Using Pre-trained Transformer Deep Learning Models to Identify Named Entities and Syntactic Relations for Clinical Protocol Analysis

Miao Chen, [1] **Fang Du,** [2] **Ganhui Lan,** [2] **Victor Lobanov,** [2]
[1]Covance, 8211 SciCor Drive, Indianapolis, IN, USA [2]Covance, 206 Carnegie Center, Princeton, NJ, USA
{miao.chen, fang.du, ganhui.lan, victor.lobanov}@covance.com

Abstract

Transformer deep learning models, such as BERT, have demonstrated their effectiveness over previous baselines on a broad range of general-domain natural language processing (NLP) tasks such as classification, named entity recognition, and question answering (Devlin et al. 2018). They also exhibit enhanced performance in domain-specific NLP tasks, including BioNLP tasks (Lee et al. 2019; Alsentzer et al. 2019). In this study, we focus on clinical trial protocols: exploring and extracting key terms (a named entity recognition task) as well as their relations (a relation extraction task) from the protocols using transformer pre-trained deep learning models. We compare several model configurations and report their results. Our NLP model achieves good performance considering the complex and unique nature of the language in real-world protocols, and has been integrated into the organization's protocol analytics practice. This approach and the extracted information will greatly facilitate trial feasibility analysis for developing new drugs.

Introduction

Clinical trial protocols (often called "study protocols") contain key information specifying the trial design and implementation, but are usually in unstructured or semi-structured format, which presents a huge challenge for running computational analysis on them. Due to protocols' critical role, drug development businesses, such as contract research organizations, have been devoting significant amount of resources in analyzing study protocols to precisely understand the operational requirements, comprehensively evaluate the systemic challenges, unbiasedly assess the probability of success, accurately forecast the cost implications for optimal business planning. Currently, this protocol analysis work is still performed in a labor-intensive fashion, involving numerous resource checking and cross referencing works. To develop safer, cheaper and more effective drugs faster for better public health, this presses an urgent need for more efficient and effective ways to process text-based protocols.

Here, we present our efforts to facilitate the protocol analysis workflow by automating the process of extracting key information from the protocols using natural language processing (NLP) techniques. More specifically, we focus on the eligibility criteria section in the protocols, which contains patient selection criteria information; we extract key clinically relevant entities (i.e. named entities) and entity relations (i.e. syntactic relations) from this section. Based on the extracted information, the unstructured protocols can be transformed into a structured network with interconnected key entities (e.g. condition, drug, observation etc.) that can be fed into various data-based analytic tasks, for example to query against various real-world evidence databases for patient population estimation, which is critical for clinical trial design in drug development.

Covance Inc. is the world's largest provider for clinical trial design, monitoring, managing and central lab testing services, and has accumulated large volume of study protocols. The presented work is our first step of a bigger mission towards solving the protocol analysis challenge. To this end, we employed the transfer learning strategy and experiment with deep learning family of algorithms by using the recently developed Bidirectional Encoder Representations from Transformers (BERT) based models and fine-tuning them on our in-house clinical trial protocol corpus to identify the named entities and their relations.

Study protocols are rigorous scientific documents with highly domain-specific terms and complex relations. These characteristics bring both benefits and challenges to NLP work: we concern less about preprocessing due to its rigorous use of language, but need to attend more to its unique yet complex clinical terms and relations. A study protocol's eligibility criteria section is usually composed of two parts: inclusion criteria and exclusion criteria, which respectively describe the unambiguous characteristics of patients to be included in and excluded from the clinical trial. The general public can access some simplified protocol texts via websites such as ClinicalTrials.gov, which already contain many clinical terminologies. However, the real protocols are much longer with even more domain-specific terms, thus more difficult for the NLP task. We employ pre-trained BERT transformers to tackle this challenging NLP task and our study provides quantified evidence of how BERT performs in the clinical trial domain.

```
{
  "inclusion criteria" : {
     "I1" : { "gender" : "male" },
     "I2" : { "age" : "45-80" },
     "I3" : { "ICD10" : "E29.1",
                "LOINC": {"2986-8" : "100-300"} },
     "I4" : { "ICD10" : "I21" }
    }
}
```


Figure 1: Structured information extracted from protocol eligibility criteria.

In our practice, the extracted information are stored in a structured format. Figure 1 shows an example: the inclusion criteria is represented as several key-value clauses such that we can query a patient database to find the patients satisfying these criteria. Through extraction we are essentially connecting dots to build a larger graph for knowledge engineering purpose, i.e. we connect protocol text to patient database records, connect protocol to condition terms in a medical ontology, and so on. Once the dots are properly connected, we are empowered to perform many protocol analysis tasks such as building a search engine for precise search, composing graph networks for graph analysis for capturing missing links, evaluating drug effectiveness by comparing with similar drugs, clustering and recommending similar protocols for study feasibility analysis.

