Biomedical Event Extraction as Multi-turn Question Answering

Xing David Wang[1], **Leon Weber**[1,2], **Ulf Leser**[1]
[1]Computer Science Department, Humboldt-Universität zu Berlin
[2]Max Delbrück Center for Molecular Medicine
{wangxida, weberple, leser}@informatik.hu-berlin.de

Abstract

Biomedical event extraction from natural text is a challenging task as it searches for complex and often nested structures describing specific relationships between multiple molecular entities, such as genes, proteins, or cellular components. It usually is implemented by a complex pipeline of individual tools to solve the different relation extraction subtasks. We present an alternative approach where the detection of relationships between entities is described uniformly as questions, which are iteratively answered by a question answering (QA) system based on the domain-specific language model SciBERT. This model outperforms two strong baselines in two biomedical event extraction corpora in a Knowledge Base Population setting, and also achieves competitive performance in BioNLP challenge evaluation settings.

1 Introduction

Biomedical event extraction (BEE) (Björne and Salakoski, 2011) aims to extract molecular events from natural text, where an event typically encompasses certain biomedical entities, such as genes, proteins, complexes or cellular components, specific trigger words determining the event type, and relationships between the entities whose roles depends on the event type. For instance, the verb *phosphorylates* is a hint to a mention of a phosphorylation event in a given sentence and typically has two entities, one that controls the phosphorylation and one that is phosphorylated. Events may also involve other events, such as the inhibition of an expression, and may ultimately form partial or entire biological pathways (Gonzalez et al., 2015).

State-of-the-art methods for BEE rely on learning textual patterns and features from annotated documents where entities and their specific role in an event structure are manually marked. They typically consist of multiple classifiers to solve the different subtasks of trigger, role, and event detection, each requiring individual training and validation data. In this paper, we instead model BEE as iterative question answering, using the same model for each of the individual steps which allows knowledge sharing and joint learning of the different event components. We show that this model is as effective in predicting event structures in two BioNLP shared tasks (GENIA, 2011 and Pathway Curation, 2013) as a baseline consisting of multiple, CNN based classifiers (Björne and Salakoski, 2018).

The paper is structured as followed: In Section 2, we give a brief overview over related work in biomedical event extraction and in question answering. We define the event extraction task, our question answering model, and our evaluation setup in Section 3. In Section 4, we present our results and discuss them before we conclude the paper. The code and pretrained models are freely available at https://github.com/WangXII/bio_event_qa.

2 Related Work

Approaches to BEE can be divided into two categories: Approaches using manually defined rules (Valenzuela-Escárcega et al., 2015) and approaches making use of machine learning algorithms. Early approaches of the latter category, such as EventMine (Miwa and Ananiadou, 2013) or the Turku Event Extraction System (TEES) (Björne and Salakoski, 2011), had in common that they achieve event extraction through a pipeline of several independent classifiers, each solving a different subtask of event extraction and each based on a set of specifically defined features extracted from the text, often after heavy and error-prone preprocessing (e.g., POS tagging, dependency parsing). More recent

Proceedings of the 11th International Workshop on Health Text Mining and Information Analysis, pages 88–96

https://doi.org/10.18653/v1/P17

works use neural architectures, where the previously manually defined features are replaced by automatically learned text representations (Björne and Salakoski, 2018; Trieu et al., 2020), involving techniques like word embeddings and other language models. While the original TEES (TEES SVM) (Björne and Salakoski, 2011), was based on a pipeline of SVMs using manually defined features, the more recent TEES CNN (Björne and Salakoski, 2018) additionally incorporates biomedical word embeddings as features and replaces the SVMs with CNNs. As pipelined models suffer from error propagation (for instance, an undetected event trigger in the first phase leads to missing the event entirely), approaches based on joint inference recently became more popular. Zhu and Zheng (2020) assign a separate probability to each event trigger, relation and event candidate and move the final decision about the veracity of an event structure to an optimization scheme solved in a post-processing step. DeepEventMine (Trieu et al., 2020) is a derivative of EventMine (Miwa and Ananiadou, 2013) and makes use of text representations learned by BERT (Devlin et al., 2018). It tries to avoid error propagation by training a multi-layer network for BEE in an end-to-end manner and achieves new state-of-the-art performance in various biomedical event extraction corpora. In contrast to these previous approaches, our model employs a network with only one single output layer for all event extraction subtasks and it does not need to introduce a new layer for each subtask.

