TpT-ADE: Transformer Based Two-Phase ADE Extraction

Suryamukhi Kuchibhotla and **Manish Singh**
Indian Institute of Technology Hyderabad
Telangana, India
cs17m19p100001@iith.ac.in and msingh@cse.iith.ac.in

Abstract

Extracting adverse reactions to medications or treatments is a crucial activity in the biomedical domain. The task involves identifying mentions of drugs and their adverse effects/events in raw text, which is challenging due to the unstructured nature of clinical narratives. In this paper, we propose TpT-ADE, a novel joint two-phase transformer model combined with natural language processing (NLP) techniques, to identify adverse events (AEs) caused by drugs. In the first phase of TpT-ADE, entities are extracted and are grounded with their standard terms using the Unified Medical Language System (UMLS) knowledge base. In the second phase, entity and relation classification is performed to determine the presence of a relationship between the drug and AE pairs. TpT-ADE also identifies the intensity of AE entities by constructing a parts-of-speech (POS) embedding model. Unlike previous approaches that use complex classifiers, TpT-ADE employs a shallow neural network and yet outperforms the state-of-the-art methods on the standard ADE corpus.

1 Introduction

Adverse Drug Event (ADE) is a negative or harmful patient outcome that seems to be associated with a medication or drug. Analyzing the adverse events (AEs) helps a practitioner to identify susceptible patients who may be at risk due to a particular drug. ADE extraction has several uses: pharmaceutical companies can identify the sections of the population that were adversely impacted by the drug. For governments and regulatory authorities, ADE information is the key to monitoring the performance of drugs already in the market and identifying any adverse effects that have not appeared during clinical trials.

Generally, ADEs are reported in an unstructured manner and are to be extracted from various sources like clinical narratives, medical journals, formal systems that report ADEs, etc. In some cases, the patients may report adverse events in social media posts, like "I got *rashes* on my back today after taking two tablets of *amoxicillin* yesterday". In this post, *"rashes"* is the adverse effect (AE) that could be caused by the drug *"amoxicillin"*. Identifying the drugs and adverse events and finding relations between them from such unstructured text is quite challenging due to the complex nature of the text containing multiple drugs and adverse events. We illustrate with an example below to understand the complexity of such texts. The text in red color is the AE and the text in blue is the drug name.

Example — "Atypical ventricular tachycardia$_{\text{AE}}$ torsade pointes$_{\text{AE}}$ induced by amiodarone$_{\text{Drug}}$: arrhythmia$_{\text{AE}}$ previously induced by quinidine$_{\text{Drug}}$ and disopyramide$_{\text{Drug}}$."

Following are the *drug, AE* relations that could be extracted from the above example:

disopyramide$_{\text{Drug}}$,quinidine$_{\text{Drug}}$ → arrhythmia$_{\text{AE}}$

amiodarone$_{\text{Drug}}$ → Atypical ventricular tachycardia$_{\text{AE}}$, torsade de pointes$_{\text{AE}}$

ADE extraction is a two-step process. The first step is identifying the mentions of the drugs and the AEs from raw text. This is similar to the task of named entity recognition. In the second step, each $< drug, AE >$ pair is examined for ADE relation, which can be cast as a classification problem.

Some methods (Dandala et al., 2017; Unanue et al., 2017) train separate models for the two steps of ADE extraction. In contrast to these works, (El-Allaly et al., 2022; Ma et al., 2022; Wadden et al., 2019; Bekoulis et al., 2018b; Zhou et al., 2017) proposed joint methods for ADE extraction that perform better in both recognizing the entities and ADE extraction tasks. A major drawback of the former approach is that if the first step of identification of drugs and AEs entities is incorrect, then the ADE

Proceedings of the 28th Conference on Computational Natural Language Learning, pages 209–218
November 15-16, 2024 ©2024 Association for Computational Linguistics

extraction will also be incorrect, thereby resulting in poor performance due to the error propagation. Also, the joint models have been proven to be effective in performing many related tasks such as part-of-speech tagging and parsing (Zhang and Clark, 2008), keyword extraction using joint modeling of local and global context (Liang et al., 2021), entity extraction and classification (Eberts and Ulges, 2019), entity and coreference extraction (Hajishirzi et al., 2013; Durrett and Klein, 2014), and many more.

In this paper, we introduce TpT-ADE, a joint two-phase model for ADE extraction from clinical texts by fine-tuning BERT (Devlin et al., 2018). In the first phase, TpT-ADE identifies and standardizes mentions of entities such as drugs and adverse effects against the Unified Medical Language System (UMLS)[1]. This ensures uniformity in naming across different mentions. The second phase uses this processed text to jointly extract entities and classify relations.

Our model employs a robust span-based extraction method, which can extract entities consisting of multiple successive tokens. That is, TpT-ADE is able to extract overlapping entities. Our approach can also detect the intensity of an adverse event, distinguishing between terms like "fever", "severe fever", and "mild fever". Unlike previous works that rely on complex relational classifiers, TpT-ADE uses a shallow neural network and yet achieves higher F1-score on the standard ADE corpus (Gurulingappa et al., 2012).