Related Work

Named entities recognition (NER) and relation extraction (RE) are two classical natural language processing (NLP) tasks, which we carry out to extract entities and syntactic relations respectively in our study. Previously, for NER, researchers have mainly investigated probabilistic sequence labeling models such like conditional random fields (CRF), maximum entropy Markov models, and hidden Markov model(Lafferty, McCallum, and Pereira 2001; McCallum, Freitag, and Pereira 2000; Bikel et al. 1998). For RE, text classification methods, such as support vector machine, logistic regression, and perceptron, along with feature engineering, have been used to assign relations between entities(Bach and Badaskar 2007; Jurafsky 2000).

In recent years, with the advances in deep neural network methods, significant performance improvement has been achieved for the NER and RE tasks. For NER tasks, embeddings are widely used in neural network models to represent words or characters as high-dimensional vectors. Recurrent neural networks (RNN), including LSTM, GRU, and their variants, are applied because their architectures represent better the sentence context as well as the dynamic sentence length in natural languages (Huang, Xu, and Yu 2015; Yang, Salakhutdinov, and Cohen 2016). The Bidirectional LSTM (Bi-LSTM) plus CRF network architecture has also been widely used to achieve better NER performance (Ma and Hovy 2016; Lample et al. 2016).

Despite the improvement from previous models, RNN and LSTM models tend to "forget" earlier context in long sequences, which limits the model performance. Transformers are subsequently proposed to counter this issue. Transformer models use the attention mechanism that attends to each word in a sequence by replacing the sequence-based RNN style network structure with dot products and multiplications between the key/value/query matrices projected from the embedding vectors (Vaswani et al. 2017). Transformers have the advantage of attending to every token in a sequence, whether long or short, and therefore they can capture associations between tokens that are even distantly separated from each other. BERT models (Bidirectional Encoder Representations from Transformers), a recent popular NLP deep learning model, is a model employing multiple layers of attentions and significantly improved NLP task performance over previous models (Devlin et al. 2018).

Additionally, transfer learning aims to transfer pre-trained model from one task to another, usually by training a general language model on general-domain data set and transferring it to a downstream task by fine-tuning on the task-specific data set. A number of pre-trained language models have been created to facilitate downstream tasks such as NER and RE, examples including ELMO, ULMFit, OpenAI GPT, and BERT, which have outperformed previous baselines and some even achieved the state-of-the-art performance(Peters et al. 2018; Howard and Ruder 2018; Radford et al. 2019).

Based on the original BERT architecture, a number of BERT variants have emerged with alterations for different purposes. For example, RoBERTa removes next sentence prediction from the original loss function along with some other hyperparameter changes; Transformer-XL captures context both within and between segments for tackling long-term dependency across sentences; and T5 advocates for encoding-decoding architecture, denoising objectives and other changes based on extensive experiments(Liu et al. 2019; Dai et al. 2019; Raffel et al. 2019).

NER and RE have also been longstanding tasks in the biomedical NLP domain. Researchers have investigated applying similar yet more customized approaches to biomedical texts, such as using CRF models and BiLSTM+CRF neural networks (Leaman and Gonzalez 2008; Lyu et al. 2017; Wei et al. 2016). With the introduction of the BERT model, BERT based models have been adopted to the biomedical domain by retraining it with biomedical corpus, among the examples are BioBERT, SciBERT, and clinical BERT(Lee et al. 2019; Beltagy, Cohan, and Lo 2019; Alsentzer et al. 2019).

In the clinical informatics field, it is important to convert unstructured criteria text to structured format because this enables people to automatically parse a criteria and query for proper patients against certain real-world evidence database. Therefore, NER and RE algorithms are an appropriate and natural fit to this practice: NER extracts concepts such as conditions and observations that is related to a patient; RE provides operational information such as the range for a particular lab test result for patient selection. Criteria2Query is a pioneering work in the space of translating study criteria to SQL queries(Yuan et al. 2019). It relies mainly on CRF

sequence labeling for the NER task and SVM classification for relation extraction. To the best of our knowledge, there has been no research and practice to use pre-trained transformer deep learning methods to extract structured information from unstructured clinical trial protocols. Motivated by the excellent performance of BERT based models on NER and RE tasks in general domains, we experiment and develop models and evaluate the performance in the clinical trial domain.

