PHEE: A Dataset for Pharmacovigilance Event Extraction from Text

Zhaoyue Sun[1], **Jiazheng Li**[1], **Gabriele Pergola**[1], **Byron C. Wallace**[2]
Bino John[3], **Nigel Greene**[3], **Joseph Kim**[3] and **Yulan He**[1,4,5]
[1]Department of Computer Science, University of Warwick
[2]Khoury College of Computer Sciences, Northeastern University
[3]AstraZeneca
[4]Department of Informatics, King's College London
[5]The Alan Turing Institute
{Zhaoyue.Sun, Jiazheng.Li, Gabriele.Pergola.1}@warwick.ac.uk
{bino.john, nigel.greene, joseph.kim1}@astrazeneca.com
b.wallace@northeastern.edu, yulan.he@kcl.ac.uk

Abstract

The primary goal of drug safety researchers and regulators is to promptly identify adverse drug reactions. Doing so may in turn prevent or reduce the harm to patients and ultimately improve public health. Evaluating and monitoring drug safety (i.e., *pharmacovigilance*) involves analyzing an ever growing collection of spontaneous reports from health professionals, physicians, and pharmacists, and information voluntarily submitted by patients. In this scenario, facilitating analysis of such reports via automation has the potential to rapidly identify safety signals. Unfortunately, public resources for developing natural language models for this task are scant. We present PHEE, a novel dataset for pharmacovigilance comprising over 5000 annotated events from medical case reports and biomedical literature, making it the largest such public dataset to date. We describe the hierarchical event schema designed to provide coarse and fine-grained information about patients' demographics, treatments and (side) effects. Along with the discussion of the dataset, we present a thorough experimental evaluation of current state-of-the-art approaches for biomedical event extraction, point out their limitations, and highlight open challenges to foster future research in this area[1].

1 Introduction

Pharmacovigilance is the pharmaceutical science that entails monitoring and evaluating the safety and efficiency of medicine use, which is vital for improving public health (World Health Organization, 2004). Unexpected adverse drug effects (ADEs) could lead to considerable morbidity and mortality (Lazarou et al., 1998). It has been reported that more than half of ADEs are preventable (Gurwitz et al., 2000). Pharmacovigilance is therefore important for detecting and understanding ADE-related events, as it may inform clinical practice and ultimately mitigate preventable hazards.

Collecting and maintaining the clinical evidence for pharmacovigilance can be difficult because it requires time-consuming manual curation to capture emerging data about drugs (Thompson et al., 2018). Much of this information can be found in unstructured textual data including medical literature, notes in electronic health records (EHR), and social media posts. Using NLP methods to discover and extract adverse drug events from unstructured text may permit efficient monitoring of such sources (Nikfarjam et al., 2015; Huynh et al., 2016; Ju et al., 2020; Wei et al., 2020).

Past work has introduced pharmacovigilance corpora to support training and evaluation of NLP approaches for ADE extraction. However, most of these datasets (e.g., the ADE corpus; Gurulingappa et al. 2012b) contain annotations only on entities (such as drugs and side effects) and their binary relations as shown in Figure 1(a). This ignores contextual information relating to human subjects, treatments administered, and more complex situations such as multi-drug concomitant use. To address this problem, Thompson et al. (2018) developed the PHAEDRA corpus, which includes annotations of not only drugs and side effects, but also subjects (human, specific species, bacteria, and so on) and events encoding descriptions of drug effects, which involve multiple arguments, and event attributes — see Figure 1(b).

Despite these refinements, however, PHAEDRA does not provide detailed, nested annotations such as dosages, conditions, and patient demographic details. This granular information may provide critical context to clinical studies. Furthermore, PHAEDRA consists of only 600 annotated abstracts of

[1]Our data and code is available at https://github.com/ZhaoyueSun/PHEE

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 5571 - 5587

medical case reports, making it challenging to train NLP models for pharmacovigilance events extraction since its annotations are in the document level and the actual annotated events are sparse.

In this work we introduce a new annotated corpus, PHEE, for adverse and potential therapeutic effect event extraction for pharmacovigilance study. The dataset consists of nearly 5,000 sentences extracted from MEDLINE case reports, and each sentence features two levels of annotations. With respect to coarse-grained annotations, each sentence is annotated with the event trigger word/phrase, event type and text spans indicating the event's associated subject, treatment, and effect. In a fine-grained annotation pass, further details are marked, such as patient demographic information, the context information about the treatments including drug dosage levels, administration routes, frequency, and attributes relating to events. An example annotation is shown in Figure 1(c).

Using PHEE as the benchmark, we conduct thorough experiments to assess the state-of-the-art NLP technologies for the pharmacovigilance-related event extraction task. We use sequence labelling and (both extractive and generative) QA-based methods as baselines and evaluate event trigger extraction and argument extraction. The extractive QA method performs best for trigger extraction with the exact match F1 score of 70.09%, while the generative QA method achieves the best exact match F1 score of 68.60% and 76.16% for the main argument and sub-argument extraction, respectively. Further analysis shows that current models perform well on average cases but often fail on more complex examples.

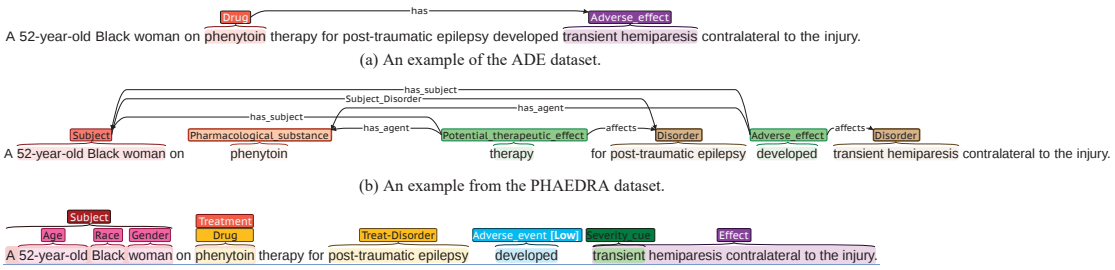
Our contributions can be summarised as follows: 1) We introduce PHEE, a new pharmacovigilance dataset containing over 5,000 finely annotated events from public medical case reports. To the best of our knowledge, this is the largest and most comprehensively annotated dataset of this type to date. 2) We collect hierarchical annotations to provide granular information about patients and conditions in addition to coarse-grained event information. 3) We conduct thorough experiments to compare current state-of-the-art approaches for biomedical event extraction, demonstrating the strength and weaknesses of current technologies and use this to highlight challenges for future research in this area.

2 Related Work

Pharmacovigilance Related Corpora Prior pharmacovigilance-related corpora mainly has focused on annotation of entities (e.g., drugs, diseases, medications) and binary relations between them, namely, drug-ADE relations (Gurulingappa et al., 2012a; Patki et al., 2014; Ginn et al., 2014), disorder-treatment relations (Rosario and Hearst, 2004; Roberts et al., 2009; Uzuner et al., 2011; Van Mulligen et al., 2012), and drug-drug interactions (Segura-Bedmar et al., 2011; Boyce et al., 2012; Rubrichi and Quaglini, 2012; Herrero-Zazo et al., 2013). More recent open challenges, including the 2018 n2c2 shared task (Henry et al., 2020) and MADE1.0 challenge (Jagannatha et al., 2019), have considered annotating additional relation types, such as drug-attribute and drug-reason relations, but they are still binary relationships.

