mihai Surdeanu
The University of Arizona, Tucson, AZ
{zhengzhongliang, enoriega, claytonm, msurdeanu}@email.arizona.edu

Abstract

Recognizing causal precedence relations among the chemical interactions in biomedical literature is crucial to understanding the underlying biological mechanisms. However, detecting such causal relation can be hard because: (1) many times, such causal relations among events are not *explicitly* expressed by certain phrases but *implicitly* implied by very diverse expressions in the text, and (2) annotating such causal relation detection datasets requires considerable expert knowledge and effort. In this paper, we propose a strategy to address both challenges by training neural models with in-domain pre-training and knowledge distillation. We show that, by using very limited amount of labeled data, and sufficient amount of unlabeled data, the neural models outperform previous baselines on the causal precedence detection task, and are ten times faster at inference compared to the BERT base model.

1 Introduction

Since 2011, more than one million new articles are added to PubMed every year (Vardakas et al., 2015). The growth rate of newly published articles makes it hard to keep up with the important discoveries just by reading them. Therefore, tremendous efforts have been made to automate knowledge discovery from biomedical papers by extracting the biochemical events described in the literature (Kim et al., 2009, 2012; Nédellec et al., 2013).

In addition to the extraction of the biochemical events, there are existing efforts to detect the causal relationships among them (Mihăilă et al., 2013; Hahn-Powell et al., 2016), i.e., whether the occurrence of one event necessarily leads to the occurrence of another event. Knowing the causal precedence order of the events helps to describe more accurately the underlying mechanisms of biological processes described on the scientific literature. However, annotating such causal event pairs requires significant domain expertise and effort (Hahn-Powell et al., 2016).

In this work, we investigate multiple strategies for improving the detection of causal precedence relations within biochemical events. The contributions of this paper are the following:

(1) We propose and investigate multiple neural architectures for detection of causal precedence among biochemical interactions trained with a few hundred annotated training examples and numerous weakly-supervised training examples.

(2) We analyze the impact of in-domain pre-training and distillation on the performance of the proposed architectures, and conclude that several compact BERT architectures can benefit from in-domain pre-training, and can potentially benefit from further distillation.

(3) Lastly, we study a hybrid methodology that combines neural models with the traditional rule/feature-based methods in a sieve-based framework, and observe that well-trained neural models can largely replace the rule/feature-based methods and do not benefit from the sieve framework.[1]

2 Related Work

The detection of causal precedence among chemical interactions from text is a long-standing problem. Early methods include rule-based approaches (Khoo et al., 2000) and machine learning-based approaches (Girju, 2003; Blanco et al., 2008; Akkasi and Moens, 2021). Other work (Sorgente et al., 2013; Hahn-Powell et al., 2016; Dasgupta et al., 2018) have also explored the combination of rule-based methods, machine learning-based or neural-based methods.

Recently, large pre-trained language models (LLM) have increased the state-of-the-art performance of many natural language processing tasks

[1] The code and data can be found at: https://github.com/clulab/releases/tree/master/acl2022-bionlp-causal

Proceedings of the BioNLP 2022 workshop, Dublin, Ireland, pages 252–263
May 26, 2022. ©2022 Association for Computational Linguistics

(Devlin et al., 2019; Liu et al., 2019b; Raffel et al., 2020). However, such LLM models require enormous computational resources, making it hard to deploy them in many applications. One popular approach to reduce the memory footprint of LLMs is distillation (Sanh et al., 2019; Jiao et al., 2020; Wu et al., 2020; Wang et al., 2020). Distillation trains a relatively small model to imitate the behavior of a larger model, such as a LLM, trading off performance for a significant reduction in the amount of parameters.

Tang et al. (2019b) shows that it is possible to distill BERT-large to a **task-specific** compact LSTM model with approximately $\frac{1}{300}$ of the model's original parameters while maintaining a comparable performance. Wasserblat et al. (2020) and Adhikari et al. (2020) investigated whether the performance of the distilled model depends on the nature of the task and the size of the student model. Turc et al. (2019) found that the general domain pre-training of the compact model is essential and helpful to the distillation on the downstream tasks. In addition, various data augmentation techniques are proposed to improve the distillation process with very limited labeled training data (Mukherjee and Awadallah, 2019; Tang et al., 2019a; Melas-Kyriazi et al., 2019). Finally, several works explored whether cross-task distillation helps the compact models to learn more robust representations (Liu et al., 2019a; Pan et al., 2021). To the best of our knowledge, this work is the first to investigate model distillation specifically for the task of causal precedence detection in the biomedical domain.

3 Dataset

We use of a dataset of causal precedence annotations of biochemical interactions (Hahn-Powell et al., 2016). The dataset contains 858 interaction pairs. Each pair is annotated with one of three classes: *E1 precedes E2*, *E2 precedes E1*, and *no precedence relationship*, with 109, 27 and 722 instances, respectively. Table 1 contains a few examples of the annotations.

Working with this dataset presents multiple challenges. Firstly, it's small, with only total of 858 annotated examples. The scarcity of training data is a challenge for a model with a relatively large number of parameters to pick up training signal from the data. Second, prediction of some examples requires more than the shallow understanding of linguistic knowledge (i.e., understanding the phrases such as "leading to"), and also requires understanding the underlying mechanistic process described in the phrase. For example, in the last row of Table 1, the model needs to understand "FoxO1 can bind to ATG7" and "FoxO1 and ATG7 complex" are referring to the same event, so that there is no precedence between them. Finally, we are aiming at obtaining a compact model that can be efficiently deployed without a GPU and with high processing speed.

4 Approach

4.1 Neural-based Approaches

We propose two neural-based architectures: A BiLSTM (Graves and Schmidhuber, 2005) and a fine-tuned BERT model (Devlin et al., 2019).