Methodology

Data Set

To facilitate our NLP approach, we selected 470 study protocols from Covance's in-house protocol database. And our protocol corpus comprises eligibility criteria sections from these selected study protocols. An eligibility criteria section typically contain 5 - 20 sentences that define the criteria to select and recruit patients for the clinical study. Our data contain a total of 30,183 criteria sentences.

Data Annotation. We have the eligibility criteria annotated using the IOB format (Ramshaw and Marcus 1999). The corpus is annotated by well-trained biomedical domain experts as the gold standard for training and testing. They manually annotate the key clinical entities and their pairwise relations if there exist any. We focus on 15 types of entities and 7 types of relations that help clinically define a patient cohort:

Entities: Condition, Observation, Procedure, Device, Drug, Investigational product, Event, Refractory condition, Demographics, Measurement, Temporal constraints, Qualifier/modifier, Anatomic location, Negation cue, Permission cue

Syntactic relations: Has value, Has temporal constraint, Modified by, Located in, Is negated, Is permitted, Specified by

Data Split. For the NER task, we randomly split the 30,183 sentences into training (60%, 18,109 sentences) and test (40%, 12,074 sentences) sets. For the RE task, before splitting the data for training and testing, we first check whether a sentence contains multiple relations and if so, we duplicate the sentence for each pair of related entities and make their relation type as the label for classification. This results in 52,470 relation sample sentences, based on which we perform a random split with stratification on relation classes to derive training (60%, 31,482 relation samples) and test sets (40%, 20,988 relation samples). Tables 1 and 2 show data statistics for the NER and RE tasks.

Table 1: Train and test data counts for the NER task.

Entity	Train	Test
Condition	12,682	8,537
Observation	7,309	5,218
Procedure	3,406	2,234
Device	221	140
Drug	7,793	5,858
Investigational product	329	224
Event	2,430	1,625
Refractory condition	381	278
Demographics	498	381
Measurement	4,540	3,344
Temporal constraints	6,968	4,589
Qualifier/modifier	7,853	5,196
Anatomic location	427	223
Negation cue	921	615
Permission cue	1,236	869

Table 2: Train and test data counts for the RE task.

Relation	Train	Test
is negated	703	468
is permitted	1,009	673
modified by	5,715	3,810
has value	3,326	2,218
has temporal constraint	6,169	4,112
is located	215	143
specified by	3,729	2,486
no relation	10,616	7,078
total count	31,482	20,988

NER Task

As previously mentioned, we use NER algorithms to extract clinically relevant entities in eligibility criteria section and particularly choose BERT, a pre-trained transformer type of deep learning model, because of its reported superior performance in many NLP tasks. Due to the attention transformer in BERT, it is able to provide dynamic context embedding for tokens, which helps addressing the polysemy issue. BERT is a language model pre-trained on a large general domain corpus and can be applied towards downstream tasks by adding simply structured task layers and fine tuning on task-specific data set. We hereby follow the fine tuning practice based on pre-trained models to derive our NER model (Devlin et al. 2018; Lee et al. 2019). We explore several options with regard to choice of pre-trained models and task layers.

NER task layers. The original BERT paper indicates that when use for NER tasks, the pre-trained BERT model can be simply followed by a softmax layer where each token is classified to their most likely entity class without adding any CRF layer(Devlin et al. 2018). However, our experiments suggest that this approach sometimes fails to recognize contiguous phrases as whole entities. To address this issue, we further experiment the architecture with BiLSTM+CRF layers as the NER task layer for its potentially better ability in capturing bi-diretional context as well as tagging likelihood at the sentence level (as opposed to token level).

Cased or uncased. The BERT model provided by Google includes versions with and without lowercasing preprocessing on the tokens. We experiment with both the cased (not applying lowercasing) and uncased (applying lowercasing) options. Consequently, the two options use different set of subword vocabularies, with cased model of 28,996 subwords and uncased model of 30,522 subwords.

Pre-trained models. We use BERT-base, a smaller ver-

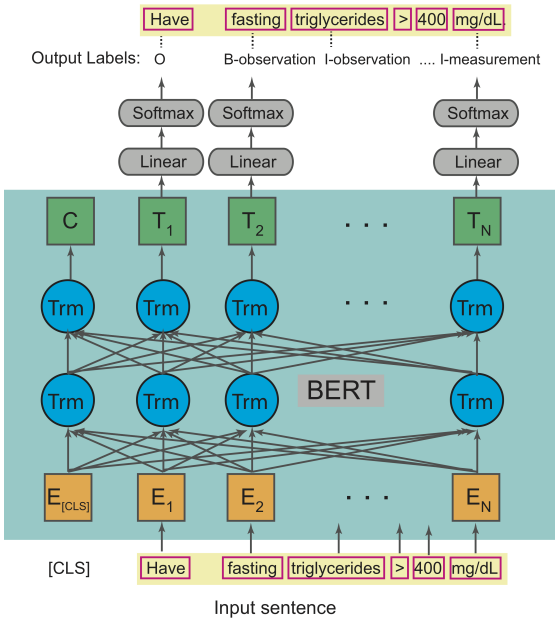

Figure 2: Neural architecture of the BERT NER task (with Softmax as the task layer).