Thompson et al. (2018) introduced the PHAEDRA corpus, extending the drug-ADE annotations to pharmacovigilance events. Compared to corpora that only annotate simple drug-ADE relations—referred to as *AE* events in PHAEDRA—they further annotate three additional relations, namely the *Potential Therapeutic Effect* (PTE) event which refers to the potential beneficial effects of drugs, the *Combination* and the *Drug-Drug Interaction* event which indicates multiple drug use and interactions between administered drugs, respectively. In addition, PHAEDRA includes the subject as a type of named entities (NEs) and annotates three types event attributes, i.e., negated, speculated and manner. However, some key informative details are still missing in PHAEDRA. As the NE annotation of PHAEDRA is usually a single noun or a short noun phrase, detailed information about the subject (such as age and gender), and of the medication (e.g., dosage and frequency) is not captured.

We set out to annotate a larger corpus with more detailed information to facilitate training of pharmacovigilance event extraction models. We build on existing corpora (PHAEDRA and ADE). The ADE corpus comprises ∼3,000 MEDLINE case reports and annotations on ∼4,000 sentences indicating adverse effects, but their annotations only involve drugs, dosages and adverse effects, and lack sufficient event details of interest. The PHAEDRA corpus reuses 227 abstracts from ADE and integrates an additional 370 abstracts (from other corpora and some novel entries). However, the PHAEDRA corpus is annotated at the document



(a) An example of the ADE dataset.

(b) An example from the PHAEDRA dataset.

(c) An example from our PHEE dataset.

Figure 1: Comparison of annotations from (a) the ADE corpus, (b) the PHEADRA corpus and (c) our developed PHEE corpus.

level, the actual annotated events are very sparse. We collected sentences in ADE and those in PHAEDRA with AE or PTE event annotations and enriched these using our proposed annotation scheme.

Biomedical Event Extraction Most existing biomedical event extraction methods work as "pipelines", treating trigger extraction and argument extraction as two stages (Björne and Salakoski, 2018; Li et al., 2018, 2020a; Huang et al., 2020; Zhu and Zheng, 2020); this can lead to error propagation. Trieu et al. (2020) propose an end-to-end model that jointly extracts the trigger/entity and assigns argument roles to mitigate the problem of error propagation, but in contrast to our span-based annotation, this requires full annotation of all entities. Ramponi et al. (2020) consider biomedical event extraction as a sequence labelling task, allowing them jointly model event trigger and argument extraction via multi-task learning.

In other domains, recent work has formulated event extraction as a *question answering* task (Du and Cardie, 2020; Li et al., 2020b; Liu et al., 2020). This new paradigm transforms the extraction of event trigger and arguments into multiple rounds of questioning, obtaining an answer about a trigger or an argument in each round. Such methods can reduce the reliance on the entity information for argument extraction and have proved to be data efficient. The current QA-based event extraction methods are mainly built on extractive QA which obtains the answer to a question by predicting the position of the target span in the original text. As such, a separate question needs to be formulated for different event and argument type. We also experiment with a generative QA method, which generates the answers directly, for comparison.

3 The PHEE Dataset

3.1 Task Definition and Schema

The PHEE corpus comprises sentences from biomedical literature annotated with information relevant to pharmacovigilance. Annotations are hierarchically structured in terms of textual *events*. Following prior work (Thompson et al., 2018), we define two main clinical event types: *Adverse Drug Effect (ADE)* and *Potential Therapeutic Effect (PTE)*, denoting potentially harmful and beneficial effects of medical therapies, respectively. Events consist of a *trigger* and several *arguments*, as defined by the ACE Semantic Structure (*LDC*, 2005). The trigger is a word or phrase that best indicates the occurrence of an event (e.g., 'induced', 'developed'), while the arguments specify the information characterizing an event, such as patient's demographic information, treatments, and (side-)effects (Figure 1(c)). We further organise arguments into two hierarchical levels, namely main and sub-arguments. Main arguments are longer text spans that contain the full description of an event aspect (e.g., *treatment*), while sub-arguments are usually words or short phrases included in main argument spans and highlighting specific details of the argument (e.g., *drug*, *dosage*, *duration*, etc).

More specifically, in PHEE, event arguments are defined as:

Subject highlights the patients involved in the medical event, with sub-arguments including *age, gender, race, number of patients* (labeled as *population*) and *preexisting conditions* (labeled as *subject.disorder*) of the subject.

Treatment describes the therapy administered to the patients, with sub-arguments specifying *drug* (and their combinations), *dosage*, *frequency, route, time elapsed*, *duration* and the

target disorder (labeled as *treatment.disorder*) of the treatment.

Effect indicates the outcome of the treatment.

We also collected annotations indicating three types of *attributes* characterizing whether an event is *negated*, *speculated* or its *severity* is indicated. See more details about the schema in Appendix A.

3.2 Data Collection and Validation

Data Collection To compose the PHEE corpus, we collect existing medical case report abstracts from the ADE (Gurulingappa et al., 2012b) and PHAEDRA (Thompson et al., 2018) datasets. We extract sentences from the abstracts and annotate them containing at least one adverse or therapeutic effect (*ADE* or *PTE*) event, for a total of over 4.8k sentences after deduplication.

Annotation Process We hired 15 annotators in total to participate in our annotation, who are PhD students in the computer science or medical domain. We consulted our annotation schema with pharmacovigilance researchers and biomedical NLP researchers before starting the annotation.

We conducted the corpus annotation through two stages to reduce the difficulty in dealing with medical text. In the first stage, we provided the annotators with sets of single sentences and asked them to highlight the event triggers and the text spans functioning as main arguments (i.e., *subject*, *treatment* and *effect*). Each annotator annotates about 330 sentences during this stage. In the next stage, we randomly assigned the annotated sentences to different annotators who were required to verify the correctness of the previous annotations. Once confirmed, the annotations were expanded specifying the possible sub-arguments (e.g., for *subjects*: *age*, *gender*, *population*, *race*, *subject.disorder*), and attributes (e.g., *negation*). To ease the cognitive demand required to highlight fine-grained sub-arguments during the second stage, the annotators were split into three groups, each specialising in just one of the three main argument types. Specifically, four annotators are allocated for subject sub-argument annotation and four for effect and attribute annotation, while seven annotators are allocated for treatment sub-argument annotation due to the task complexity. Each annotator is responsible for around 1.4k or 700 instances during this stage. Additional notes on the annotation process can be found in the Appendix B.

Data Validation To ensure quality annotations, each stage of annotation was proceeded by several rounds of annotation trials, after which we discussed frequent inconsistencies. When questions about specific instances surfaced during the annotation process, annotators flagged these sentences for review. While the main annotations of stage one were double-checked by the annotators in stage two, we randomly duplicated 20% of the stage-two samples and assigned them to different groups to measure Inter-Annotator Agreement (IAA).

We compute F1-score[2] as a measure of agreement between annotators. We calculate F1 scores between the sets of duplicated cases by (arbitrarily) selecting one annotation set as a "reference" to the other. Specifically, we adopted the EM_F1 (at span-level) and Token_F1 (at token-level) metric which are explained in details in Section 4.2. We report agreement scores in Table 1.