2 Related Work

In this section, we discuss the related works in ADE extraction. We first discuss the pipeline based approaches that extract ADEs by training separate models for entity extraction and relation extraction tasks. Then, we discuss the joint methods that follow an end-to-end approach to extract ADEs. Under the joint models, we discuss the related works that are BiLSTMs based, Graph Convolutional Networks based and Span-based models.

The pipeline based approaches (Dai et al., 2020; Wei et al., 2020; Dandala et al., 2017) are designed to complete one subtask and then go ahead with the next subtask. In the case of ADE extraction, the output from the entity extraction model is passed as the input for the relation extraction task. Both the models are trained separately with different loss functions. (Wei et al., 2020; Xu et al., 2017; Dandala et al., 2017) use BiLSTM based models for ADE extraction. (Wei et al., 2020) employs the same BiLSTM based classifiers for both entity and relation extraction. Other works train two different classifiers for the two tasks. (Alfattni et al., 2021; Dai et al., 2020) employ a hybrid approach by combining feature based machine learning classifiers and neural networks.

Identifying negative entities is a crucial step for extracting ADEs. Negative entities are those that are not drugs or adverse effects. Towards this, (Wei et al., 2020) proposed an Attention based Bi-LSTM model that reduced the number of negative instances, helping to overcome the imbalance class problem. Their method could also handle the discontinous entitiess. More recently, (He et al., 2022) proposed an LSTM based adaptive knowledge distillation model. The authors used BERT to adaptively distill the knowledge to the LSTM model. Other recent works proposed in this regard are (Wang et al., 2022; He et al., 2023; Liu et al., 2023).

Joint entity and relation extraction methods (Bekoulis et al., 2018a,b; Ma et al., 2022; Eberts and Ulges, 2019; Wadden et al., 2019) have been recently proposed to capture the dependency between the two tasks in ADE extraction. (Bekoulis et al., 2018a,b) utilize character and Word2Vec embeddings to represent their input. Then, they use BiLSTM model combined with conditional random field (CRF) model to jointly extract entities and their relations.

(Wang and Lu, 2020; Wang et al., 2021; Yan et al., 2021; Ma et al., 2022) cast the ADE extraction problem as a table-filling problem. These methods construct a table that jointly represents the entities and relations and each element in the table depicts the presence of a relation between entities. Then, the relation triples are extracted from the filled table. (Yan et al., 2021) constructed a partition filter network to learn the feature representations that can classify entities and the relations. Then, the relation triples extracted by following a table-filling approach. Similarly, (Ma et al., 2022) proposed a table-filling method that learns contextualized representations to compute entity mentions and capture long-range dependencies. For relation extraction, a tensor dot product is used to predict the relation labels. However, these table-filling methods are computationally expensive due to building and deconding these tables for relation

[1] https://www.nlm.nih.gov/research/umls/index.html

triples (Chen et al., 2024).

Span-based methods (Luan et al., 2019; Wadden et al., 2019; Eberts and Ulges, 2019; Wan et al., 2023) have shown remarkable performance in obtaining contextualized representations. In contrast to works that follow the BIO (beginning, inside, outside)/BILOU (beginning, inside, last, outside, unit)/BIES(Begin, Inside, End, Single) (Zheng et al., 2017; Zhou et al., 2017), span-based approach can identify the overlapping entities. In our work, we follow a span-based approach and combine BERT (Devlin et al., 2018) with POS embedding model. In contrast to the previous works, we follow a two phase joint modelling approach that standardizes the entity mentions in the input text with their representative terms. In addition, unlike the above works that use complex classfiers, we use a shallow neural network for ADE extraction.

3 Methodology

In this section, we detail the two phases of TpT-ADE model and training it. In the first phase Phase I, we extract the entities and represent them with their standard medical terms. In Section 4, we show the effectiveness of this step. The second phase, Phase II utilizes this processed text for ADE extraction.

3.1 Phase I: Entity Extraction

In this phase, we perform entity mention extraction or recognition and find the most representative term for each entity mention in the raw clinical text corpus. The architecture for entity recognition is shown in Figure 1. Towards this, we propose a span based BERT model. BERT learns word representations from input text by considering both the left and right contexts. We first tokenize the input sentences into a sequence of tokens T using a subword tokenization algorithm called Byte-Pair Encoding (BPE) (Sennrich et al., 2015). BPE tokenizes the input sentences in such a way that the most common words are represented in the vocabulary as a single token. The infrequent words are divided into commonly occurring subwords. For example, the infrequent word *townhall* can be divided into frequently occurring *town* and *hall*. Thus, BPE can be used by BERT to map out of vocabulary words and limit the vocabulary size. BPE tokens extracted from each input sentence are passed to the BERT model to obtain an embedding sequence as follows:

$$(c, e_1, e_2, ...e_n) = BERT(T) \tag{1}$$

The first token c in BERT is the classifier token (cls), shown in Figure 1, that captures the overall input sentence context. We then construct spans considering all the token subsequences. For instance, the token sequence *carbamazepine toxicity symptoms* can result into token subsequences or spans like *carbamazepine*, *carbamazepine toxicity*, etc. The span based approach ensures we search all the possible combinations, is more robust, and is expected to extract the entity that may be composed of multiple successive tokens.