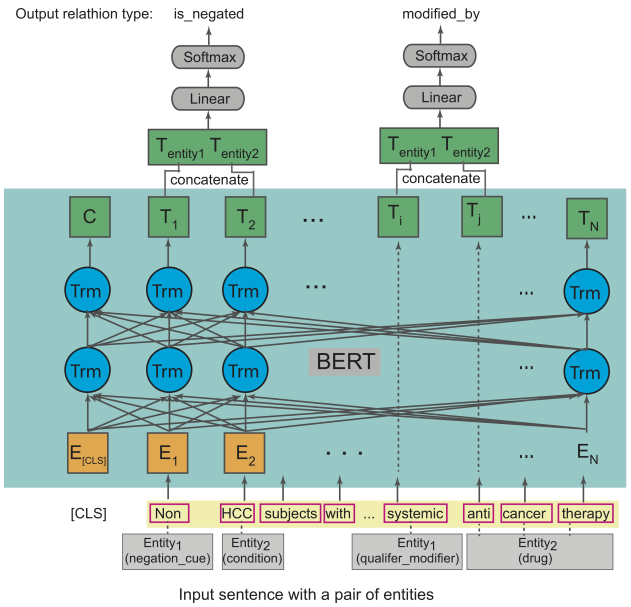

Figure 3: Neural architecture of the BERT RE task (with Softmax as the task layer).

sion of BERT that comprises 110 millions of parameters, in our first set of experiments. BERT also has a larger version, BERT-large, with 340 millions parameters. We opt to use BERT-base for exploration purposes. In our second set of experiments, we test the BioBERT model that is retrained using large-scale biomedical texts on the basis of the original BERT model. BioBERT has only a cased version and shares the same vocabulary as BERT-base cased (with size of 28,996).

Hyperparameters. For both BERT-base and BioBERT models, we set num_of_epochs=20, learning_rate=$2 * 10^{-5}$, training_batch_size=32, max_sequence_length=32. For cases when using BiLSTM+CRF as task layers, we set bilstm_layer_size=128.

The above model options result in 6 NER models:

- $BERT_{base,uncased}$, $Softmax$: BERT base uncased pre-trained model, softmax as NER task layer
- $BERT_{base,cased}$, $Softmax$: BERT base cased pre-trained model, softmax as NER task layer
- $BioBERT$, $Softmax$: BioBERT pre-trained model (cased), softmax as NER task layer
- $BERT_{base,uncased}$, $BiLSTM + CRF$: BERT pre-trained uncased model, BiLSTM+CRF as NER task layer
- $BERT_{base,cased}$, $BiLSTM + CRF$: BERT base pre-trained cased model, BiLSTM+CRF as NER task layer
- $BioBERT$, $BiLSTM + CRF$: BioBERT pre-trained model (cased), BiLSTM+CRF as NER task layer

The layout of the BERT NER neural architecture is shown in Figure 2.

RE Task

The RE task is also treated as a downstream task to the pre-trained models. The original BERT paper did not include RE task as one of their downstream tasks, whereas the BioBERT study investigated it due to its importance in the biomedical NLP domain(Lee et al. 2019). BioBERT handles relation extraction as a classification task on the sentence or sequence level. In particular, it assumes that each sentence contains at most one relation and classifies whether a whole sentence, instead of a particular pair of entities, contains a relation of interest, e.g. Gene-disease relation. This approach is not directly applicable to our data for 2 reasons: 1) our data contain multiple types of relations, and 2) in our data set, one sentence often contains multiple relations (52,470 relations/30,183 sentences = 1.7 relations/sentence on average).

We employ the following strategy for the RE task: In training, we first scan through each sentence for entities using human annotations, and record the token positions of each entities; if a sentence contains n $(n > 1)$ pairs of entities with human annotated relation, we duplicate this sentence n times so that each instance target represents one pair of entities and their relation; In prediction, we use NER pipeline results to locate entities, enumerate all legitimate entity pairs, and duplicate sentences accordingly. Since we record the token positions of each entity pair, we can get BERT output vectors for them based on their position information, concatenate the two vectors and then feed it to a softmax layer to classify their relation. The result can be one of the 7 relations listed in Table 2 or 'no relation'.