Both architectures take as input the text span containing both biochemical interactions (events). The text span is encoded as:

```
[E1 tokens] + [SEP] + ...  +
      [SEP] + [E2 tokens]
```

Where `...` represents the text between both events. If E1 is adjacent to E2 (i.e., there is no text between them), the input sequence becomes:

```
[E1 tokens] + [SEP] + [E2 tokens]
```

How much context to include in the input is a design choice. An alternative design is to include more text in the input sequence, such as the text preceding `E1` and the text following `E2`. However, the model might fail to learn to concentrate on the most essential part for the causal relation detection when the context is too long, especially considering there are very limited labeled data in our task. Therefore we did not include the context preceding `E1` and following `E2` in our current model. We leave the impact of such design choices to future work.

BiLSTM

We use a single layer BiLSTM with input dimension of 100 and hidden dimension $h \in \{200, 700, 750\}$. The output of the BiLSTM model H is a tensor of size $l \times 2h$, where l is the number of tokens in the input and h is the hidden dimension of the BiLSTM. The output vector tensor H is then max-pooled over the sequence, creating a vector H' with size $2h$. The pooled hidden representation H' is then passed to a 2-layer MLP to predict the class of the input sequence.

Text spans with a pair of biochemical interactions	Label	Explanation
IKKalpha then phosphorylates the C-terminal region of p100 *leading to* subsequent processing of the p100 and RelB complex into p52 and RelB and its translocation into the nucleus	E1 precedes E2	The expression "leading to" suggests the precedence relationship.
Given that an oxidant inhibits the catalytic action of Cdc25 on wt Ras that is an enhancement of the wt Ras bound GDP, the oxidant evidently targets the ternary complex.	E2 precedes E1	The expression "enhancement of" indicates the precedence relationship.
We next studied the effect of these growth factors on the tyrosine phosphorylation of Gab1 and its binding to SHP-2. EGF, *but not* IGF or PDGF, led to both increased tyrosine phosphorylation of Gab1 and binding to SHP-2, suggesting a selective effect of EGF on Ras and MAPK activation mediated by Gab1 and SHP-2.	No precedence	The expression "but not" indicates there is no precedence relationship.
FoxO1 can bind to ATG7, which is an important regulator in autophagosome expansion, and the FoxO1 and ATG7 complex may impact autophagy in human colon cancer HCT116 cells or in HeLa cells.	No precedence	The two events are equivalent although the expressions are a little different.

Table 1: Examples of relations in the causal precedence dataset. Each example contains a span of text from either one or two adjacent sentences. The text contains a pair of biochemical interactions. The first interaction (E1) is colored in red and the second (E2) in blue. The boundary of each event is extracted by REACH. The classification problem is to predict whether there is an existence of a causal precedence relations between E1 and E2.

BERT

For BERT, in addition to the common encoding, we prepend a `[CLS]` token to the input sequence. Then, the sequence is passed through BERT, generating a list of embeddings (with size h) of all l input tokens. Then a 2-layer MLP is placed on top of the embedding of the `[CLS]` token to obtain the final prediction result.[2]

We evaluate 4 pre-trained variants of BERT:

BERT-base: The original BERT-base model released by Google. It contains approximately 110M parameters. In the experiment we use the `bert-base-uncased` model provided by the huggingface library.[3]

BioBERT-base: (Lee et al., 2020) This model has the same amount of parameters as BERT-base. It was further pre-trained on PubMed papers. We use the `BioBERT-base-cased V1.1` in our experiments.[4]

BERT-L8H128A2: (Turc et al., 2019) A compact BERT model pre-trained on the same corpus as BERT-base, but with only 8 layers, hidden size of 128 and 2 attention heads. It has 5.5M parameters.

BERT-L4H256A4: (Turc et al., 2019) Similar to BERT-L8H128A2 but with only 4 layers, hidden size of 256 and 4 attention heads. It contains 11M parameters.

4.2 Pre-training

Previous works have shown that both general-domain pre-training (Turc et al., 2019) and in-domain pre-training (Lee et al., 2020) can improve the model's performance on the down-stream tasks. Gururangan et al. (2020) shows that even the pre-training in a non-target but similar-to-target domain can help with the later fine-tuning. In this work we investigate whether in-domain pre-training can help with the compact BiLSTM or BERT classifiers.

Pre-training Corpus

We use REACH, a bio-medical domain information extraction tool (Valenzuela-Escárcega et al., 2018), to extract 10,000 biomedical papers from PMC Open Access.[5] The corpus is composed of papers that contain biochemical events, such as *phosphorylation*, *methylation* and a few others.[6] We cleaned the text (e.g., remove the sentences that are too short, usually citations) and split the sentences using the NLTK toolkit (Bird et al., 2009). The total number of sentences is 1.5M. We use the sentences of 9,000 papers as the training set and the sentences of the remaining 1,000 papers as the eval-

[2] We use the `BertForSequenceClassification` function from the huggingface library.

[3] `https://huggingface.co/transformers/v3.0.2/model_doc/bert.html?highlight=bertforsequenceclassification`

[4] `https://github.com/dmis-lab/biobert`

[5] `https://www.ncbi.nlm.nih.gov/pmc/tools/openftlist/`

[6] For the complete list of the keywords we use to retrieve the papers, please see Appendix A.

uation set. The evaluation set here is solely used to determine when to stop training the language model in the pre-training stage and is not used for the evaluation of the causal relation detection task. For the rest of this work, we will refer to this corpus as PMC-10000.

BiLSTM Pre-training

We investigate the impact of pre-training to a BiLSTM model in two ways. First, we evaluate whether it is helpful to train a skip-gram model (Mikolov et al., 2013) on PMC-10000 to use as input to the LSTM. We also evaluate whether it is helpful to pre-train the model using a language modeling task. We employ a similar protocol to (Mousa and Schuller, 2017): Given an input sequence of tokens [t_1, t_2, ..., t_l], the forward LSTM is taught to predict the tokens [t_2, t_3, ..., t_l] and the backward LSTM is taught to predict the tokens [t_{l-1}, t_2, ..., t_1].