Consistency across trigger and argument types is over 80%, indicating the effectiveness of two-stage approaches. Agreement on sub-arguments is lower, which is expected due to the higher complexity of fine-grained medical annotations. In particular, we notice a difficulty in consistency over the annotation of *duration* and *time_elapsed*. One type of common inconsistent cases is "generalized expressions" (e.g., "chronic", "long-term", "shortly after"), which are annotated by some annotators but ignored by others. In addition, it is easy for annotators to confuse these two types of annotation. For example, the phrase "48 months" in "48 months postchemotherapy" is mistakenly annotated to be *duration*, which, however, is generally believed should be *time_elapsed*. Other less inconsistent sub-argument types including *frequency* and *subject.disorder*. For *frequency*, inconsistent cases including generalized expressions (e.g., "repeated", "continuous") and certain specific expressions such as "0.32mg/kg/day" that some annotators prefer to annotate "0.32mg/kg" as *dosage* and "/day" as *frequency* while others prefer to annotate the whole span as *dosage*. For *subject.disorder*, conflicts exist in "neutral" expressions that describe the subject's health condition but not necessarily to be a disorder, such as "pregnant" and "nondiabetic". Apart from the difficult cases, inconsistency also occurs in the

[2]Traditional Cohen's Kappa as IAA evaluation is not applicable for span-level computation due to an unknown number of negative cases. We therefore follow previous work (Thompson et al., 2018; Gurulingappa et al., 2012b) choosing the F1 score as the more relevant IAA measurement.

	EM_F1	Token_F1
Trigger	88.17	88.81
Main-argument	84.57	90.14
Sub-argument	80.25	83.24
Attribute	46.41	48.04

Table 1: Inter-Annotator Agreement (IAA).

	Train	Dev	Test	Total
# Sentences	2,898	961	968	4,827
# Events	3,006	1,003	1,010	5,019
- ADE	2,710	886	889	4,485
- PTE	296	117	121	534

Table 2: Dataset statistics on train/dev/test sets.

choices of span boundaries, especially for long arguments, or sometimes due to accident mistakes. Attribute annotations are inconsistent, probably due to their rarity in the corpus.

3.3 Dataset Statistics and Analysis

PHEE includes a total of 4,827 sentences and 5,019 annotated events. This makes PHEE the largest annotated dataset on adverse drug events of which we are aware. We randomly divided train, dev, and test splits based on documents. Details about these splits are provided in Table 2.

Table 3 reports statistics of the main event arguments. In general, each event contains at most one main argument of a particular type, but arguments might be discontinuous, leading to multiple spans representing a single argument. The average number of tokens per argument is about 3-4, which is generally longer than other datasets focusing only on biomedical entities (drugs, diseases or effects).

Statistics about sub-argument annotations are provided in Figure 2. For the sub-arguments of the *subject*, *age* is the most frequently mentioned feature. *Gender*, *population* and *subject.disorder* are also comparatively common; *race* is the rarest attribute. For *treatment*, *drug* names are the most frequently mentioned, even higher than the number of *treatment* arguments due to the administered combinations of drugs. The target *disorder* of the treatment is the second most mentioned, providing context information in which the therapeutic or adverse events occurred. In contrast, the other treatment's sub-arguments occur less frequently, resulting in a rather imbalance argument distribution.

Statistics of attributes are in Appendix C.

	# ann.	# spans	# ann./sentence	avg. tokens/ann.
Subject	2,424	2,502	0.50	3.95
Treatment	5,018	5,329	1.04	3.25
Effect	4,593	4,871	0.95	3.67

Table 3: Main argument statistics.

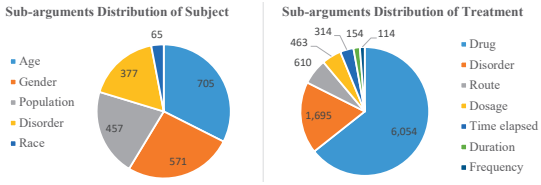


Figure 2: Distribution of sub-arguments.

4 Experiments

We conduct evaluation of sequence labelling and QA-based methods (both extractive and generative) on our PHEE dataset. We describe our experimental design, evaluation metrics, and main results in this section. Reproduction details are in Appendix D.

4.1 Models

Sequence Labelling Given a sentence $S_i = \{x_1, x_2, ...x_j, ..., x_n\}$, we encode event structures using token-level labels $Y_i = \{y_1, y_2, ...y_j, ..., y_n\}$. We use the "I-O" scheme, in which the label "I-X" indicates a token is within a span of argument type X, and "O" indicates it is outside of any argument span. As the main arguments and their associated sub-arguments usually overlap, we set the label to be "I-A.B" if the token is in a main argument span of type A and a sub-argument span of type B simultaneously. Correspondingly, the label will be "I-A" or "I-B" if the token only appears in a main argument or a sub-argument. For triggers, labels denote event types. An example of the flattened label sequence is shown in Figure 3(a).

We use the ACE (Wang et al., 2021) model, which reaches state-of-the-art results for Named Entity Recognition (NER), as a representative sequence labelling method in our experiments.

Extractive QA We build our extractive QA model upon the EEQA method (Du and Cardie, 2020). Event triggers, main arguments, and sub-arguments are extracted in three sequential steps as shown in Figure 3(b). We fine-tune the pre-trained BioBERT (Lee et al., 2020) base model on our dataset. In particular, the input is a sentence paired with a question, formatted as: '`[CLS] <question> [SEP] <sentence>`', where

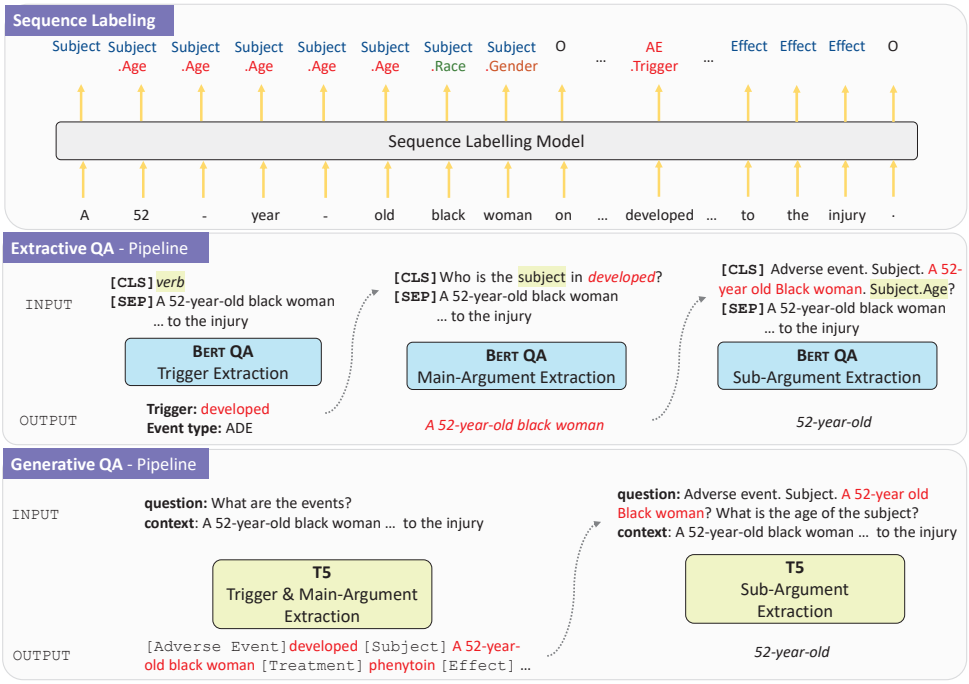


Figure 3: Illustrations of three baseline methods with the example: "*A 52-year-old Black woman on phenytoin therapy for post-traumatic epilepsy developed transient hemiparesis contralateral to the injury.*" The diagram shows the extraction of the *subject* for main argument extraction and the *age* of the subject for sub-argument extraction.