We treat the entity mention extraction problem as a classification problem where each span is classified into one of three categories, namely, *Drug*, *AE* or *None* by the *Entity Classifier* in Figure 1. *None* means the span is neither *Drug* or *AE* and these are filtered out. Initially, a pre-trained BERT model is utilized and adjusted to the clinical domain to explore the information in the clinical text documents. The model is then fine-tuned for classifying the spans into the aforementioned three categories. We fine-tune the pre-trained BERT model by adding a task-specific layer on top of it and training the whole model end-to-end with a suitable loss function. This is detailed in Section 3.3.

Let $s_i = (e_1, e_2, ..., e_k)$ be a span consisting of k token subsequences. The BERT embeddings of the token subsequences are combined using max-pooling and the span embedding of span s_i is represented as follows:

$$s^{s_i} = \textit{max-pooling}(e_1, e_2, ..., e_k) \oplus c \tag{2}$$

where $\oplus$ denotes concatenation. We note that any span longer than ten tokens are filtered out to limit the cost of entity classification.

The raw clinical text is collected from varied sources and hence the same drug or AE entities could be mentioned with different names. For instance, consider the following two texts from our dataset:

Example 1 — *After gastric-outlet obstruction was recognized in several infants who received prostaglandin E1, we studied the association between the drug and this complication*

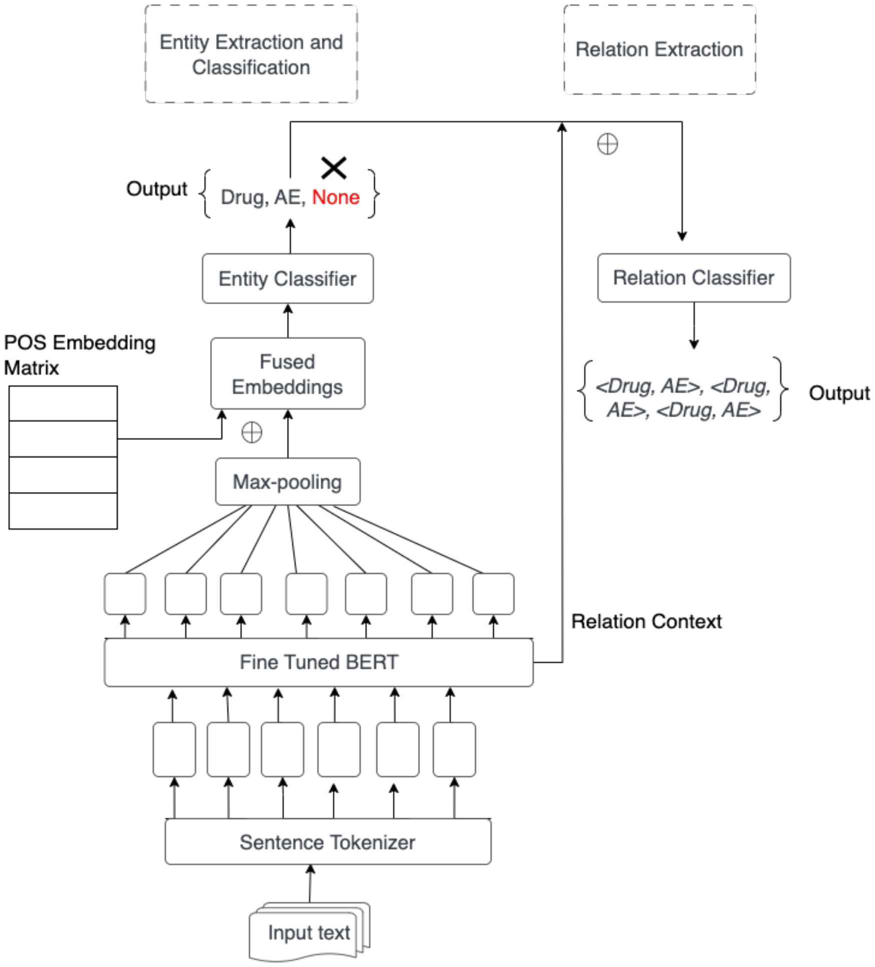

Figure 1: Entity and Relation Extraction

Example 2 — *The clinical symptoms of gastric mucosa foveolar hyperplasia due to long-term PGE1 therapy simulate hypertrophic pyloric stenosis*

In the above two examples, the drug references *"prosta- glandin E1"* and *"PGE1"* refer to the same drug, whose standard UMLS name is given by Metamap[5] as *"alprostadil"*. As the next step in this phase, we replace the entities with their most representative terms. We will observe in Section 4.3 that standardising the entity mentions with their representative terms improves the overall performance of TpT-ADE.