BERT Pre-training

We train the model using the standard Masked Language Modeling (MLM) on PMC-10000 with whole-word masking but without Next Sentence Prediction (NSP) task. The length of each sentence is limited to 50 (after applying the sub-word tokenizer). The mask probability is set to 0.15, as in (Devlin et al., 2019). The model is trained with a batch size of 64 using Adam optimizer with the learning rate of 5e-5 for 12 epochs (for a total of approximately 284K optimization steps).

4.3 Distillation

Although the large language models such as BERT and BioBERT have shown strong performance on various tasks, they consume a lot of computation resources and could have a high inference latency when deployed without a GPU. Such a high inference latency is undesirable when thousands and millions of biomedical publications need to be processed. Therefore we are motivated to develop a compact model that can be deployed with a low inference latency even without a GPU.

However, compact models usually could not reach a comparable performance as large pre-trained language models. Therefore we seek to use knowledge distillation to transfer the knowledge of a large language model into compact neural models.

We first fine-tune BioBERT-base with the causal precedence dataset. For each labeled event pair, the model is trained to predict the precedence relationship using a cross-entropy loss. The fine-tuned model will serve as the teacher during the distillation process. We train several BiLSTM student models and compact BERT (BERT-L8H128A2 and BERT-L4H256A4) student models. Following (Tang et al., 2019b), the loss between the teacher and the student is formulated as the Mean Square Error (MSE) loss between the logits of the teacher $z^{(B)}$ and the logits of the student $z^{(S)}$.

$$L = ||z^{(B)} - z^{(S)}||_2^2$$

A distillation process may suffer from a small labeled training set, and data augmentation techniques are frequently used to obtain numerous unlabeled data (Tang et al., 2019b). Similarly, we use both the labeled data D_l and unlabeled data D_u for distillation. However, we don't use data augmentation to obtain D_u, but generate D_u by processing 88,000 PubMed articles with REACH (Valenzuela-Escárcega et al., 2018) and extract 20,001 unlabeled event pairs.

4.4 Baselines

We consider a rule-based heuristic and a feature-based classifier, both of which are proposed and elaborated in (Hahn-Powell et al., 2016). Here we briefly introduce these two baselines, and more details can be found in (Hahn-Powell et al., 2016).

Rule-based heuristic

The event pair causal precedence relation is predicted using a few hand-written deterministic rules. There are three types of rules: **intra-sentence** rules, **inter-sentence** relations and **verbal-tense**.[7]

Feature-based classifier

Event pairs are transformed into a feature vector representation using hand-crafted rules. The encoded pairs are used to train a SVM. Some of the features include the interaction type (i.e. "phosphorylation", "ubiquitination"), the text between the events, coreference resolution, etc.

5 Results

5.1 The Impact of Pre-training

Table 2 shows the impact of in-domain pre-training (as detailed in section 4). For each row, we run experiments with five different random seeds and

[7] A slightly more detailed description can be found in Appendix B.

Model	Dev P.	Dev R.	Dev. F1	Test P.	Test R.	Test F1
BiLSTM-small-w2v-VO1	0.489 (0.028)	0.481 (0.021)	0.484 (0.009)	0.534 (0.025)	0.325 (0.033)	0.403 (0.027)
BiLSTM-small-w2v-VO2	0.326 (0.028)	0.646 (0.067)	0.430 (0.015)	0.404 (0.055)	0.554 (0.075)	0.459 (0.014)
BiLSTM-large-w2v-VO1	0.447 (0.034)	0.545 (0.039)	0.489 (0.014)	0.528 (0.027)	0.400 (0.041)	0.454 (0.026)
BiLSTM-large-w2v-VO2	0.317 (0.014)	0.683 (0.014)	0.433 (0.015)	0.384 (0.008)	0.586 (0.037)	**0.464** (0.012)
BiLSTM-large-w2v-ID-VO1	0.496 (0.040)	0.628 (0.032)	**0.552** (0.021)	0.471 (0.023)	0.421 (0.060)	0.442 (0.033)
BiLSTM-large-w2v-ID-VO2	0.418 (0.029)	0.715 (0.039)	0.526 (0.014)	0.357 (0.039)	0.523 (0.036)	0.422 (0.027)
BiLSTM-large-WP	0.404 (0.024)	0.609 (0.073)	0.484 (0.032)	0.413 (0.038)	0.426 (0.067)	0.418 (0.050)
BERT-L8H128A2	0.340 (0.022)	0.655 (0.047)	0.446 (0.018)	0.407 (0.040)	0.523 (0.049)	0.456 (0.035)
BERT-L8H128A2-Bio	0.375 (0.015)	0.650 (0.031)	0.475 (0.005)	0.491 (0.043)	0.557 (0.064)	0.518 (0.030)
BERT-L8H128A2-Bio-RV	0.364 (0.020)	0.709 (0.042)	0.481 (0.022)	0.449 (0.031)	0.630 (0.017)	0.524 (0.021)
BERT-L4H256A4	0.351 (0.006)	0.561 (0.031)	0.431 (0.010)	0.499 (0.045)	0.485 (0.023)	0.491 (0.027)
BERT-L4H256A4-Bio	0.408 (0.029)	0.622 (0.051)	0.491 (0.015)	0.554 (0.048)	0.549 (0.041)	**0.548** (0.007)
BERT-L4H256A4-Bio-RV	0.420 (0.036)	0.612 (0.030)	**0.497** (0.026)	0.557 (0.065)	0.525 (0.035)	0.537 (0.026)
BERT	0.420 (0.037)	0.605 (0.053)	0.492 (0.013)	0.537 (0.045)	0.512 (0.057)	0.520 (0.030)
BioBERT	0.437 (0.031)	0.705 (0.055)	**0.537** (0.019)	0.547 (0.079)	0.539 (0.072)	**0.535** (0.023)