In this work, we will model BEE as a an iterative question answering (QA) process. This idea was brought up first by McCann et al. (2018), who showed how to model ten different NLP tasks, among them machine translation, summarization, and sentiment analysis, as question answering tasks over a properly defined context. Li et al. (2019) proposed a specific question answering framework for event extraction based on the idea of extracting the entities of individual relations using so-called "question turns". In each turn, the question answering procedure asks a question for a new entity from the relation followed by a text passage where a span is marked as the output entity. Found entities from previous turns are included in the questions of subsequent turns to allow for more precise subsequent queries. The process is controlled by predefined question templates which determine the sequence of turns depending on the event type. However, this work is not applicable to BEE, because it assumes a fixed number of arguments and has no support for nested events (events that serve as arguments for other events), which are two defining characteristics of the BEE task.

In this paper, we develop a similar framework for the extraction of nested biomedical events. Our framework applies SciBert (Beltagy et al., 2019), a domain-specific refinement of BERT (Devlin et al., 2018), as underlying QA method. BERT (and SciBert) is a pre-trained transformer model (Vaswani et al., 2017) which relies on an attention mechanism to learn relationships between different parts of a sequential input, which was shown to better capture long-term dependencies than Convolutional and Recurrent Neural Networks. The parameters of its final layers can be used as input features to other models, or can be used in a fine-tuning procedure involving a further, task-specific layer for problems like question answering, sentence similarity quantification, or sentence continuation prediction.

3 Material and Methods

3.1 Event Extraction

Biomedical event structures are used to model biomedical processes. In general, they consist of signal words, called trigger of the event, and biomedical entities, called arguments of the event. The trigger determines the event type, which in turn determines the semantics (or roles) of its arguments. Event triggers are often verbs or nouns such as *phosphorylation*, *transcription* or *binds*, whereas biomedical entities typically are proper nouns, such as *NF-kappa B*, *ATP* or *glucose*. The role *theme* denotes the central object of interest in an event, while the role *cause* often is the facilitator or driver of the event. Notably, events can be arguments of other events, for instance when a protein *A* (the cause) *activates* (the trigger) the phosphorylation (the theme, in this case a nested trigger) of another protein *B* (the argument of the nested event). A typical biomedical event structure is illustrated in Figure 1.

3.2 Multi-turn Question Answering

In order to find simple and complex event structures, we adopt the multi-turn question answering approach of Li et al. (2019) to BEE. We cast it as a series of QA tasks, where each individual QA problem is modeled as a sequence labeling task in

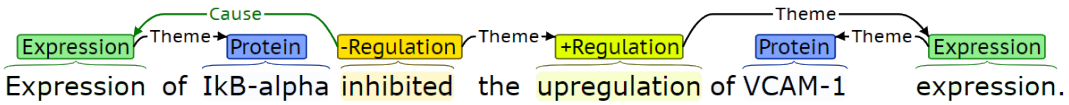

Figure 1: Event visualization using BRAT by Stenetorp et al. (2011)

Table 1: Our question template and the expected answers when applied to the example from Figure 1. In the first question we ask for simple events involving our the chosen entity as a theme. If the entity is part of an event we retrieve the corresponding event trigger, its type and position in the text. Then, we ask for other event arguments belonging to the trigger-theme pair. Subsequent questions aim to uncover recursive events containing the just extracted simple event as a theme. The recursive descent ends as soon as the event is not found to be part of another structure.