`<question>` and `<sentence>` are placeholders for a question template and an input sentence, respectively. The output is the text span extracted from the input sentence as answers. We experiment with different question templates for event triggers, main arguments and sub-arguments.

For event trigger extraction, the model predicts a probability distribution across all events types (including a non-event case) for each input token based on BioBERT representations. Argument extraction is done for each argument type, where the probabilities of being the start/end position of an argument span are predicted for each token by a classification layer added on top of the BioBERT encoder. All possible <start, end> pairs are then filtered by thresholds of scores of the [CLS] token (which indicates a non-event prediction) to retrieve the extracted arguments. We also filter out spans that overlap with other spans with better scores. We train the QA models for main-argument extraction and sub-argument extraction separately.

Generative QA Under the generative QA setting, we split the event extraction task into two stages. In the first, event triggers and main arguments are extracted simultaneously. In the second, sub-arguments are extracted. We fine-tune SciFive (PMC; Phan et al. 2021) model, a T5 model pre-trained on Pubmed. An example of the input/output in the QA pipeline is shown in Figure 3(c).

For the first stage, the question is simply '*What are the events?*'. Each sentence is paired with the question in the form of '`question: <question> context: <sentence>`', where `question:` and `context:` are the fixed prompts, and `<question>` and `<sentence>` are placeholders for a pre-defined question and an input sentence, respectively. The gold-standard answer is constructed using a template '`[<event type>] <trigger> [<main argument type>] <main argument content> ...`', where <·> is a placeholder to be replaced by the relevant content. For each event, the trigger comes first, followed by the main arguments in the order of *subject*, *treatment* and *effect*. Multiple events are flattened into a sequence. The QA model then generates answers from which we obtain the event type, event trigger, and main argument spans via pattern matching. For the second stage, we use the questions defined for sub-arguments in extractive QA. The model input and gold-standard answers are formulated in a similar way as the first stage.

4.2 Evaluation Metrics

We evaluate model performance on event trigger extraction and argument extraction separately. Punc-

tuation and articles are ignored during evaluation.

Event trigger extraction Following Lin et al. (2020), we use the F1 metric for the evaluation of event trigger identification and event trigger classification. Specifically, *trigger identification* (Trig-I) evaluates how well the trigger words match their corresponding references; *trigger classification* (Trig-C) evaluates not only the mentioning words but also event types. As the event trigger words could be ambiguous even for humans, and the detection of the presence of an ADE or PTE event is arguably more important, we further compute the *event classification* (Event-C) F1 score, which evaluates whether the event type of a trigger word matches its reference.

Argument extraction Argument evaluation is also conducted from both *identification* and *classification* perspectives. Specifically, an argument span is correctly *identified* if its event type and offsets match a gold-standard span, and it is correctly *classified* if the argument type also matches. Considering that argument spans could be long and the exact match (i.e., span-level) evaluation might be too strict, we additionally report token-level evaluation results. Specifically, *EM_F1* measures the percentage of the predicted spans that match the ground truth spans exactly and *Token_F1* measures the average token overlap between the predictions and references. As there might be multiple spans for each argument, we compute both metrics by micro-averaging. That is, we accumulate the number of matched spans (or tokens) across the corpus as the True Positive (TP) value, and compute the precision, recall and F1 accordingly.

4.3 Results and Analysis

We compare three families of baselines on the PHEE dataset. For the extractive QA and the generative QA approaches, we explored several question templates and report only the results of templates which perform best on the development set. A more extensive analysis on different template formats is discussed in Appendix E.

4.3.1 Evaluation: Trigger Extraction

Table 4 reports the performance of the three baselines on trigger extraction. Extractive QA achieves the best result on both trigger identification and classification. However, it is worth mentioning that due to the EEQA design, only the first token of the trigger could be used for training and evaluation. Nevertheless, the comparison is still relevant as the trigger has the only linguistic function of representing an event occurrence but a limited semantic content. Instead, the generative QA model obtained the best comparative performance when classifying the event type(s) of the whole sentence, independently of the particular trigger extracted.

	Trig-I	Trig-C	Event-C
Seq Labelling	67.98	67.98	90.69
Extractive QA	**70.71**	**70.09**	85.61
Generative QA	68.25	68.25	**95.16**

Table 4: Results for trigger extraction.

	Main-arguments		Sub-arguments	
	EM_F1	Token_F1	EM_F1	Token_F1
Argument Identification				
Sequence Labelling	59.85	73.98	70.75	73.12
Extractive QA	65.87	77.00	66.71	69.97
Generative QA	**68.85**	**81.63**	**77.33**	**78.38**
Argument Classification				
Sequence Labelling	59.61	73.16	68.88	69.31
Extractive QA	65.70	75.92	64.98	66.69
Generative QA	**68.60**	**80.04**	**76.16**	**76.10**

Table 5: Results for arguments extraction.

4.3.2 Evaluation: Argument Extraction

We present the main argument and sub-argument extraction results in Table 5. Generative QA achieves the best results in both main argument and sub-argument extraction. Extractive QA performs better than sequence labelling in main argument extraction, but worse in sub-argument extraction. We sampled and analysed the error cases of the three approaches, and present some of them in Table A5.

In particular, for main argument extraction, we observe that a common error of extractive QA is the failure of detecting an event trigger in an early stage, making it skip extracting main arguments in the subsequent stages. Generative QA performs better probably because it extracts the trigger and main arguments simultaneously thus avoiding the problem of error propagation. For sequence labelling, the most prominent problem is the incompleteness of the extracted main arguments, especially for the *subject* argument. One possible reason is that the main argument and sub-argument labels are flattened into one sequence, which results in the loss of the information about the relations between the main argument and its sub-arguments, therefore hurting the extraction performance.

	Seq Labelling		Extractive QA		Generative QA	
	EM_F1	Token_F1	EM_F1	Token_F1	EM_F1	Token_F1
Subject	39.72	66.57	64.26	75.13	**68.72**	**80.77**
- Age	82.91	88.06	82.07	89.68	**89.25**	**93.48**
- Disorder	**30.26**	**38.99**	21.36	25.17	24.19	33.07
- Gender	82.61	83.03	84.05	84.50	**88.73**	**89.13**
- Population	66.28	63.55	65.79	64.89	**75.90**	**80.37**
- Race	75.00	77.78	85.71	80.00	**93.33**	**87.50**
Treatment	58.85	70.28	61.35	71.97	**63.66**	**75.80**
- Drug	78.63	79.48	74.63	75.07	**85.28**	**85.28**
- Disorder	58.93	63.30	54.19	62.62	**65.68**	**70.77**
- Route	63.06	67.32	63.40	69.87	**72.48**	**77.13**
- Dosage	50.52	56.95	57.95	62.47	**63.10**	**71.36**
- Time elapsed	**45.21**	**60.57**	39.68	54.49	38.02	58.00
- Duration	22.54	42.67	27.59	37.70	**30.77**	**45.26**
- Frequency	36.92	42.62	**54.55**	**53.85**	51.16	50.67
- Combination.Drug	60.71	59.74	34.46	39.55	**69.11**	**67.62**
Effect	70.75	79.72	71.21	80.26	**74.00**	**83.63**

Table 6: Classification results for each argument type. Best results for each argument type are highlighted in bold.