In this phase of TpT-ADE, we also identify the intensity of the AEs caused by drugs as discussed in Section 1. Specific modifiers which precede the identified entity may need to be added to the entity itself. For example, entity *fever* is distinguished from the entity *severe fever* as both are different AEs. The same holds true for many modifiers like "Severe", "Reversible", "Paradoxical", "Unusual", "Chronic", etc. Towards this we identify the adjectives of the entities using spaCy[6]. SpaCy is NLP library used for generating POS tags of tokens in a given input sentence. We used ScispaCy (Neumann et al., 2019) trained with *en_core_sci_sm* that processes clinical or biomedical text. The POS embedding matrix is trained to obtain the representation of POS tags. Adjectives specifying the entities are then concatenated with the BERT embeddings. In Section 4.3, we demonstrate that the performance of the model improves when POS tag embeddings are included. Finally, the POS tag embeddings and the BERT embeddings are concatenated to obtain the following entity representation.

$$x^{s_i} = s^{s_i} \oplus p^{s_i} \tag{3}$$

where p^{s_i} is POS tag embeddings that specify the intensity of span s_i.

Next, the softmax classifier given below is used to obtain a posterior for each entity category, i.e.,

[5] https://lhncbc.nlm.nih.gov/ii/tools/MetaMap.html

[6] https://spacy.io/

drug, *AE* and *none*. The output of this phase of our model is the processed clinical text.

$$Y^{s_i} = softmax(W^{s_i} \cdot x^{s_i} + b^{s_i}) \tag{4}$$

3.2 Phase II: ADE Extraction

In this phase, we jointly extract entities and perform relation classification using the processed text from Phase I as shown in Figure 1. The text is tokenized using BPE tokenizer and the fused span embeddings are constructed using BERT model for entity classification. Max-pooling fusion function is used as it performed the best. The spans having a length of more than ten tokens are filtered as too longer spans are highly unlikely to represent entities. The entities classified into *none* class are filtered out. Let $\mathcal{E}$ be the set of entity spans classified as either *Drug* or *AE*.

The next step of this phase is relationship classification. The relationship classifier takes each pair of fused BERT span embedding from entity spans in $\mathcal{E} \times \mathcal{E}$ and checks the presence of a relation between them. Let f^{s_i} be the fused BERT embedding of the entity span $s_i = (e_1, e_2, ..., e_k)$, which is calculated as follows:

$$f^{s_i} = max\text{-}pooling(e_1, e_2, ..., e_k) \tag{5}$$

To understand the presence of a relation, it is important to understand the context. One way to obtain the context is the classifier token c from the embedding span representation, as discussed above. However, the context c would not be precise and could represent multiple relations for longer sentences. Thus, we derive the relationship context between entity spans localized to their direct surrounding entities. Let s_i and s_j be two entity spans considered to check the presence of a relation. The relation context $c^{rel}(s_i, s_j)$ is derived from the fused BERT embedding of the span ranging from the end of s_i entity to the beginning of the s_j entity. For obtaining $c^{rel}(s_i, s_j)$, we found the max-pooling function performing the best. In case the entities are next to each other or overlapping, we set $c^{rel}(s_i, s_j) = 0$.

Another consideration for relationship classification between two entities could be asymmetrical. That is, s_i could indicate *drug* and s_j could be *AE*, or vice versa. Therefore, we need to consider both (s_i, s_j) and (s_j, s_i) for relationship classification. Hence, we have the following two representations as input to the relation classifier.

$$\begin{aligned} Rel(x^{s_i, \rightarrow s_j}) &= f^{s_i} \oplus c^{rel}(s_i, s_j) \oplus f^{s_j} \\ Rel(x^{s_j, \rightarrow s_i}) &= f^{s_j} \oplus c^{rel}(s_j, s_i) \oplus f^{s_i} \end{aligned} \tag{6}$$

These two inputs are passed to a shallow single layer relationship classifier with a threshold α. A high response in the sigmoid layer indicates the presence of relationship between s_i and s_j. We consider that the relationship exists based on threshold value α; any relation with score $\geq \alpha$ is considered as related and assumed no relationship otherwise.

3.3 Training TpT-ADE Model

In this section, we detail the process to learn the parameters W^{s_i}, b^{s_i}, W^r, and b^r, thereby fine-tuning our BERT model in this process. Our model consists of two phases, and these parameters are learned in a supervised manner. That is, the entities and relations are labeled in our dataset. For both phases, training is done in batches. We draw positive and negative samples for each batch for the classifiers in both phases. We detail the positive and negative sample selection and loss functions for both phases below.

For entity classification, all the labeled entities in the ground truth dataset are taken as positive samples. Let this set be $\mathcal{E}^g$. We take a fixed number of negative samples $\mathcal{E}^{ne}$ in each batch. We illustrate the selection of positive and negative samples for entity classification with an example. In the given sentence: “Nine **azotemic**$_{\text{AE}}$ patients who developed a **blood coagulation disorders**$_{\text{AE}}$ associated with the use of either **cephalosporins**$_{\text{Drug}}$ or **moxalactam**$_{\text{Drug}}$ antibiotics are reported.” the ones marked as *Drug* or *AE* constitute the positive samples. Negative samples such as **associated**$_{\text{Drug}}$ and **reported**$_{\text{AE}}$ are randomly selected.