More specifically, the input fed to the BERT RE model is sentence text along with positions of entity pairs. We do not make use of entity type information for the following reasons: 1) this end-to-end (i.e. tokens-to-relation) practice

makes the RE model more useful as a standalone tool that does not require entity type; 2) when in prediction mode, the errors in entity prediction could propagate to the RE task, which we mitigate by including only the entity position information. Figure 3 shows the neural architecture of our RE task.

For training purposes, we randomly generate negative samples for the 'no relation' class as two entities can have no relations with each other. We use two ways to obtain negative samples: one way is to randomly choose two unrelated entities in a sentence, the other is to break an existing related entity pair and establish a non-related pair between one of the entities in the original pair and another unrelated entity in the sentence.

Similar to the NER task, we experiment with 3 pre-trained models with softmax as the task layer for all of them:

- $BERT_{base,uncased}$: BERT base pre-trained model, uncased
- $BERT_{base,cased}$: BERT base pre-trained model, cased
- $BioBERT$: BioBERT pre-trained model (cased)

Following hyperparameter configuration is used: num_of_epochs=20, learning_rate=$2 * 10^{-5}$, training_batch_size=32, max_sequence_length=32.

Results and Analysis

We implement the NER and RE tasks using Tensorflow based on the BERT neural architecture and run experiments on an AWS p2.xlarge GPU instance.

NER Results

We follow the practice in the SemEval-2013's Drug-Drug Interactions task and evaluate NER performance by 3 matching standards: strict, exact, and partial (Segura-Bedmar, Martínez, and Herrero-Zazo 2013). The strict matching evaluates both boundary and entity type of entity phrases; the exact matching evaluates the exact boundary regardless of entity type; and the partial matching measures the partial boundary of entities regardless of entity type (thus the most lenient). We calculate precision(P)/recall(R)/f1-score(F) for the three evaluation types, and additionally, we also report macro average P/R/F results. The results are shown in Table 3.

In our experiments, fine-tuning the pre-trained BioBERT model achieves slightly better performance than its BERT counterparts. For example, $BioBERT, Softmax$ has f1-score of 70.61, better than $BERT_{base,uncased}, Softmax$'s 69.80 and $BERT_{base,cased}, Softmax$'s 69.68. Similarly, $BioBERT, BiLSTM + CRF$ holds a higher f1-score than $BERT_{base,uncased}, BiLSTM + CRF$ and $BERT_{base,cased}, BiLSTM + CRF$ for all the four evaluation types.

When comparing the cased and uncased strategies, we notice that the uncased pre-trained models outperform the cased ones with the same neural architecture: e.g. $BERT_{base,uncased}, BiLSTM + CRF$ achieves f1-score of 70.28 for the strict evaluation type, higher than the f1-score of 69.89 from $BERT_{base,cased}, BiLSTM + CRF$.

Table 3: NER task results: Precision(P), Recall(R), F1 Score(F).

NER Model	Type	P	R	F
$BERT_{base,uncased}$, $Softmax$	strict	67.76	71.98	69.80
	exact	71.02	75.44	73.16
	partial	75.28	79.96	77.55
	macro	62.65	66.83	64.63
$BERT_{base,cased}$, $Softmax$	strict	67.82	71.66	69.68
	exact	71.19	75.22	73.15
	partial	75.41	79.68	77.49
	macro	63.04	66.37	64.63
$BioBERT$, $Softmax$	strict	68.73	**72.60**	70.61
	exact	71.87	**75.91**	73.83
	partial	75.99	**80.26**	78.06
	macro	62.97	67.27	65.03
$BERT_{base,uncased}$, $BiLSTM + CRF$	strict	68.59	72.06	70.28
	exact	71.85	75.49	73.62
	partial	76.10	79.95	77.98
	macro	63.43	66.45	64.88
$BERT_{base,cased}$, $BiLSTM + CRF$	strict	68.09	71.80	69.89
	exact	71.34	75.22	73.23
	partial	75.55	79.67	77.56
	macro	62.68	66.41	64.45
$BioBERT$, $BiLSTM + CRF$	strict	**69.12**	72.47	**70.76**
	exact	**72.35**	75.85	**74.06**
	partial	**76.55**	80.25	**78.36**
	macro	**63.79**	**67.44**	**65.54**

This finding suggests that applying lowercase to preprocessing actually enhances performance slightly, which is counter-intuitive for NER tasks as the entities are often case-sensitive. Meanwhile, we also find that the two BioBERT models, which are cased, perform better than their peer models of the same neural architecture. But since BioBERT only offers the cased option, we cannot discern the relative contribution from being cased in the BioBERT pre-trained model.