For the sub-argument extraction, the performance of the extractive QA drops to the worst, probably due to further error propagation from the previous two stages. For the other two methods, the sequence labelling method seems to be more severely affected by trigger extraction errors. In some cases, the argument spans are matched, but no trigger is detected in the sentence, thus leading to a failure. The generative QA method's performance at this stage is relatively less influenced by the main argument extraction results compared to the other two approaches, but we notice it could easily fail to extract less frequent sub-arguments. One possible downside of generative QA models when used for information extraction is that they may generate tokens not in the original input sentence, but in our sampled cases, such errors are very rare.

4.3.3 Evaluation for Each Argument Type

In Table 6, we present the results for each argument type. Firstly, among all main argument types, the *effect* seems to be the easiest one to be extracted. This is probably due to its abundant occurrences and relatively distinct features compared to other argument types. Although the *treatment* also occur frequently in the corpus, models perform much poorer on *treatment* extraction. The main reason is that the length of the treatment spans varies, and the information of the treatment could be more complex, which leads to the fragmented extraction results. The *subject* while appearing less frequently than *treatment* and *effect*, have relatively simpler linguistic patterns. As such, their exaction results are better than *treatment* when using QA models.

For the sub-arguments, highly frequent arguments with simpler linguistic patterns such as *age, gender*, and *drug* get promising results. Some arguments with relatively limited expressions, such as *race* and *frequency*, although very rare in our dataset, still have merit or moderate extraction result. Some sub-arguments such as *subject.disorders* and *treatment.disorders*, can be confusing even for human annotators, getting relatively low extraction performance. Models' performance on *subject.disorder* is even poorer due to its less occurrence in the dataset. Another pair of arguments that are easily confused is *time elapsed* and *duration*, both of which contain temporal expressions. Combined with the low occurrence frequencies, these two arguments also get quite low extraction results.

5 Challenges and Future Directions

Our analysis of experimental results, suggests the following open challenges for the extraction of pharmacovigilance events. Firstly, the models perform poorly on arguments with similar entity mentions but different argument roles. For example, a disease mentioned in text could be annotated as *treatment.disorder* if it is the target of the treatment or *subject.disorder* if it refers to someone's disease but not targeted for treatment. A similar problem can be observed for arguments of temporal expressions such as *time elapsed* and *frequency*. The poor performance on such arguments seems to indicate that existing models are not able to perform deep semantic analysis. Additional constraints encoding linguistic constructs between entity mentions and

main argument types could be explored to guide the event extraction model through, for example, posterior regularisation.

Secondly, the models' performance on argument types with limited annotated training instances deteriorates drastically. One path forward is therefore to explore efficient few-shot learning strategies to improve models' generalisability on rare argument types. Also, there might exist corpora annotated with similar argument roles but for different purposes, for example, the corpora for medication extraction (Jagannatha et al., 2019) where drug dosage and frequency are annotated. It is possible to leverage external drug or disease knowledge through knowledge distillation.

Finally, none of the existing models cope well with the presence of multiple events in a sentence. This is mainly because existing annotations rely heavily on event triggers to differentiate events and require explicit linking between arguments and their respective event triggers. However, trigger identification itself is ambiguous and difficult even for human annotators. In some cases, multiple events could share the same trigger. For pipeline-based models, i.e., the QA models in our work, detection of multiple triggers is prone to error, thus making it hard for subsequent argument extraction due to error propagation. For the sequence labelling model, it is difficult to flatten the annotations of multiple events into a single label sequence. We thus duplicate the multi-event cases during training, and only provide a single event annotation for each case at one time. However, it becomes impossible to obtain full extract results for multiple events during the inference stage. In the future, rather than sequence labelling or QA-based extraction approaches, it is worth exploring graph-based approaches for multi-event extraction in which entity mentions are nodes in the graph while event extraction can be framed as soft clustering of entity mentions.

6 Conclusion

In this paper, we present the development of a novel corpus, PHEE, composed of sentences from the medical case reports annotated with pharmacovigilance-related events. Events in PHEE are hierarchically annotated with coarse and fine-grained information about patient demographics, treatments, and (side) effects. We use it to evaluate state-of-the-art NLP models for pharmacovigilance event extraction. Experimental results show that current models could capture reasonable information in common cases but face challenges for complex situations such as distinguishing semantically-similar arguments, dealing with the low resource setting, and extracting multi-events from text.

Limitations

This study has several limitations. First, despite the implementation of a quality control process, the collected annotations inevitably have some quality issues. For example, to reduce cognitive load, we split the annotation process into two stages and required annotators working on the second-stage annotation to check and correct first-stage annotations. However, we noticed that many annotators are more willing to keep the previous annotations as they are unless the errors can be easily identified. This may lead to inflated IAA results.

Also, although trained for the task, the lack of medical background of annotators may have some impact on the quality of the dataset. Second, our dataset only contains two event types, Adverse Drug Event (ADE) and Potential Therapeutic Effect (PTE). It is worth considering adding sentences with the *null* event type, that is, not associated with ADE or PTE. Furthermore, only one base PLM model for each baseline was chosen in our experiments, and more encoding methods are worth exploring in the future. Finally, although we have provided the annotations of event attributes such as *speculation*, *negation* and *severity*, we have not implemented baseline models for event attribution detection, partly due to the few annotated cases. In the future, we will explore semi-supervised learning approaches for event attribute detection.

Ethics

We used abstracts of publicly published medical reports as data sources in which no patient sensitive information is revealed. It should be noted that the adverse events and potential therapeutic events presented in this work are only based on textual-level information extraction and do not necessarily indicate any causal relations between drugs and effects. Causality assessment for pharmacovigilance should follow a rigorous assessment framework such as the assessment criteria of various causality categories defined in the WHO-UMC system[3].

[3] https://who-umc.org/media/164200/who-umc-causality-assessment_new-logo.

Acknowledgements

This work was supported in part by the UK Engineering and Physical Sciences Research Council (grant no. EP/T017112/1, EP/V048597/1, EP/X019063/1), and the National Science Foundation (NSF) grant 1750978. YH is supported by a Turing AI Fellowship funded by the UK Research and Innovation (grant no. EP/V020579/1).

References

Jari Björne and Tapio Salakoski. 2018. Biomedical event extraction using convolutional neural networks and dependency parsing. In *Proceedings of the BioNLP 2018 workshop*, pages 98–108.

Richard D Boyce, Gregory Gardner, and Henk Harkema. 2012. Using natural language processing to extract drug-drug interaction information from package inserts. In *BioNLP: Proceedings of the 2012 Workshop on Biomedical Natural Language Processing*, pages 206–213.

Xinya Du and Claire Cardie. 2020. Event extraction by answering (almost) natural questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 671–683.

Rachel Ginn, Pranoti Pimpalkhute, Azadeh Nikfarjam, Apurv Patki, Karen O'Connor, Abeed Sarker, Karen Smith, and Graciela Gonzalez. 2014. Mining twitter for adverse drug reaction mentions: a corpus and classification benchmark. In *Proceedings of the fourth workshop on building and evaluating resources for health and biomedical text processing*, pages 1–8. Citeseer.

Harsha Gurulingappa, Abdul Mateen-Rajpu, and Luca Toldo. 2012a. Extraction of potential adverse drug events from medical case reports. *Journal of biomedical semantics*, 3(1):1–10.