Questions:	*Answers:*
1. What are events of *VCAM-1*?	The Expression *expression* at (62,72).
2. What are arguments of the *Expression* of *VCAM-1*?	None.
3. What are events of the *Expression* of *VCAM-1*?	The Positive regulation *upregulation* at (39,51).
4. What are arguments of the *Positive regulation* of the *Expression VCAM-1*?	None.
5. What are events of the *Positive regulation* of the *Expression* of *VCAM-1*?	The Negative regulation *inhibited* at (25,34).
6. What are arguments of the *Negative regulation* of the *Positive regulation* of the *Expression* of *VCAM-1*?	The Cause *expression* at (1,11).
7. What are events of the *Negative regulation* of the *Positive regulation* of the *Expression* of *VCAM-1*?	None.

which the model decides for each token whether it belongs to an answer of the current question and if it does, which role it has. This can be interpreted as a kind of multitask learning in which the different tasks are not defined by different loss functions but through different types of questions. Triggers determine the specific event type whereas entities take one of the event argument roles. The formulation as sequence labeling tasks allows for multiple text spans to be tagged as answers of the same question which is beneficial as (1) an entity can participate in two distinct event structures and (2) an event can have multiple different arguments. The model assumes gold standard annotation of all entities in the corpus and uses these to structure the iterative QA process, treating each gold-standard entity as a potential theme argument. It expands events from there by iteratively asking for corresponding event triggers, event arguments and nested regulation events.

We introduce the notion of a question template which defines the different types of questions we use in our model and the sequence of turns we pose them. Our question template follows a recursive procedure and distinguishes two main question types, one for detecting event triggers and one for detecting event arguments. The process iterates through all given entities and asks whether there are any events with this entity as theme. This first question belongs to the **Triggers** question type and detects triggers corresponding to a theme candidate. In the subsequent **Arguments** question we ask for arguments belonging to a previously discovered (theme, trigger) combination. Applying the first question type **Triggers** to our example from Figure 1, we ask for all event triggers and their event type belonging to the protein *VCAM-1*. Note that this question addresses all different mentions of the entity *VCAM-1* in the given document. In our example, the assignment of answer triggers to entity evidences is clear as *VCAM-1* is mentioned exactly once in the document; in cases where an argument is mentioned more than once, we need to perform the correct assignment in a subsequent step (see next section). As the answer to our question we mark the event trigger *expression* with the event type *Expression*. In every **Arguments** question (cf. Table 1) we incorporate the event trigger found from the previous answer into the formulation of the new question. Next, we query for non-theme arguments belonging to the *Expression* of *VCAM-1* which yields no answers in this example.

The subsequent questions deal with finding nested structures and rely on the same schema of alternating **Triggers** and **Arguments** question turns. We ask which other events our previously found event could be a theme of, i.e., we ask "*Which are the events of the Expression of VCAM-1?*". In our

example, we find that the *VCAM-1 expression* is *upregulated*; the trigger *upregulated* denotes an event of type *positive regulation*. If we found multiple answers of different event types to the same entity or event in a **Triggers** question, we expand each single of these into a separate event structure (see next section). In our example, we find exactly one answer to the nested trigger question and proceed again by querying for the arguments of the found event. The recursion can go on for an arbitrary amount of steps as it only stops when there is no new event trigger for a **Triggers** question[1]. In the example, we recurse twice and then stop with the result that the *upregulation* of the *VCAM-1 expression* is itself *inhibited* in a *negative regulation* event which is caused by an *expression* event.

An overview of the application of our method to the example from Figure 1 can be found in Table 1. Pseudocode of our framework is given in Algorithm 1.

We transform all the event annotations provided by the tasks to natural language questions. The mapping from event annotation to question is straightforward and not described further here.