From Table 3, it is not surprising that for a given model, the partial evaluation usually holds the highest score, followed by exact, strict, and macro. Another observation is that when we loosen evaluation type from strict to exact, i.e. focusing on entity boundary without penalizing entity type errors, the performance is improved but still remains in the 73.15-74.06 range, suggesting that the experimented BERT based models fail identify entity boundary very precisely, which can be of interest for future investigation.

In our experiments with simple Softmax as the task layer, we observe more boundary detection errors. This in fact is the motivation for us to add the BiLSTM+CRF layers as the NER task layer. However, the results show that given the same pre-trained model configuration, it is debatable that BiLSTM+CRF could consistently improve performance. For example, $BioBERT, BiLSTM + CRF$ slightly outperforms $BioBERT, Softmax$ in strict matching precision and f1-score, but $BioBERT, Softmax$ beats $BioBERT, BiLSTM + CRF$ in strict matching recall.

We also find that the recall score is consistently higher than the precision score for all models at all evaluation stan-

dards, indicating that the models tend to have more false positive predictions than false negative predictions. The macro scores show lower performance than strict/exact/partial because it simply averages the performance of different entity types and some small-sample entity types have lower performance due to lack of training data.

Overall, $BioBERT, BiLSTM + CRF$ produces the best precision and f1-scores for all the four evaluation types whereas $BioBERT, Softmax$ holds the highest recalls. These results suggest that fine-tuning BioBERT lends itself better to the NER tasks in the clinical trial domain, which seems intuitive. But for task layer, the choice between Softmax and BiLSTM+CRF does not significantly affect the performance.

RE Results

RE evaluation results are shown in Table 4, in which we report micro/macro/weighted precision(P), recall(R), and f1-score(F).

Table 4: RE task results: Precision(P), Recall(R), F1 Score(F).

RE Model	Type	P	R	F
$BERT_{base,uncased}$	micro	**78.10**	**79.49**	**78.79**
	macro	**76.43**	**76.22**	**76.24**
	weighted	**78.03**	**79.49**	**78.72**
$BERT_{base,cased}$	micro	73.61	75.33	74.46
	macro	69.56	68.63	68.80
	weighted	73.41	75.33	74.27
$BioBERT$	micro	74.37	74.83	74.60
	macro	70.30	68.34	69.08
	weighted	74.17	74.83	74.44

From the above performance chart, we find that $BERT_{base,uncased}$ has the highest f1-scores, whereas $BERT_{base,cased}$ has the lowest. Comparing $BERT_{base,cased}$ and $BioBERT$ indicates that BioBERT can help with performance slightly, at least for this cased scenario. On the other hand, $BERT_{base,uncased}$ noticeably improves over its cased peer, $BERT_{base,cased}$, by a 4.33 percentage margin. Therefore, just like the NER task, the RE task is also case insensitive, probably because uncased situations reduce vocabulary variations in processing. We also observe that recall and precision are close to each other with precision slightly higher for the macro evaluation, but on the contrary, precision is slightly higher than recall for micro and weighted. These observations suggest that the model has higher precision score than recall score in classes with less samples, such as 'is_located' and 'is_negated' (in Table 1). And when doing macro evaluation, the contribution from the smaller classes becomes more visible.

Overall, the $BERT_{base,uncased}$ model prevails - it outperforms the other two models on each evaluation type and measures. For example, it has f1-score of 78.79 for micro, compared to $BERT_{base,cased}$'s 74.46 and $BioBERT$'s 74.60. These results indicate again that the lowercasing preprocessing helps the NLP tasks even in the clinical trial domain where many terms are represented in capital letters. Secondly, $BioBERT$ beating $BERT_{base,cased}$ with a small margin may suggest that although pre-training in the biomedical domain could bring in some benefit, it is still not specific enough for clinical trials. Since there is no uncased BioBERT pre-trained model available, it is unclear whether training on biomedical corpus with lowercasing preprocessing could synergistically improve the performance. Considering the big improvement from $BERT_{base,cased}$ to $BERT_{base,uncased}$, we believe the uncased scenario of current BioBERT model is worth future investigation.

Error Analysis

We present and inspect NER prediction results from one of the models ($BERT_{base,uncased}, Softmax$) in a Brat server, an open source tool that can help visualize annotation results using color bars (Stenetorp et al. 2012). We overlay human and prediction annotations together in Brat to facilitate the comparison.