Harsha Gurulingappa, Abdul Mateen Rajput, Angus Roberts, Juliane Fluck, Martin Hofmann-Apitius, and Luca Toldo. 2012b. 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.

Jerry H Gurwitz, Terry S Field, Jerry Avorn, Danny McCormick, Shailavi Jain, Marie Eckler, Marcia Benser, Amy C Edmondson, and David W Bates. 2000. Incidence and preventability of adverse drug events in nursing homes. *The American journal of medicine*, 109(2):87–94.

Sam Henry, Kevin Buchan, Michele Filannino, Amber Stubbs, and Ozlem Uzuner. 2020. 2018 n2c2 shared task on adverse drug events and medication extraction in electronic health records. *Journal of the American Medical Informatics Association*, 27(1):3–12.

María Herrero-Zazo, Isabel Segura-Bedmar, Paloma Martínez, and Thierry Declerck. 2013. The ddi corpus: An annotated corpus with pharmacological substances and drug–drug interactions. *Journal of biomedical informatics*, 46(5):914–920.

Kung-Hsiang Huang, Mu Yang, and Nanyun Peng. 2020. Biomedical event extraction with hierarchical knowledge graphs. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1277–1285.

Trung-Tin Huynh, Yulan He, Alistair Willis, and Stefan M. Rüger. 2016. Adverse drug reaction classification with deep neural networks. In *COLING*.

Abhyuday Jagannatha, Feifan Liu, Weisong Liu, and Hong Yu. 2019. Overview of the first natural language processing challenge for extracting medication, indication, and adverse drug events from electronic health record notes (made 1.0). *Drug safety*, 42(1):99–111.

Meizhi Ju, Nhung TH Nguyen, Makoto Miwa, and Sophia Ananiadou. 2020. An ensemble of neural models for nested adverse drug events and medication extraction with subwords. *Journal of the American Medical Informatics Association*, 27(1):22–30.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Jason Lazarou, Bruce H Pomeranz, and Paul N Corey. 1998. Incidence of adverse drug reactions in hospitalized patients: a meta-analysis of prospective studies. *Jama*, 279(15):1200–1205.

Linguistic Data Consortium LDC. 2005. English annotation guidelines for events.

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.

Diya Li, Lifu Huang, Heng Ji, and Jiawei Han. 2020a. Biomedical event extraction based on knowledge-driven tree-lstm. In *Proc. 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT2019)*.

Fayuan Li, Weihua Peng, Yuguang Chen, Quan Wang, Lu Pan, Yajuan Lyu, and Yong Zhu. 2020b. Event extraction as multi-turn question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 829–838.

Lishuang Li, Yang Liu, and Meiyue Qin. 2018. Extracting biomedical events with parallel multi-pooling

pdf

convolutional neural networks. *IEEE/ACM transactions on computational biology and bioinformatics*, 17(2):599–607.

Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020. A joint neural model for information extraction with global features. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7999–8009.

Jian Liu, Yubo Chen, Kang Liu, Wei Bi, and Xiaojiang Liu. 2020. Event extraction as machine reading comprehension. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1641–1651.

Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Azadeh Nikfarjam, Abeed Sarker, Karen O'connor, Rachel Ginn, and Graciela Gonzalez. 2015. Pharmacovigilance from social media: mining adverse drug reaction mentions using sequence labeling with word embedding cluster features. *Journal of the American Medical Informatics Association*, 22(3):671–681.

Apurv Patki, Abeed Sarker, Pranoti Pimpalkhute, Azadeh Nikfarjam, Rachel Ginn, Karen O'Connor, Karen Smith, and Graciela Gonzalez. 2014. Mining adverse drug reaction signals from social media: going beyond extraction. *Proceedings of BioLinkSig*, 2014:1–8.

Long N Phan, James T Anibal, Hieu Tran, Shaurya Chanana, Erol Bahadroglu, Alec Peltekian, and Grégoire Altan-Bonnet. 2021. Scifive: a text-to-text transformer model for biomedical literature. *arXiv preprint arXiv:2106.03598*.

Alan Ramponi, Rob van der Goot, Rosario Lombardo, and Barbara Plank. 2020. Biomedical event extraction as sequence labeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5357–5367.

Angus Roberts, Robert Gaizauskas, Mark Hepple, George Demetriou, Yikun Guo, Ian Roberts, and Andrea Setzer. 2009. Building a semantically annotated corpus of clinical texts. *Journal of biomedical informatics*, 42(5):950–966.

Barbara Rosario and Marti A Hearst. 2004. Classifying semantic relations in bioscience texts. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 430–437.

Stefania Rubrichi and Silvana Quaglini. 2012. Summary of product characteristics content extraction for a safe drugs usage. *Journal of Biomedical Informatics*, 45(2):231–239.

Isabel Segura-Bedmar, Paloma Martinez, and Cesar de Pablo-Sánchez. 2011. Using a shallow linguistic kernel for drug–drug interaction extraction. *Journal of biomedical informatics*, 44(5):789–804.

Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 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*, pages 102–107.

Paul Thompson, Sophia Daikou, Kenju Ueno, Riza Batista-Navarro, Jun'ichi Tsujii, and Sophia Ananiadou. 2018. Annotation and detection of drug effects in text for pharmacovigilance. *Journal of cheminformatics*, 10(1):1–33.

Hai-Long Trieu, Thy Thy Tran, Khoa NA Duong, Anh Nguyen, Makoto Miwa, and Sophia Ananiadou. 2020. Deepeventmine: end-to-end neural nested event extraction from biomedical texts. *Bioinformatics*, 36(19):4910–4917.

Özlem Uzuner, Brett R South, Shuying Shen, and Scott L DuVall. 2011. 2010 i2b2/va challenge on concepts, assertions, and relations in clinical text. *Journal of the American Medical Informatics Association*, 18(5):552–556.

Erik M Van Mulligen, Annie Fourrier-Reglat, David Gurwitz, Mariam Molokhia, Ainhoa Nieto, Gianluca Trifiro, Jan A Kors, and Laura I Furlong. 2012. The eu-adr corpus: annotated drugs, diseases, targets, and their relationships. *Journal of biomedical informatics*, 45(5):879–884.

Xinyu Wang, Yong Jiang, Nguyen Bach, Tao Wang, Zhongqiang Huang, Fei Huang, and Kewei Tu. 2021. Automated concatenation of embeddings for structured prediction. 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 2643–2660.

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.

World Health Organization. 2004. Pharmacovigilance: ensuring the safe use of medicines. Technical report, World Health Organization.

Lvxing Zhu and Haoran Zheng. 2020. Biomedical event extraction with a novel combination strategy based on hybrid deep neural networks. *BMC bioinformatics*, 21(1):1–12.

Appendix

A Annotation Schema

We present a full list of the definitions of all items we annotated.

Event An event is annotated with an event *trigger*, several *arguments* and *attributes* (if any). We annotate the following types of events:

Adverse Drug Effect: The use of a drug or combination of drugs cause a harmful effect on the human patient.

Potential Therapeutic Effect: The use of a drug or combination of drugs bring a potential beneficial effect on the human patient.

Combination (sub-event): More than one drug is treated for the patient. The combination subevent consists of a trigger and several drug arguments. It usually plays the role of a subargument under the *treatment* argument of an ADE/PTE.

Arguments An argument describes the information characterizing an event.