Table 2: The impact of in-domain pre-training for the BiLSTM and BERT architectures. w2v and w2v-ID are the general-domain/in-domain Word2Vec embeddings. VO1 and VO2 are the two options to build the LSTM vocabulary. WP is the LSTM pre-trained by the language modeling task using WordPiece tokenizer. RV is the reduced vocabulary for BERT. All of these models are discussed throughout Section 5.1.

report mean and standard deviation of the different metrics. Each experiment is a 5-fold cross validation, using 64% of the dataset for training, 16% for validation and 20% for testing. Each model is trained for 40 epochs. The validation F1 is used for early stopping using a patience counter of 5. We used Adam optimizer (Kingma and Ba, 2015). For the LSTM models, we experimented with different hidden sizes, word embedding options and vocabulary options (explained later in the text), and the learning rate is set to 1e-4. For all BERT-based models the learning rate is set to 2e-5.

All of the models in Table 2 contain less than 12M parameters, with the exception of *BERT* and *BioBERT*, which have approximately 110M parameters. Table 3 shows a detailed presentation of the model's size and inference time. Results show that pre-training the compact BERT models on PMC-10000 boosts the models' performance, obtaining test F1s even slightly higher than the large BioBERT. On the other hand, the pre-training of LSTM models using PMC-10000 does not help.

BERT-based models

Rows *BERT* and *BioBERT* in table 2 show the performance of BERT models fine-tuned for the causal precedence task. BioBERT showed both higher F1 scores on dev and test sets, and a lower discrepancy between the dev and test scores compared with other compact models. Since this is a small dataset, we hypothesize that the in-domain pre-training of BioBERT boosts the performance of the fine-tuned model compared to the open domain BERT.

W2V embeddings

For the LSTM models, the *w2v* embeddings were trained using *Word2Vec* over 1 million PubMed papers as introduced in (Hahn-Powell et al., 2016), whereas *w2v-ID* embeddings were trained using the same method but on the PMC-10000 corpus. Both w2v and w2v-ID were trained with biomedical papers, but w2v-ID's corpus is smaller and focused on narrower topics. Results show that the *w2v-ID* embeddings trained on PMC-10000 largely increase the dev scores of the models, which doesn't transfer to the test scores, suggesting the models are overfitting to the dev examples. We suspect that the reason for this is that the PMC-10000 corpus is too small, and not diverse enough for the *w2v-ID* embeddings to learn general and robust representations.

The vocabulary of LSTM models

We found that the composition of the vocabulary used by the LSTM models can impact their performance. We tried two different vocabularies: `VO1`, which contains any word that appears in the training set; and `VO2`, which contains words that occur at least twice in the training set. To deal with out-of-vocabulary words (OOV), `VO1` uses the `unk` vector as trained by Word2Vec (not fine-tuned on our causal detection dataset) whereas in `VO2` the `unk` vector is further fine-tuned in our causal detection dataset. The trade-off is that fine-tuning the `unk` embedding should yield a more accurate representation for it, but it also reduces the vocabulary size. We found that using `VO2` works better than `VO1` for *w2v* but for *w2v-ID*.[8] This is likely due to the fact that *w2v-ID* already obtains a fairly accurate `unk` embedding through in-domain pre-training, so that the model benefits more from a larger vocabulary than a fine-tuned `unk` embedding. On the other hand, for *w2v*, the `unk` embedding is not good enough without fine-tuning.

LSTM sizes

Since the size of the model may affect LSTM architecture's performance on some tasks (Adhikari et al., 2020), we investigate the impact of the model size. Results show that for our task the larger LSTM works slightly better than the smaller LSTM but the difference is negligible. Note that the BiLSTM-small is only about 1/8 of BiLSTM-large (size comparison in Table 3).

In-domain BiLSTM language modeling

LSTM-large-WP is trained with the language modeling task introduced in section 4 using the PMC-10000 corpus. However, if we use the regular vocabulary, its size and the embedding layer's size would be large. We reduce both sizes using two strategies: (1) we use the same WordPiece tokenization algorithm that BERT uses; (2) to further reduce the number of embedding vectors, we keep only the top 10,000 tokens by corpus frequency in PMC-10000 and use only 10,000 token pieces. In perspective, the WordPiece tokenization model of *bert-base-uncased* has 30,522 tokens. With this approach the vocabulary size and the number of embeddings is reduced by $\frac{2}{3}$.

Our results show that pre-training the LSTM model using in-domain language modeling task does not help with the fine-tuning of our causal precedence detection task. The pre-trained LSTM has a relatively large gap between the dev F1 (0.484) and test F1 (0.418), and the test F1 is even lower than using the *w2v-ID* embeddings. This is probably because the LSTM is not pre-trained on the general domain corpus (like BERT), therefore it doesn't benefit from any transfer learning signal.

In-domain pre-training of BERT

BERT-L8H128A2 and BERT-L4H256A4 are pre-trained on BookCorpus (Zhu et al., 2015) and English Wikipedia (the same as the regular BERT) but not trained on any in-domain datasets (such as any PubMed articles). Our results show that fine-tuning BERT-L8H128A2 yields similar results to BiLSTM-large. Fine-tuning BERT-L4H256A4 yields better results than the LSTM models, but it has twice the number of parameters than the BiLSTM-large model (comparison in Table 3).

However, if we pre-train them on PMC-10000, corresponding to models BERT-L8H128A2-Bio and BERT-L4H256A4-Bio, both the dev and test F1 scores largely improve (the improvement ranges from 0.03 to 0.06) compared to the equivalent models without in-domain pre-training.