The NER errors can be broadly categorized into boundary errors and entity type errors, as reflected by the four evaluation types. For boundary errors, one pattern is that BERT tends to mis-annotate some words inside a multi-word phrase. For example, as shown in Figure 4, "at least a 3 month" is one temporal constraint entity, but the NER model only captures "at"+"3 month" while misses the words in the middle ("least a"). This reflects a potential problem with BERT NER models: although it can assign entity classes relatively well, lack of structure enforcement on its output layer may possibly cause the inconsistent label within a full phrase.

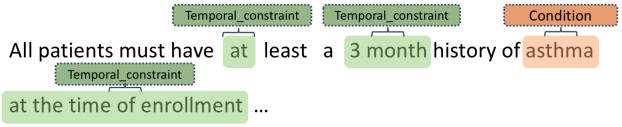

Figure 4: An example of the NER engine mis-annotating tokens within a phrase.

In some cases, the NER model captures longer entities than the human annotator. For example, the model annotates "[*cardiac mechanical assist device*]|Device"; whereas the gold standard annotates the same phrase as "[*cardiac*]|AnatomicLocation" + "[*mechanical assist device*]|Device". In some other cases, the situation reverses and the NER model chunks one entity in the gold standard into multiple ones. For example, "[*non-steroidal anti-inflammatory drugs*]|Drug" is chunked into a Qualifier/Modifier and a drug: "[*non-steroidal*]|Qualifier/Modifier [*anti-inflammatory drugs*]|Drug". The boundary merging and chunking issues, as illustrated by these two examples, occur frequently with the Qualifier/Modifier class as it is arguable that a complex term can be annotated by one whole entity or as a Qualifier/Modifier plus an entity.

For the entity type error, we observe a few cases, such as "*urinalysis*—Procedure type" is predicted as an Observation entity, and "*gastrointestinal motility*—Condition type"

is predicted as Drug. The type errors occur less frequently than boundary errors according to our manual inspection.

For the RE task, we manually screen the predictions from the $BERT_{base,uncased}, Softmax$ model against the gold standards. We first observe that the NER boundary errors can propagate to the RE task. Note that we only use named entity positions but not types in the RE task, and therefore only NER boundary errors can affect the RE performance. For example, "Transient neurologic deficits", annotated as one Condition entity in the gold standard, is split into "*Transient*—Qualifier/Modifier" and "*neurologic deficits*—Condition", thus causing the RE task to predict a 'modified by' relation between the two entities which actually does not exist in the gold standard. Another major category of RE classification error is that a number of actual relations misclassified as 'no relation', while misclassification between other classes is much less frequent.

Conclusion and Future Work

In this study, we focus on extracting clinically relevant terms and relations from protocol eligibility criteria by applying pre-trained transformer deep learning NLP models for NER and RE tasks. We experiment with several configurations of the pre-trained BERT models and report our results and findings.

Our results demonstrated the effectiveness of NLP models in processing clinical trial protocols. Despite of the fact that the processed texts are unique with specific clinical and medical terms and logical relations, BERT and BioBERT models returned acceptable performances. We also find that in general, BioBERT, which is pre-trained on biomedical corpus, outperforms BERT, which is pre-trained on general domain corpus. This agrees with the general understanding of the importance of domain-specific training for achieving higher model performance in domain-specific tasks.

A surprising finding is that even though the clinical trial domain largely contains capitalized terminologies, lowercasing preprocessing improves the performances of both NER and RE tasks. Our hypothesis is that maintaining less token variation (i.e. lowercasing has less variation) is more important than maintaining casing for these tasks.

It is also worth noting that there are rooms to improve the quality of our gold standard. Due to the complex nature of the protocols that cover many different sub-domains in biomedical and clinical sciences such as therapeutic areas, even human experts can easily make mistakes or be inconsistent. In fact, we found many cases that the model predictions are in fact correct, although different from the gold standard. To address this annotation quality issue, we employed an iterative annotating pipeline that asks human experts to verify the pre-annotated documents by the NLP models. We anticipat that this practice can help partly address this issue.

We believe that the model performance can be further improved. To do that, we can further explore in several directions. The first approach is to train a biomedical BERT model using a domain-specific vocabulary from scratch. BERT model handles tokens by splitting them into subwords using a predefined subword vocabulary. For example, 'myocarditis' and 'pericarditis', two heart conditions sharing the same suffix 'carditis', are however represented as 'my'+'##oca'+'##rdi'+'##tis' and 'per'+'##ica'+'##rdi'+'##tis' respectively. This way of tokenization does not represent the suffix in a biomedically meaningful way due to the lack of biomedical subwords in the vocabulary. We assume subwords generated from the biomedical domain reflecting word root patterns can further enhance the word representation for BERT models and thus improve downstream task performance. We can train a BERT model from scratch using a biomedical corpus and a biomedical subword vocabulary.