Subject: highlights the patients involved in the medical event. Sub-arguments of *subject* are:

- ***Subject.Age:*** concrete age or span that indicates an age range.
- ***Subject.Gender:*** the span that indicates the subject's gender.
- ***Subject.Population:*** the number of patients receiving the treatment.
- ***Subject.Race:*** the span that indicates the subject's race/nationality.
- ***Subject.Disorder:*** preexisting conditions, i.e., disorders that the subject suffers other than the target disorder of the treatment.

Treatment: describes the therapy administered to the patients.

- ***Treatment.Drug:*** drugs used as therapy in the event.
- ***Treatment.Dosage:*** the amount of the drug is given.
- ***Treatment.Frequency:*** the frequency of drug use.
- ***Treatment.Route:*** the route of drug administration.
- ***Treatment.Time_elapsed:*** the time elapsed after the drug was administered to the occurrence of the (side) effect.
- ***Treatment.Duration:*** how long the patient has been taking the medicine (usually for long-term medication).
- ***Treatment.Disorder:*** the target disorder of the medicine administration.

Effect: indicates the outcome of the treatment.

Attribute *Attributes* interpret certain properties of events, i.e., indicating whether an event is *negated* or *speculated*, and the *severity* level of the event.

negated: the attribute *negated* denotes whether or not there is any textual cues showing the event is negated, i.e., for ADE, the adverse effect does not exist; or for PTE, the therapy is ineffective.

speculated: the attribute *speculated* indicates if there is any uncertain or speculation as to whether an event will actually happen. Considering the speculative nature of the medical case reports, we only annotate a *speculated* attribute when the speculative attitude of the author is explicitly remarked.

severity: the attribute *severity* refers to the severity level of the adverse effect. For example, the fatal effect is a 'high severity', while a minor symptom could be a 'low severity'. In general, we do not annotate 'severity' for PTE events.

B Annotation Process Supplement

To facilitate annotation we used *brat* (Stenetorp et al., 2012), a web-based tool. Annotators were instructed to mark trigger and argument spans as *brat-entities*, and discontinuous spans as *brat-fragments*. As for the case where a sentence contains multiple events, each argument will be connected to the trigger of its corresponding event with *brat-links*.

All annotators are volunteering and paid by the hour. In total we hired 15 annotators that spent around 20 and 30 hours per person on the first and the second stage of annotation, respectively.

C Statistics of Attributes

The occurrence of attributes is relatively rare in our dataset, and their statistics are illustrated in Table A1. Specifically, although severity annotations,

# Speculated	# Negated	# Severity
633	412	76

Table A1: Distribution of event attributes.

which refer to the severity level of the adverse effect, are helpful in adverse effect vigilance, it can be seen that there are very few mentions of this attribute in the dataset.

D Training Details and Hyperparameter Setting

Experiments – Sequence Labelling For sequence labelling experiments, we use the code of ACE[4](Wang et al., 2021). ACE develops a neural architecture search algorithm to automatically find better concatenations of transformer-based embeddings. With limited computing resources, we choose BERT (Base Cased, 1.1M parameters; Kenton and Toutanova 2019) and BioBERT (Base v1.1, 1.1M parameters; Lee et al. 2020) as the base embeddings. We firstly fine-tune the BERT and BioBERT separately on our data, and run the ACE algorithm with fixed fine-tuned embeddings. We follow the default released hyper-parameter setting of the ACE algorithm. When fine-tuning the embeddings, the AdamW (Loshchilov and Hutter, 2018) optimizer is used with a learning rate of 5×10^{-5} and the model is trained for 20 epochs. For training the ACE controller, we use the Stochastic Gradient Descent (SGD) optimizer with an initial learning rate of 0.1. The controller will anneal the learning rate by 0.5 if there is no improvement on the development set for 5 epochs. The batch sizes for both embedding fine-tuning and controller training are 32. We run the experiments on an NVIDIA TITIAN RTX GPU. The PLM fine-tuning costs about 2 hours for each model, while the ACE controller training costs about one day for a maximum of 150 epochs run.

Experiments – Extractive QA For extractive QA experiments, we fine-tune the EEQA(Du and Cardie, 2020) model on our data[5]. We use the BioBERT (Base Cased, 1.1M parameters; Kenton and Toutanova 2019) as the base model. The SGD algorithm is used as the optimizer, and the learning rate is set to 1×10^{-5}, 5×10^{-5}, 5×10^{-5} for trigger extraction, main argument extraction and sub-argument extraction, respectively. We use a batch size of 32 for trigger extraction and 16 for argument extraction. We set the maximum training epochs to 10. Experiments are conducted on an NVIDIA TITIAN RTX GPU. Training time for trigger extraction, main argument extraction and sub-argument extraction are about 0.5, 1, 4 hours, respectively. The training time varies according to the number of argument(or trigger) types that need to be asked for each instance.

Experiments – Generative QA For generative QA experiments, we run the experiments with the Huggingface example code for question answering[6]. We fine-tune the SciFive (PMC Base, 2.2M parameters; Phan et al. 2021) model, a T5 model pre-trained on a large-scale Pubmed corpus, on our dataset. The training batch size is 16. The learning rate is 5×10^{-4} for main argument extraction and 5×10^{-5} for sub-argument extraction. We train the model for no more than 20 epochs with early stopping patience as 2 epochs. We use beam search for decoding with the beam size of 3. We use an NVIDIA TITIAN RTX GPU for model training. The training of the generative QA model costs one hour (or less) and about four hours for (trigger and) main arguments extraction and sub-arguments extraction, respectively.

E Experimental Results using Different Question Templates

We present the experimental results on the development set with different question templates below. Table A2 and Table A3 show the results of the extractive QA model when using different templates for trigger extraction and main argument extraction, respectively. Table A4 shows the sub-argument extraction results of the extractive QA and the generative QA method.

Overall, we observe that using different templates does not have a large impact on the results, which is probably due to the fact that our dataset involves few event types and relatively fixed argument types. Specifically, templates that achieve the best results on different metrics vary. The query template *verb* obtains the best trigger and event type detection performance, while a full-sentence question *What is the trigger in the event?* get modestly better trigger identification performance.

[4] https://github.com/Alibaba-NLP/ACE

[5] https://github.com/xinyadu/eeqa

[6] https://github.com/huggingface/transformers/tree/main/examples/pytorch/question-answering

For main argument extraction, the template including a query of the argument and the event trigger achieves the best scores. For sub-argument extraction, the best-performing templates also slightly differ depending on the model. For the extractive QA model, a brief question including information on the event type, the main argument type and extracted span, and the queried sub-argument type gets the best exact match result, while also giving this information but changing the query argument type to a complete sentence would achieve the best token-level score. For the generative QA model, the argument type-specific query with all information about the event type and the main argument performs best on both span-level and token-level evaluation.

F Sampled Error Cases

Table A5 lists some example error cases as complementary material for the discussion in Section 4.3. We present one example for each argument type in the table.

Template	Trigger_CLS_F1	Trigger_IDT_F1	Event_CLS_F1
What is the trigger in the event?	74.14	**75.34**	87.24
What happened in the event?	73.76	74.97	87.64
trigger	73.67	74.87	87.14
t	73.72	74.73	87.59
action	72.98	74.48	87.24
verb	**74.18**	75.00	**87.81**
null	73.61	75.04	86.68

Table A2: Trigger extraction results with different templates for the extractive QA method.