The size of *BERT-L4H256A4-Bio* is much larger than other compact models in the table. This is mostly explained by the size of the embedding layer. We experiment reducing the embedding layer size using the similar approach as with the BiLSTM model: Keep the top 10,000 word pieces by frequency of the *base-base-uncased* tokenizer in PMC-10000 and resize the vocabulary to 10,000. The original pre-trained embeddings are used to initialize the embedding layers of the Reduced Vocab BERT-L4H256A4 (see Appendix C for details). The new models resulting of this procedure are identified by the *-RV* suffix in tables 2, 3 and 4.

Both BERT-L8H128A2-Bio-RV and BERT-L4H256A4-Bio-RV are pre-trained on PMC-10000 before fine-tuned on the causal precedence dataset. Previous work has shown that larger vocabulary sizes could slightly boost the performance of BERT-based models (Conneau et al., 2020). We observed different impacts of vocabulary reduction on BERT-L8H128A2 and BERT-L2H256A4. The test F1 of BERT-L4H256A4 drops from 0.548 to 0.537 whereas that of BERT-L2H128A2 even increases from 0.518 to 0.524. This shows that the impact of the vocabulary size to the BERT's performance is task- and model-dependent. Further, it is possible to gain some improvement by reducing the vocabulary size of BERT.

Model size and inference time

Table 3 shows the number of parameters of the models and their inference times on CPU and GPU. In general, all compact BERT models yield much better inference time than LSTM models on CPU. For example, both BiLSTM-large and BERT-L8H128A2 have approximately 5M parameters, with an inference time on CPU are 0.026s and 0.013s, respectively. This clearly shows the

[8] See the "W2V embedding" section of Section 5.1 for the explanation of *w2v* and *w2v-ID*.

Model	# Param.	# Embd. Param.	CPU Inf. T	GPU Inf. T
BiLSTM-small-VO2	0.66M	0.14M	0.007	0.002
BiLSTM-large-w2v(-ID)-VO2	5.40M	0.14M	0.026	0.006
BiLSTM-large-WP	5.63M	1M	0.031	0.009
BERT-L8H128A2(-Bio)	5.58M	3.91M	0.013	0.007
BERT-L8H128A2-Bio-RV	2.95M	1.28M	0.014	0.007
BERT-L4H256A4(-Bio)	11.17M	7.81M	0.011	0.005
BERT-L4H256A4-Bio-RV	5.92M	2.56M	0.011	0.004
BioBERT	108.31M	22.27M	0.119	0.012

Table 3: Model sizes and inference times. For all models, we show the total number of parameters, the number of parameters in the embedding layers (which can be reduced by reducing the model's vocabulary), the average CPU and GPU inference time (seconds per input sequence). The numbers are averaged across 5 runs of all examples.

transformer architecture of BERT is better suited for parallelization. Furthermore, BERT-L4H256A4 has about twice number of parameters as BERT-L8H256A4, but it has smaller inference time (0.011s vs 0.013s) because of fewer layers.

5.2 The Impact of Distillation

Previous work shows that knowledge distillation from a large model (teacher) to a compact model (student) does not always work and is highly dependent on the nature of task. For example, Wasserblat et al. (2020) found that distillation can be helpful for the tasks that require general lexical semantics. However, the distillation on our dataset is very challenging because: (1) there are only about 580 labeled training samples for the teacher, and (2) after fine-tuning, our teacher can only reach a 0.54 test F1 (BioBERT in Table 2).

We adopt a three-stage pipeline for distillation. (1) The teacher model (BioBERT) is fine-tuned on the labeled training data. (2) The teacher model runs inference on the labeled data (and optionally on the unlabeled data) to get the predictions scores for each example. (3) The student model is trained to reproduce the teacher's score on each training example with the loss function introduced in Section 4. Depending on how many unlabeled data to use, we evaluate 3 distillation settings: labeled, labeled + 2k unlabeled and labeled + 20k unlabeled. The results are shown in Table 4.

The impact of distillation on out-of-domain pre-trained models

Among the models we evaluate, BiLSTM(-small/large)-w2v and BERT-L4H256A4 were not pre-trained using PMC-10000, the in-domain corpus. For BiLSTM(-small/large)-w2v, distillation using only the labeled data is not helpful compared with direct fine-tuning. However, distillation becomes helpful when more unlabeled data are used. With BiLSTM-small-w2v, the testing F1 score increases from 0.452, when only labeled data is used for distillation, to 0.489, when using the labeled and 2k unlabeled examples for distillation. The testing F1 further improves to 0.496 when using labeled and 20k unlabeled examples for distillation. A similar trend is also found for BiLSTM-large-w2v. The trend for BERT-L4H256A4 is slightly different. When we use only labeled data, labeled data plus 2k unlabeled examples, and labeled data plus 20k unlabeled examples for distillation, testing F1 scores are 0.502, 0.499 and 0.516, respectively. Both BiLSTM(-small/large)-w2v and BERT-L4H256A4, using labeled data, plus 20k unlabeled examples for distillation attain better testing F1 scores compared to only using the labeled data for fine-tuning (table 2). It shows that in general, out-of-domain pre-trained models can largely benefit from distillation, especially when there are sufficient unlabeled data.

The impact of distillation on in-domain pre-trained models

We observed a similar pattern when distilling in-domain, pre-trained models. For most cases, the model's testing F1 score increased as more unlabeled data became available for distillation. The testing F1 scores of BiLSTM-large-WP increased from 0.400 to 0.430 and 0.487 when using either only labeled data, labeled data + 2k unlabeled examples, labeled + 20k unlabeled examples, respectively for distillation. Similar trends are also found for BERT-L4H256A4-Bio and BERT-L4H256A4-Bio-RV. The only exception we observed was BiLSTM-large-w2v-ID, whose testing F1 score was 0.441 when using the labeled data for distillation, then peaked at 0.477 when using labeled + 2k unlabeled data, just to decrease to