The second strategy is to deploy multi-task co-training: since NER and RE tasks are dependent on each other, namely, knowing one task's output can facilitate the other task's, and therefore joint learning on them is expected to improve performances for both.

Our third strategy for future improvement is to reduce unnecessary relations currently predicted from the RE model. Our current greedy prediction pipeline enumerates all possible entity pairs that results in an unnecessarily large testing base set. One way to address this issue is to consider dependency parsing information, which can be used to indicate whether two terms has dependency relations to prune unnecessary entity pairs.

The extracted information from the NER and RE tasks has the great potential of assisting drug development business especially for study feasibility analysis. The derived information is the basis for a local knowledge graph for the protocols and a global graph when merging with external structured information such as drug ontologies. In conclusion, this is our first step towards a greater mission to apply deep learning to business cases in drug development, and the subsequent analysis based on the derived graph can even further enhance our contribution and insights to this research area.

References

Alsentzer, E.; Murphy, J. R.; Boag, W.; Weng, W.-H.; Jin, D.; Naumann, T.; and McDermott, M. 2019. Publicly available clinical bert embeddings. *arXiv preprint arXiv:1904.03323*.

Bach, N., and Badaskar, S. 2007. A review of relation extraction. *Literature review for Language and Statistics II* 2.

Beltagy, I.; Cohan, A.; and Lo, K. 2019. Scibert: Pretrained contextualized embeddings for scientific text. *arXiv preprint arXiv:1903.10676*.

Bikel, D. M.; Miller, S.; Schwartz, R.; and Weischedel, R. 1998. Nymble: a high-performance learning name-finder. *arXiv preprint cmp-lg/9803003*.

Dai, Z.; Yang, Z.; Yang, Y.; Cohen, W. W.; Carbonell, J.; Le, Q. V.; and Salakhutdinov, R. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Howard, J., and Ruder, S. 2018. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*.

Huang, Z.; Xu, W.; and Yu, K. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Jurafsky, D. 2000. *Speech & language processing*. Pearson Education India.

Lafferty, J.; McCallum, A.; and Pereira, F. C. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *ICML proceedings*.

Lample, G.; Ballesteros, M.; Subramanian, S.; Kawakami, K.; and Dyer, C. 2016. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

Leaman, R., and Gonzalez, G. 2008. Banner: an executable survey of advances in biomedical named entity recognition. In *Biocomputing 2008*. World Scientific. 652–663.

Lee, J.; Yoon, W.; Kim, S.; Kim, D.; Kim, S.; So, C. H.; and Kang, J. 2019. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Lyu, C.; Chen, B.; Ren, Y.; and Ji, D. 2017. Long short-term memory rnn for biomedical named entity recognition. *BMC bioinformatics* 18(1):462.

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

McCallum, A.; Freitag, D.; and Pereira, F. C. 2000. Maximum entropy markov models for information extraction and segmentation. In *ICML*, volume 17, 591–598.

Peters, M. E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; and Zettlemoyer, L. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; and Sutskever, I. 2019. Language models are unsupervised multitask learners. *OpenAI Blog* 1(8).

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv e-prints*.

Ramshaw, L. A., and Marcus, M. P. 1999. Text chunking using transformation-based learning. In *Natural language processing using very large corpora*. Springer. 157–176.

Segura-Bedmar, I.; Martínez, P.; and Herrero-Zazo, M. 2013. Semeval-2013 task 9: Extraction of drug-drug interactions from biomedical texts (ddiextraction 2013). Association for Computational Linguistics.

Stenetorp, P.; Pyysalo, S.; Topić, G.; Ohta, T.; Ananiadou, S.; and Tsujii, J. 2012. brat: a web-based tool for NLP-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, 102–107. Avignon, France: Association for Computational Linguistics.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

Wei, Q.; Chen, T.; Xu, R.; He, Y.; and Gui, L. 2016. Disease named entity recognition by combining conditional random fields and bidirectional recurrent neural networks. *Database* 2016.

Yang, Z.; Salakhutdinov, R.; and Cohen, W. 2016. Multi-task cross-lingual sequence tagging from scratch. *arXiv preprint arXiv:1603.06270*.

Yuan, C.; Ryan, P. B.; Ta, C.; Guo, Y.; Li, Z.; Hardin, J.; Makadia, R.; Jin, P.; Shang, N.; Kang, T.; et al. 2019. Criteria2query: a natural language interface to clinical databases for cohort definition. *Journal of the American Medical Informatics Association* 26(4):294–305.