For training the relationship classifier, we use all ground truth relationships as positive samples. Instead of randomly selecting negative samples, we devise a method to select only the strong negative samples $\mathcal{E}^{nr}$ drawn from the entity pairs $\mathcal{E}^g \times \mathcal{E}^g$ that were not labeled as any relation. For example, the positive samples in the above example are *(cephalosporins, blood coagulation disorders)* and *(moxalactam, blood coagulation disorders)*, then the unlabelled relations like *(cephalosporins, azotemic)*, *(moxalactam, azotemic)* are taken as negative samples. Such strong negative samples instead of random pairs of entities help to improve the performance of the model.

In the first phase, the model learns parameters W^{s_i}, b^{s_i} used for entity recognition. Using the training set with annotated entities, the loss function for the first phase, $\mathcal{L}^1$, is defined as the entity classifier's cross-entropy loss over entity classes *Drug, AE, none*. The joint loss function for entity classification and relation classification in the second phase is defined by combining the losses from both the classifiers as follows:

$$\mathcal{L}^2 = \mathcal{L}^e + \mathcal{L}^r \tag{7}$$

where $\mathcal{L}^e$ is the entity classifier cross-entropy loss over all the three entity classes and $\mathcal{L}^r$ is the binary entropy loss averaged over batches' samples.

4 Evaluation

In this section, we present the evaluation of TpT-ADE and compare it with the state-of-the-art (SOTA) methods. We first start by describing our dataset. Next, we present the evaluation results of our model against the SOTA methods. Lastly, we perform ablation studies with various variants of our model and show the effectiveness of various components of our model.

4.1 Experimental Setup

We use the ADE corpus dataset (Gurulingappa et al., 2012) to train and evaluate our model. It contains 5,063 drugs, 5,776 adverse effects and 6,821 relations between them, extracted from 4,272 unique samples.

Table 1: Dataset Statistics

Statistics	Train	Val	Test
Drugs	3646	922	495
AEs	4151	1062	563
Relations	4877	1285	659
Documents	3076	769	427

To evaluate our model, we divided the dataset into training, validation and test sets, as shown in Table 1. Our model TpT-ADE is trained on the training set. We conduct 10-fold cross-validation on the validation set, and the evaluation is performed on the test set.

We used $\text{BERT}_{\text{BASE}}$[7] transformer with 768 dimensional embeddings and 110M parameters, pre-trained with 3 billion plus English words. In our experiments, we use Adam Optimizer with learning rate of 0.00005, weight decay of 0.01, *lr* warmup of 0.1, batch size of 2. The number of negative samples in both entity and relation classification, $\mathcal{E}^{ne}$ and $\mathcal{E}^{nr}$ are set to 80 per document. We run the model for 30 epochs with the relation classifier threshold set to 0.04. We obtained the best results with these parameter values. The BERT model weights are updated during the training process.

We evaluate our model for both entity extraction and relationship classification. If the predicted span of an entity and its type, that is, either Drug or AE are found exactly matching with the ground truth data, then the entity is considered to be correctly predicted. For relationships, both entities of the relationship must be correctly predicted as given in the ground truth. As in previous works (Bekoulis et al., 2018b; Eberts and Ulges, 2019), we use precision, recall and F1 scores averaged over folds as performance metrics to evaluate our model.

4.2 Baseline Methods

To evaluate the effectiveness of our TpT-ADE model in both entity and relationship classification, we compare its performance with the state-of-the-art methods listed below.

1. **Joint CNN Model (Li et al., 2016)**: This method uses transition-based feed-forward CNN to perform greedy transition-based decoding and jointly performs ADE extraction.

2. **Joint BiLSTM-RNN Model (Li et al., 2017)**: This method uses a BiLSTM-RNN model to learn the representations of entities and their contexts from the input text. Then, another BiLSTM-RNN model is built to learn the relations between the entities based on the shortest dependency path between them.

3. **Joint Multi-head Selection Model (Bekoulis et al., 2018b)**: This method uses character and Word2Vec embeddings to represent the input text. Then BiLSTM-CRF model is trained to extract entities and ADEs.

4. **SpERT (Eberts and Ulges, 2019)**: This method uses span based BERT models for extracting entities and adverse relations.

5. **TablERT-CNN (Ma et al., 2022)**: This BERT based method extracts ADEs by casting ADE extraction as a table-labelling problem. Two-dimensional CNN is used to encode the local

[7] https://huggingface.co/bert-base-cased

dependencies between the cells and predict the their labels.

6. **SMAN (Wan et al., 2023)**: This span based approach constructs a multi-model attention network to capture the interactions between the spans and model information such as tokens and labels. The context and span position information is extracted simultaneously.