Model	Dev P.	Dev R.	Dev. F1	Test P.	Test R.	Test. F1
labeled						
BiLSTM-small-w2v-VO2	0.387 (0.014)	0.600 (0.090)	**0.467** (0.024)	0.426 (0.034)	0.491 (0.073)	0.452 (0.037)
BiLSTM-large-w2v-VO2	0.393 (0.026)	0.703 (0.045)	**0.503** (0.012)	0.414 (0.058)	0.508 (0.061)	0.450 (0.022)
BiLSTM-large-w2v-ID-VO1	0.524 (0.051)	0.587 (0.051)	0.550 (0.029)	0.488 (0.046)	0.407 (0.037)	0.441 (0.028)
BiLSTM-large-WP	0.462 (0.047)	0.598 (0.053)	**0.518** (0.036)	0.444 (0.051)	0.373 (0.051)	0.400 (0.024)
BERT-L4H256A4	0.364 (0.015)	0.592 (0.058)	**0.450** (0.024)	0.468 (0.044)	0.550 (0.042)	**0.502** (0.011)
BERT-L4H256A4-Bio	0.389 (0.022)	0.645 (0.048)	0.483 (0.008)	0.514 (0.044)	0.569 (0.040)	0.537 (0.007)
BERT-L4H256A4-Bio-RV	0.417 (0.031)	0.625 (0.057)	**0.498** (0.020)	0.554 (0.066)	0.555 (0.038)	**0.551** (0.035)
labeled + unlabeled 2k						
BiLSTM-small-w2v-VO2	0.467 (0.031)	0.574 (0.085)	**0.510** (0.029)	0.541 (0.049)	0.452 (0.057)	**0.489** (0.037)
BiLSTM-large-w2v-VO2	0.480 (0.031)	0.616 (0.082)	**0.535** (0.033)	0.561 (0.088)	0.438 (0.050)	**0.483** (0.018)
BiLSTM-large-w2v-ID-VO1	0.531 (0.027)	0.650 (0.066)	**0.583** (0.038)	0.535 (0.047)	0.434 (0.033)	**0.477** (0.023)
BiLSTM-large-WP	0.494 (0.027)	0.568 (0.079)	**0.525** (0.034)	0.498 (0.044)	0.384 (0.057)	**0.430** (0.037)
BERT-L4H256A4	0.394 (0.034)	0.591 (0.071)	**0.468** (0.013)	0.520 (0.066)	0.498 (0.076)	**0.499** (0.010)
BERT-L4H256A4-Bio	0.393 (0.024)	0.688 (0.045)	**0.499** (0.015)	0.509 (0.060)	0.584 (0.021)	0.541 (0.024)
BERT-L4H256A4-Bio-RV	0.408 (0.030)	0.678 (0.031)	**0.508** (0.018)	0.532 (0.057)	0.592 (0.030)	**0.558** (0.023)
labeled + unlabeled 20k						
BiLSTM-small-w2v-VO2	0.464 (0.027)	0.580 (0.080)	**0.512** (0.028)	0.539 (0.065)	0.468 (0.054)	**0.496** (0.029)
BiLSTM-large-w2v-VO2	0.458 (0.039)	0.629 (0.043)	**0.527** (0.015)	0.524 (0.082)	0.473 (0.054)	**0.490** (0.027)
BiLSTM-large-w2v-ID-VO1	0.512 (0.057)	0.575 (0.090)	0.537 (0.055)	0.521 (0.040)	0.399 (0.060)	**0.449** (0.039)
BiLSTM-large-WP	0.474 (0.026)	0.572 (0.040)	**0.517** (0.005)	0.563 (0.050)	0.432 (0.024)	**0.487** (0.021)
BERT-L4H256A4	0.382 (0.026)	0.617 (0.037)	**0.470** (0.008)	0.492 (0.075)	0.553 (0.062)	**0.516** (0.044)
BERT-L4H256A4-Bio	0.388 (0.028)	0.666 (0.046)	0.489 (0.014)	0.508 (0.057)	0.602 (0.039)	0.547 (0.014)
BERT-L4H256A4-Bio-RV	0.393 (0.017)	0.647 (0.043)	0.488 (0.007)	0.526 (0.062)	0.602 (0.033)	**0.558** (0.021)

Table 4: The compact model's performance using distillation with different amount of unlabeled data. The improved F1s (compared with the fine-tuned model's F1 in Table 2) are shown in **bold** text. All experiments are run for 5 seeds and 5-fold cross validation. The standard deviation across 5 random seeds is shown in the parenthesis.

Model	Dev P.	Dev R.	Dev. F1	Test P.	Test R.	Test. F1
Rule	0.534	0.272	0.360	0.523	0.170	0.257
SVM	0.361	0.407	0.383	0.395	0.364	0.379
Rule -> SVM	0.367	0.537	**0.436** ↑	0.383	0.445	**0.412** ↑
Rule -> BiLSTM-small-FT	0.325	0.713	**0.445** ↑	0.393	0.622	**0.476** ↑
Rule -> BiLSTM-large-FT	0.314	0.739	**0.441** ↑	0.381	0.659	**0.482** ↑
BiLSTM-small-FT -> SVM	0.308	0.767	**0.439** ↑	0.362	0.649	**0.461** ↑
BiLSTM-large-FT -> SVM	0.299	0.785	0.433 -	0.348	0.662	0.456 ↓
Rule -> BiLSTM-small-DS	0.414	0.618	0.493 ↓	0.482	0.507	0.492 ↓
Rule -> BiLSTM-large-DS	0.406	0.645	0.497 ↓	0.472	0.513	0.488 ↓
Rule -> BERT-L4-Bio-RV-DS	0.357	0.647	0.459 ↓	0.476	0.616	0.535 ↓
Rule -> BERT-L4-Bio-RV-FT	0.376	0.625	0.469 ↓	0.503	0.562	0.529 ↓
BiLSTM-small-DS -> SVM	0.377	0.692	0.487 ↓	0.411	0.593	0.485 ↓
BiLSTM-large-DS -> SVM	0.369	0.717	0.486 ↓	0.409	0.590	0.482 ↓
BERT-L4-Bio-RV-DS -> SVM	0.339	0.731	0.463 ↓	0.417	0.684	0.518 ↓
BERT-L4-Bio-RV-FT -> SVM	0.349	0.708	0.467 ↓	0.423	0.624	0.504 ↓
BiLSTM-s -> BERT-L4-DS	0.376	0.678	0.482 ↓	0.498	0.630	0.552 ↓
BiLSTM-l -> BERT-L4-DS	0.374	0.685	0.482 ↓	0.491	0.635	0.550 ↓
Rule + BiLSTM-l + BERT-L4	0.488	0.609	**0.540** ↑	0.585	0.478	0.521 ↓
SVM + BiLSTM-l + BERT-L4	0.458	0.612	0.523 ↓	0.551	0.495	0.517 ↓