Template	Identification		Classification	
	EM_F1	Token_F1	EM_F1	Token_F1
<argument type >	70.35	83.60	70.27	81.67
<argument type > in <event type>	70.71	82.83	70.60	81.67
<argument type> in <event trigger>	71.80	84.11	71.64	82.68
<argument query>	69.82	82.83	69.58	81.32
<argument query> in <event type>	70.20	83.28	70.08	82.12
<argument query> in <event trigger>	**72.33**	**85.06**	**72.17**	**83.57**

Table A3: Main argument extraction results with different templates for the extractive QA method.

Template	Extractive QA		Generative QA	
	EM_F1	Token_F1	EM_F1	Token_F1
<sub-argument type>	71.21	74.26	71.94	79.15
<sub-argument type> in <event type>	72.53	76.81	73.16	80.89
<sub-argument type> in <main argument span>	76.14	77.56	77.13	84.19
<event type>. <main argument type>, <main argument span>. <sub-argument type>?	**76.92**	78.40	77.00	84.22
<sub-argument query>?	72.21	74.40	73.60	80.87
<sub-argument query> in <event type>?	72.03	75.14	74.26	81.68
<sub-argument query> in <main argument span>?	76.56	78.98	76.70	83.86
<event type>. <main argument type>, <main argument span>. <sub-argument query>?	76.38	**79.43**	**77.23**	**84.57**

Table A4: Sub-argument extraction (classification) results with different templates for extractive and generative QA methods.

Input	Output
Query argument type: Subject **Sentence:** We report **a patient with inoperable pancreatic cancer** who developed gastrointestinal bleeding secondary to radiation-recall related to gemcitabine and review literature.	**Sequence Labelling:** (ADE) a patient with **Extractive QA:** (ADE) a patient with inoperable pancreatic cancer **Generative QA:** (ADE) a patient with inoperable pancreatic cancer
Query argument type: Treatment **Sentence:** Supravenous hyperpigmentation in association with **CHOP chemotherapy** of a CD30 (Ki-1)-positive anaplastic large-cell lymphoma.	**Sequence Labelling:** (ADE) CHOP chemotherapy of a CD30 (**Extractive QA:** (ADE) CHOP chemotherapy **Generative QA:** (ADE) CHOP chemotherapy
Query argument type: Effect **Sentence:** CONCLUSIONS: **Priapism** is an uncommon but potentially serious adverse effect of zuclopenthixol that practitioners, as with many other antipsychotics, should be aware of.	**Sequence Labelling:** (ADE) Priapism **Extractive QA:** (None) None **Generative QA:** (ADE) Priapism
Query argument type: Subject.Age **Sentence:** CONCLUSIONS: Musculoskeletal complaints were the presenting symptoms in four of 44 **children** (9%) treated for relapsed Wilms' tumors with ifosfamide, a derivative of cyclophosphamide.	**Sequence Labelling:** (None) children **Extractive QA:** (None) None **Generative QA:** (ADE) children
Query argument type: Subject.Gender **Sentence:** We report the case of a 74-year-old **female** patient who received the antide-pressant amitriptyline because of a major depression.	**Sequence Labelling:** (ADE) female **Extractive QA:** (None) None **Generative QA:** (ADE) female
Query argument type: Subject.Population **Sentence:** Musculoskeletal complaints were the presenting symptoms in **four of 44** children (9%) treated for relapsed Wilms' tumors with ifosfamide, a derivative of cyclophosphamide.	**Sequence Labelling:** (None) four of 44; (9% **Extractive QA:** (None) None **Generative QA:** (ADE) four of 44
Query argument type: Subject.Race **Sentence:** CASE SUMMARY: A febrile 36-year-old seaman from **Mumbai (Bombay)** was prescribed >5 times the usual dose of chloroquine for malaria diagnosed empirically onboard ship.	**Sequence Labelling:** (PTE) None **Extractive QA:** (PTE) None **Generative QA:** (PTE) None
Query argument type: Subject.Disorder **Sentence:** We describe a 57-year-old man with **acral erythrocyanosis** progressing to acute digital ischemia and gangrene that developed after combined chemotherapy (bleomycin and methotrexate) used to treat a metastatic squamous cell carcinoma of the hypopharynx.	**Sequence Labelling:** (ADE) None **Extractive QA:** (ADE) None **Generative QA:** (ADE) None

Query argument type: Treatment.Drug **Sentence:** We conclude that **MB** is an effective treatment for **ifosfamide**-induced encephalopathy. **Note:** Nested events are included in this case. MB is the Treatment.Drug for the PTE event, ifosfamide is the Treatment.Drug for the ADE event.	**Sequence Labelling:** (ADE) ifosfamide **Extractive QA:** (ADE) ifosfamide **Generative QA:** (PTE) MB
Query argument type: Treatment.Dosage **Sentence:** Severe rhabdomyolysis following **massive** ingestion of oolong tea: caffeine intoxication with coexisting hyponatremia.	**Sequence Labelling:** (ADE) None **Extractive QA:** (ADE) None **Generative QA:** (ADE) None
Query argument type: Treatment.Route **Sentence:** Three hundred and thirty eight patients with moderate to severe painful diabetic neuropathy despite receiving their maximum tolerated dose of gabapentin, had **oral** prolonged-release oxycodone or placebo **tablets** added to their therapy for up to 12 weeks.	**Sequence Labelling:** (None) oral; tablets **Extractive QA:** (None) None **Generative QA:** (PTE) None
Query argument type: Treatment.Frequency **Sentence:** A 36-y-o patient with schizophrenia, who had consumed gradually increasing quantities of oolong tea that eventually reached 15 L **each day**, became delirious and was admitted to a psychiatric hospital.	**Sequence Labelling:** (ADE) each day **Extractive QA:** (ADE) None **Generative QA:** (ADE) None
Query argument type: Treatment.Duration **Sentence:** A 10-year-old boy with osteosarcoma and normal renal function manifested laboratory evidence of impending renal toxicity and extreme elevation of aspartate aminotrasferase and alanine aminotransferase within 2 hours after the completion of a **4-hour** infusion of high-dose methotrexate (MTX) (12 g/m2), and went on to develop acute renal failure with life-threatening hyperkalemia 29 hours later.	**Sequence Labelling:** (ADE) hour **Extractive QA:** (ADE) None **Generative QA:** (ADE) None
Query argument type: Treatment.Time_elapsed **Sentence:** She was placed on adjuvant Adriamycin (doxorubicin) chemotherapy, but **6 months** later died of Adriamycin toxicity.	**Sequence Labelling:** (None) 6 month later **Extractive QA:** (None) None **Generative QA:** (ADE) None
Query argument type: Treatment.Disorder **Sentence:** Pulmonary edema during acute infusion of epoprostenol in a patient with **pulmonary hypertension and limited scleroderma**.	**Sequence Labelling:** (ADE) pulmonary hypertension and limited scleroderma **Extractive QA:** (ADE) epoprostenol **Generative QA:** (ADE) pulmonary hypertension; limited scleroderma
Query argument type: Combination.Drug **Sentence:** Thus, tardive seizures in our cases are thought to be related to **piperacillin** and **cefotiam**.	**Sequence Labelling:** (ADE) piperacillin; cefotiam **Extractive QA:** (ADE) piperacillin **Generative QA:** (ADE) piperacillin; cefotiam

Table A5: Example error cases. Standard gold annotations are bolded in the original sentence. Model detected event types are shown in (·) before the predicted argument span. Error predictions are shown in red.