4.2.1 Results

Table 2 shows the performance on both entity and relation extraction tasks on the test set. The table shows some missing values as these numbers weren't reported by the corresponding SOTA methods. For entity extraction task (NER), our model achieved an F1-score of 91.17%, which is 1.47% higher than the TablERT-CNN and 0.22% over SMAN. In addition, our model shows a significant improvement of 4.58% over the popular state-of-the-art model SpERT and 1.57% over the SMAN method in the case of ADE extraction (RE). Unlike all these baseline methods, our model finds the most representative clinical term of each entity mention. This step makes the training process of our model's first phase more robust, thereby improving the performance of the ADE relation extraction in the second phase. Thus, on the unseen test data, even if the input text entity is called by any other alias phrase name, it can still be detected and mapped to its most representative name. We performed error analysis on our model and observed that it gives the least number of false-negatives in both NER and RE tasks. One reason for this could be that TpT-ADE can identify complex and ambiguous entities.

Qualitative analysis shows that TpT-ADE is able to correctly identify **Ventricular tachycardia**$_{\text{AE}}$ and **Arrhythmia**$_{\text{AE}}$ referring to the same AE. Also, the abbreviation V-tach or VT is correctly recognized as ventricular tachycardia. In addition, the interactions between the AE **torsade pointes**$_{\text{AE}}$ and drugs **amiodarone**$_{\text{Drug}}$, **quinidine**$_{\text{Drug}}$ and **disopyramide**$_{\text{Drug}}$ were extracted by our model, unlike the previous models that were able to extract only the interaction between **torsade pointes**$_{\text{AE}}$ and **amiodarone**.

Compared to (Eberts and Ulges, 2019) (SpERT) and (Wan et al., 2023), which also uses a span based model, our model shows improved performance on both the NER and RE tasks. Span based approach to extract entities thus is more effective than to use BILOU/BIS labels as in (Bekoulis et al., 2018b; Li et al., 2017) (Joint Multi-head Selection, Joint BiLSTM-RNN Model). We also note that the input text also contains the intensity of the AEs that can be identified by our model in contrast to the baseline methods. Specifically, the ADE dataset contains 148 of such instances. In addition, unlike most of the baseline methods, our span based model detects the entity phrases that might contain overlapping entities. Specifically, the ADE dataset contains 120 of such overlapping instances.

4.3 Ablation Studies

In this section, we perform experiments on variants of our model and hyperparameters settings to demonstrate their impact on our model.

Effectiveness of Entity Standardization — In this study, we analyze the effectiveness of finding the representative term for each entity mention in the raw input text. We illustrate with an example from our dataset. The entities *common skin rashes*, *rashes*, *skin eruptions*, *cutaneous eruptions*, all refer to the same adverse effect. The representative term for all of them is *Exanthema*. Our model was trained using the training set that contains *rashes*, *skin eruptions*. The test set contains *cutaneous eruptions* that was correctly mapped to its representative term *Exanthema*. From Table 3, it can be observed that the F1-score of *W/O Entity Standardization* (removing entity standardization from TpT-ADE) drops a little by 0.87% in the case of NER task and significantly decreases (3.02%) in the case of RE task when compared to our TpT-ADE model.

Effectiveness of Entity Intensity Identification — We also investigate the effectiveness of enriching the BERT embeddings of the entities with the POS tag embeddings that provide the intensity information. For this purpose, we compare our TpT-ADE model with *W/O Entity Intensity Identification* model (without the POS embedding matrix) as shown in Table 3. It can be observed that there is a decrease in the performance of both the tasks in *W/O Entity Intensity Identification* (1% for NER and more than 2% for RE) compared to our Tpt-ADE model. Therefore, this shows the importance of linguistic information obtained by training the POS embedding matrix.

Effectiveness of Relation Context — Here, we examine the effect of using relation context in the ADE extraction phase detailed in Section 3.2 in-

	NER			RE		
Method	Precision	Recall	F1-score	Precision	Recall	F1-score
Joint CNN	79.50	79.60	79.50	64.00	62.90	63.40
Joint BiLSTM-RNN	82.70	86.70	84.60	67.50	75.80	71.40
Joint Multi-head Selection	84.72	88.16	86.40	72.10	77.24	74.58
SpERT	**89.26**	89.26	89.25	78.09	80.43	79.24
TablERT-CNN	-	-	89.7	-	-	80.5
SMAN	-	-	90.95	-	-	82.25
TpT-ADE	89.24	**93.2**	**91.17**	**81.91**	**85.83**	**83.82**

Table 2: Comparison of TpT-ADE with the baseline methods results(%)

	NER			RE		
Method	Precision	Recall	F1-score	Precision	Recall	F1-score
TpT-ADE	89.24	93.2	91.17	81.91	85.83	83.82
w/o Entity Standardization	88.71	91.95	90.30	78.44	83.35	80.82
w/o Entity Intensity Identification	88.74	91.63	90.16	79.86	83.61	81.69
Classifier Token Context	-	-	-	73.5	80.22	76.71
Weak Random Sampling	-	-	-	76.39	81.7	78.96

Table 3: Ablation studies results (%). w/o indicates the specific module is removed from TpT-ADE.

stead of using the classifier context, which uses a special token to capture the meaning of the entire sentence. The relation context particularly extracts the context from the part of the sentence that depicts the presence of a relationship between the entities the most. From Table 3, we can see that the performance of the TpT-ADE model, which uses the relation context in ADE extraction phase (RE task) achieves an F1-score of 83.82%, while the *Classifier Token Context* (CTC) model performs poorly with F1-score of 76.7%. Moreover, the precision drops by 8.41% as compared to TpT-ADE. Thus, this shows that training the model with relation context is better in ADE extraction.