Table 5: Results of the sieve models. "X -> Y" means model X's prediction is firstly used in the sieve then Y. FT means fine-tuning (entries in Table 2) and DS means distillation using **20k unlabeled data and labeled data** (entries in Table 4). The upside and downside arrows besides the scores indicate whether the sieve score outperforms the best individual model in the sieve. In the last four rows, BiLSTM-s is BiLSTM-small-DS, BiLSTM-l is BiLSTM-large-DS and BERT-L4 is BERT-L4H256A4-Bio-RV-DS. For all BiLSTM models we use w2v general embedding with VO2. The last two rows of the table are the ensemble model's performance.

0.449 when using labeled + 20k unlabeled data. This indicates in general the in-domain pre-trained models can still benefit from distillation when there are sufficient unlabeled data.

5.3 Comparison among Rule-based, Feature-based and Neural-based Models

Previous work has explored combining multiple models by using a sieve method (Mirza, 2014; Hahn-Powell et al., 2016). Generally speaking, a sieve method starts by using the model with the highest precision to predict the class of an input. If the prediction is positive, it is returned as the result, otherwise the input is forwarded to the model with the second best precision, and the process is repeated until a model makes a positive prediction or all the models are exhausted. In this work we explore the performance of a sieve method composed of multiple combinations of feature/rule/neural-based models. We rank the models by their decreasing precision in the development set. Table 5 contains the performance of different sieves.

Rules and SVM complement each other

As shown in Table 5, combining the rule-based and feature-based models into a sieve results in an improvement over either of them individually. However, this sieve is not on par with the performance of sieves that contain neural models.

Non-distilled neural models are complemented by non-neural models

Table 5 shows that in three out of four cases, sieves with a rule-based classifier or SVM classifier boost the performance of the LSTM models that are fine-tuned but not distilled. The benefits are more evident for the rule classifier than for the SVM classifier.

Well-trained neural models are not complemented by non-neural models

For models distilled with labeled and 20k unlabeled examples (BiLSTM-small/large-DS, BERT-L4-DS) and the model pre-trained both in the general domain and in the target domain (BERT-L4-FT), neither the rule-based classifier nor the SVM classifier result on increase the performance when sieved. This hints that well-trained neural models could learn to represent the same high-level hand-crafted features in the Rule and SVM classifier.

Different neural models are not likely to complement each other in a sieve

As the last two rows of Table 5 show, although BiLSTM and BERT are very different models, they do not tend to complement each other in a sieve.

6 Conclusion

In this work we trained several neural models for causal precedence detection in the biomedical literature. To help with the deployment of neural models on systems without a GPU, we restricted the sizes of our architectures to approximately $\frac{1}{20}$ of the size of a state-of-the-art language model such as BERT. Moreover, to overcome the challenge of scarcity of labeled training data, we used in-domain unlabeled data combined with pre-training and distillation and obtained robust neural models. Finally, we compared our neural models with previous rule-based and feature-based classifiers and found the in-domain pre-trained models can mostly replace them.

Acknowledgements

Thank Gus Hahn-Powell for his help on replicating the data and experiments from his previous work (Hahn-Powell et al., 2016). This work was supported by the Defense Advanced Research Projects Agency (DARPA) under the World Modelers and Automating Scientific Knowledge Extraction (ASKE, HR00111990011) programs. Mihai Surdeanu declares a financial interest in lum.ai. This interest has been properly disclosed to the University of Arizona Institutional Review Committee, and is managed in accordance with its conflict of interest policies.

References

Ashutosh Adhikari, Achyudh Ram, Raphael Tang, William L. Hamilton, and Jimmy Lin. 2020. Exploring the limits of simple learners in knowledge distillation for document classification with DocBERT. In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 72–77, Online. Association for Computational Linguistics.

Abbas Akkasi and Mari-Francine Moens. 2021. Causal relationship extraction from biomedical text using deep neural models: A comprehensive survey. *Journal of Biomedical Informatics*, 119:103820.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.".

Eduardo Blanco, Nuria Castell, and Dan Moldovan. 2008. Causal relation extraction. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08).*

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Tirthankar Dasgupta, Rupsa Saha, Lipika Dey, and Abir Naskar. 2018. Automatic extraction of causal relations from text using linguistically informed deep neural networks. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 306–316, Melbourne, Australia. Association for Computational Linguistics.

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, Minneapolis, Minnesota. Association for Computational Linguistics.

Roxana Girju. 2003. Automatic detection of causal relations for question answering. In *Proceedings of the ACL 2003 workshop on Multilingual summarization and question answering*, pages 76–83.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural networks*, 18(5-6):602–610.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Gus Hahn-Powell, Dane Bell, Marco A. Valenzuela-Escárcega, and Mihai Surdeanu. 2016. This before that: Causal precedence in the biomedical domain. In *Proceedings of the 15th Workshop on Biomedical Natural Language Processing*, pages 146–155, Berlin, Germany. Association for Computational Linguistics.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. TinyBERT: Distilling BERT for natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4163–4174, Online. Association for Computational Linguistics.