Effectiveness of Negative Sampling — We also examine the effectiveness of choosing strong negative samples in the ADE extraction phase against using random negative samples. Negative samples are randomly drawn, and the entity pairs do not match with any ground truth relation pairs. Unlike choosing strong negative samples from the entity candidate set $\mathcal{E}$, these weak samples are randomly drawn. From Table 3, it can be observed that the performance of the *Weak Random Sampling* model drops by almost 5% (F1-score) compared to our TpT-ADE model. We performed another experiment wherein the weak negative samples are drawn from the set without filtering the entities that belong to *none* class. In this case, the F1-score further dropped by 7.2% compared to our TpT-ADE model.

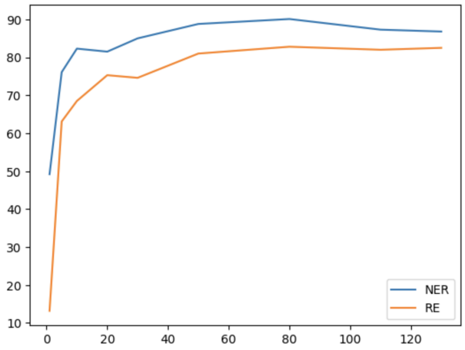

Figure 2: Negative Sampling Analysis

In our model, we chose the number of negative samples in case of both entity extraction and relation extraction ($\mathcal{E}^{ne} = \mathcal{E}^{nr}$) as 80 per sentence in the input sentence. As shown in Figure 2, if $\mathcal{E}^{ne} = \mathcal{E}^{nr} < 5$, the F1-score reaches to 68.2% and 53.7% for entity extraction and relation extraction, respectively. As the values of $\mathcal{E}^{ne}$ and $\mathcal{E}^{nr}$ increases, the model performs better. We observe that when $\mathcal{E}^{ne} = \mathcal{E}^{nr} > 80$, the performance of the model stagnates. Hence, we chose

$\mathcal{E}^{ne} = \mathcal{E}^{nr} = 80$.

5 Conclusion

In this paper, we proposed TpT-ADE, a two-phase transformer based model to improve the efficiency of ADE extraction from raw clinical text. Through various experiments, we have shown that finding the representative terms for the entities in the input text and combining the trained BERT embeddings with the POS tag embeddings of the modifier words of the entities to identify their intensities yield better results. In addition, using a simple shallow neural network and a strong negative sampling method in our model, showed considerable improvements over prior works.

References

Ghada Alfattni, Maksim Belousov, Niels Peek, Goran Nenadic, et al. 2021. Extracting drug names and associated attributes from discharge summaries: text mining study. *JMIR medical informatics*, 9(5):e24678.

Giannis Bekoulis, Johannes Deleu, Thomas Demeester, and Chris Develder. 2018a. Adversarial training for multi-context joint entity and relation extraction. *arXiv preprint arXiv:1808.06876*.

Giannis Bekoulis, Johannes Deleu, Thomas Demeester, and Chris Develder. 2018b. Joint entity recognition and relation extraction as a multi-head selection problem. *Expert Systems with Applications*, 114:34–45.

Juan Chen, Jie Hu, Tianrui Li, Fei Teng, and Shengdong Du. 2024. An effective relation-first detection model for relational triple extraction. *Expert Systems with Applications*, 238:122007.

Hong-Jie Dai, Chu-Hsien Su, and Chi-Shin Wu. 2020. Adverse drug event and medication extraction in electronic health records via a cascading architecture with different sequence labeling models and word embeddings. *Journal of the American Medical Informatics Association*, 27(1):47–55.

Bharath Dandala, Diwakar Mahajan, and Murthy V Devarakonda. 2017. Ibm research system at tac 2017: Adverse drug reactions extraction from drug labels. In *TAC*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Greg Durrett and Dan Klein. 2014. A joint model for entity analysis: Coreference, typing, and linking. *Transactions of the association for computational linguistics*, 2:477–490.

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

Ed-Drissiya El-Allaly, Mourad Sarrouti, Noureddine En-Nahnahi, and Said Ouatik El Alaoui. 2022. An attentive joint model with transformer-based weighted graph convolutional network for extracting adverse drug event relation. *Journal of biomedical informatics*, 125:103968.

Harsha Gurulingappa, Abdul Mateen Rajput, Angus Roberts, Juliane Fluck, Martin Hofmann-Apitius, and Luca Toldo. 2012. Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports. *Journal of biomedical informatics*, 45(5):885–892.

Hannaneh Hajishirzi, Leila Zilles, Daniel S Weld, and Luke Zettlemoyer. 2013. Joint coreference resolution and named-entity linking with multi-pass sieves. In *EMNLP*, pages 289–299. Citeseer.

Haorui He, Yuanzhe Ren, Zheng Li, and Jing Xue. 2022. Adaptive knowledge distillation for efficient relation classification. In *International conference on artificial neural networks*, pages 148–158. Springer.

Kai He, Yucheng Huang, Rui Mao, Tieliang Gong, Chen Li, and Erik Cambria. 2023. Virtual prompt pre-training for prototype-based few-shot relation extraction. *Expert Systems with Applications*, 213:118927.

Fei Li, Meishan Zhang, Guohong Fu, and Donghong Ji. 2017. A neural joint model for entity and relation extraction from biomedical text. *BMC bioinformatics*, 18(1):1–11.

Fei Li, Yue Zhang, Meishan Zhang, and Donghong Ji. 2016. Joint models for extracting adverse drug events from biomedical text. In *IJCAI*, volume 2016, pages 2838–2844.

Xinnian Liang, Shuangzhi Wu, Mu Li, and Zhoujun Li. 2021. Unsupervised keyphrase extraction by jointly modeling local and global context. *arXiv preprint arXiv:2109.07293*.

Zhaoran Liu, Haozhe Li, Hao Wang, Yilin Liao, Xinggao Liu, and Gaojie Wu. 2023. A novel pipelined end-to-end relation extraction framework with entity mentions and contextual semantic representation. *Expert Systems with Applications*, 228:120435.

Yi Luan, Dave Wadden, Luheng He, Amy Shah, Mari Ostendorf, and Hannaneh Hajishirzi. 2019. A general framework for information extraction using dynamic span graphs. *arXiv preprint arXiv:1904.03296*.

Youmi Ma, Tatsuya Hiraoka, and Naoaki Okazaki. 2022. Joint entity and relation extraction based on table labeling using convolutional neural networks. In *Proceedings of the sixth workshop on structured prediction for NLP*, pages 11–21.

Mark Neumann, Daniel King, Iz Beltagy, and Waleed Ammar. 2019. Scispacy: fast and robust models for biomedical natural language processing. *arXiv preprint arXiv:1902.07669*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Inigo Jauregi Unanue, Ehsan Zare Borzeshi, and Massimo Piccardi. 2017. Recurrent neural networks with specialized word embeddings for health-domain named-entity recognition. *Journal of biomedical informatics*, 76:102–109.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019. Entity, relation, and event extraction with contextualized span representations. *arXiv preprint arXiv:1909.03546*.

Qian Wan, Luona Wei, Shan Zhao, and Jie Liu. 2023. A span-based multi-modal attention network for joint entity-relation extraction. *Knowledge-Based Systems*, 262:110228.

An Wang, Ao Liu, Hieu Hanh Le, and Haruo Yokota. 2022. Towards effective multi-task interaction for entity-relation extraction: A unified framework with selection recurrent network. *arXiv preprint arXiv:2202.07281*.

Jue Wang and Wei Lu. 2020. Two are better than one: Joint entity and relation extraction with table-sequence encoders. *arXiv preprint arXiv:2010.03851*.

Yijun Wang, Changzhi Sun, Yuanbin Wu, Hao Zhou, Lei Li, and Junchi Yan. 2021. Unire: A unified label space for entity relation extraction. *arXiv preprint arXiv:2107.04292*.

Qiang Wei, Zongcheng Ji, Zhiheng Li, Jingcheng Du, Jingqi Wang, Jun Xu, Yang Xiang, Firat Tiryaki, Stephen Wu, Yaoyun Zhang, et al. 2020. A study of deep learning approaches for medication and adverse drug event extraction from clinical text. *Journal of the American Medical Informatics Association*, 27(1):13–21.

Jun Xu, Hee-Jin Lee, Zongcheng Ji, Jingqi Wang, Qiang Wei, and Hua Xu. 2017. Uth_ccb system for adverse drug reaction extraction from drug labels at tac-adr 2017. In *TAC*.

Zhiheng Yan, Chong Zhang, Jinlan Fu, Qi Zhang, and Zhongyu Wei. 2021. A partition filter network for joint entity and relation extraction. *arXiv preprint arXiv:2108.12202*.

Yue Zhang and Stephen Clark. 2008. Joint word segmentation and pos tagging using a single perceptron. In *Proceedings of ACL-08: HLT*, pages 888–896.

Suncong Zheng, Feng Wang, Hongyun Bao, Yuexing Hao, Peng Zhou, and Bo Xu. 2017. Joint extraction of entities and relations based on a novel tagging scheme. *arXiv preprint arXiv:1706.05075*.

Peng Zhou, Suncong Zheng, Jiaming Xu, Zhenyu Qi, Hongyun Bao, and Bo Xu. 2017. Joint extraction of multiple relations and entities by using a hybrid neural network. In *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data: 16th China National Conference, CCL 2017, and 5th International Symposium, NLP-NABD 2017, Nanjing, China, October 13-15, 2017, Proceedings 16*, pages 135–146. Springer.