Christopher SG Khoo, Syin Chan, and Yun Niu. 2000. Extracting causal knowledge from a medical database using graphical patterns. In *Proceedings of the 38th annual meeting of the association for computational linguistics*, pages 336–343.

Jin-Dong Kim, Ngan Nguyen, Yue Wang, Jun'ichi Tsujii, Toshihisa Takagi, and Akinori Yonezawa. 2012. The genia event and protein coreference tasks of the bionlp shared task 2011. In *BMC bioinformatics*, volume 13, pages 1–12. BioMed Central.

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 BioNLP 2009 workshop companion volume for shared task*, pages 1–9.

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

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.

Linqing Liu, Huan Wang, Jimmy Lin, Richard Socher, and Caiming Xiong. 2019a. Mkd: a multi-task knowledge distillation approach for pretrained language models. *arXiv preprint arXiv:1911.03588*.

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

Luke Melas-Kyriazi, George Han, and Celine Liang. 2019. Generation-distillation for efficient natural language understanding in low-data settings. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 124–131, Hong Kong, China. Association for Computational Linguistics.

Claudiu Mihăilă, Tomoko Ohta, Sampo Pyysalo, and Sophia Ananiadou. 2013. Biocause: Annotating and analysing causality in the biomedical domain. *BMC bioinformatics*, 14(1):1–18.

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

Paramita Mirza. 2014. Extracting temporal and causal relations between events. In *Proceedings of the ACL 2014 Student Research Workshop*, pages 10–17, Baltimore, Maryland, USA. Association for Computational Linguistics.

Amr Mousa and Björn Schuller. 2017. Contextual bidirectional long short-term memory recurrent neural network language models: A generative approach to sentiment analysis. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1023–1032.

Subhabrata Mukherjee and Ahmed Hassan Awadallah. 2019. Distilling transformers into simple neural networks with unlabeled transfer data. *arXiv preprint arXiv:1910.01769*, 1.

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.

Haojie Pan, Chengyu Wang, Minghui Qiu, Yichang Zhang, Yaliang Li, and Jun Huang. 2021. Meta-KD: A meta knowledge distillation framework for language model compression across domains. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3026–3036, Online. Association for Computational Linguistics.

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*, 21:1–67.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Antonio Sorgente, Giuseppe Vettigli, and Francesco Mele. 2013. Automatic extraction of cause-effect relations in natural language text. *DART@ AI* IA*, 2013:37–48.

Raphael Tang, Yao Lu, and Jimmy Lin. 2019a. Natural language generation for effective knowledge distillation. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 202–208, Hong Kong, China. Association for Computational Linguistics.

Raphael Tang, Yao Lu, Linqing Liu, Lili Mou, Olga Vechtomova, and Jimmy Lin. 2019b. Distilling task-specific knowledge from bert into simple neural networks. *arXiv preprint arXiv:1903.12136*.

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Well-read students learn better: On the importance of pre-training compact models. *arXiv preprint arXiv:1908.08962*.

Marco A Valenzuela-Escárcega, Özgün Babur, Gus Hahn-Powell, Dane Bell, Thomas Hicks, Enrique Noriega-Atala, Xia Wang, Mihai Surdeanu, Emek Demir, and Clayton T Morrison. 2018. Large-scale automated machine reading discovers new cancer-driving mechanisms. *Database*, 2018. Bay098.

Konstantinos Z Vardakas, Grigorios Tsopanakis, Alexandra Poulopoulou, and Matthew E Falagas. 2015. An analysis of factors contributing to pubmed's growth. *Journal of Informetrics*, 9(3):592–617.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in Neural Information Processing Systems*, 33:5776–5788.

Moshe Wasserblat, Oren Pereg, and Peter Izsak. 2020. Exploring the boundaries of low-resource BERT distillation. In *Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing*, pages 35–40, Online. Association for Computational Linguistics.

Bowen Wu, Huan Zhang, MengYuan Li, Zongsheng Wang, Qihang Feng, Junhong Huang, and Baoxun Wang. 2020. Towards non-task-specific distillation of bert via sentence representation approximation. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 70–79.

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 IEEE international conference on computer vision*, pages 19–27.

A Types of Events of PMC-10000

phosphorylation, phosphorylates, ubiquitination, ubiquitinates, hydroxylation, hydroxylates, sumoylation, sumoylates, glycosylation, glycosylates, acetylation, acetylates, farnesylation, farnesylates, ribosylation, ribosylates, methylation, methylates, binding, binds, activation, activates.

B More Description of the Rule-based Classifier

The event pair causal precedence relation is predicted using a few hand-written deterministic rules. There are three types of rules: The first type are **intra-sentence** rules, where the two events are in the same sentence. Patterns of this type operate over the syntactic dependency graph of the sentence. The second type are rules for **inter-sentence** relations, where the two events occur on different sentences and a dependency graph is not available. These kind of rules use the presence of patterns, such as “leads to”, “result in” to predict causal precedence. The third kind of rules, also for inter-sentence event pairs, use verbal-tense information. Phrases such as “has been phosphorylated” are used to detect the existence of causal precedence.

C Reducing the Vocabulary of BERT-L4H256A4 and Resizing the Embeddings

The original vocabulary of the *bert-base-uncased* model has a size of 30,522. As discussed in Section 5.1, we count the frequency of the word pieces in PMC-10000 and only maintain the top 10000 most frequent word pieces.

The next step would be to resize the embedding layer of BERT-L4H256A4. Note that the original embeddings of BERT-L4H256A4 are pre-trained in the language modeling task on BookCorpus and English Wikipedia. We don’t want to lose such information during the resizing of the embedding layer by initializing the 10000 token embeddings randomly. Instead, the new embedding weights are initialized with the values of the corresponding weights of the original embedding layer (i.e., all the embedding weights in the new embedding layer reuses the pre-trained weights of the old embedding layer).