3.3 Event Merging

The answers from our question answering model results in only basic and partly underdetermined event structures that do not fit the format of events in our evaluation corpora. We apply two different post-processing steps: Event matching, where we identify the text span best matching the prior event structure from the question and the entity/trigger from the answer, and event merging, where we merge the prior event structure and the entity/trigger from the answer into one single event structure.

We illustrate both procedures using the example from Figure 1. In the first **Triggers** questions, we receive the expression trigger at character positions (62,72) as an answer for *VCAM-1*. In the matching step, we need to identify which *VCAM-1* entity in the text the expression trigger at position (62,72) belongs to. We look up an entity and trigger dictionary, which stores positions of all entities and of all detected triggers. We then compute the differences of starting positions for each mention of the entity and the starting position of the trigger and choose the occurrence with the smallest difference. In our example, the *VCAM-1* entity at position (55,61) is identified as a match for the trigger *expression* at position (62,72) with a difference of 7 characters. In the merging step, we combine the trigger *expression* at position (62,72) and the entity *VCAM-1* at position (55,61) to a single new event structure.

The specific algorithm for event merging depends on the question type and the possible answers. We explain the differences using two examples. Assume we found a phosphorylation event with theme *A* in the first **Triggers** question. Asking for arguments belonging to this prior event, assume we receive four answers, namely cause *B*, cause *C*, site *D* and site *E*. In this case of multiple argument types, we enumerate all possible cause site combinations, merge them with the prior event and receive four new phosphorylation events, i.e., phosphorylation of theme *A* with cause *B* and site *D*, phosphorylation of theme *A* with cause *B* and site *E* etc. Details regarding the performance for this merging heuristic is found in Table 4, query five. A more sophisticated merging approach is needed for binding and pathway events which may contain multiple participants. For these events, we store a directed graph per event trigger where nodes are participants and a directed edge exists from entity *A* to entity *B* if *B* is answer to the **Arguments** question of *A*. After the graph is constructed, we transform it into an undirected graph where we keep all edges which exist in both directions. In the final step, we detect maximal cliques in the graph and form a distinct binding/pathway event for each clique. The results for binding/pathway merging is found in Table 4, query six. We use similar heuristics for the merging step of nested regulations and other event types.

3.4 Implementation

We use Huggingface's Transformers[2] (Wolf et al., 2019) library in Pytorch for our implementation. For the initialization of the pretrained BERT neural network model we use SciBERT[3] (Beltagy et al., 2019) which has been pretrained on scientific literature. We add one softmax layer as output on top of the final hidden representation of each token as we fine-tune the model parameters for our question answering task. In the final output layer each token in a given document sequence is tagged in IOB2-style as either being inside, outside or the beginning of

[1] The deepest nesting occurring in our two evaluation corpora is three.

[2] https://github.com/huggingface/transformers

[3] https://github.com/allenai/scibert

Algorithm 1 Pseudocode of our QA framework for the extraction of event structures. We expand event structure candidates around potential theme arguments, adding corresponding event triggers in the question **Triggers** and corresponding event arguments in the question **Arguments**. If we have found new events we add their (theme, trigger)-pair to our event candidates list for the next iteration, where we ask whether the just found event is a theme to a (new) nested event.

```
event_candidates = proteinsFromDocument()
while event_candidates ≠ ∅ do
    new_events = ∅
    for candidate in event_candidates do
        new_triggers = Triggers(candidate)
        new_arguments = Arguments(candidate)
        new_events.add(Event(candidate, new_triggers))
    end for
    event_candidates = new_events
end while
```

an answer token. The beginning and inside tags are further divided into the different event type and event argument classes according to the structures seeked in a corpus. The same BERT neural network model is shared across the whole task and all questions. This allows knowledge sharing and joint learning of the different questions.

For training, we create all existing questions in the training set exactly once in the beginning and then draw randomized batches as our training examples. We use the default AdamW configuration with learning rate 5e-5, no weight decay, β_1 = 0.9, β_2 = 0.999 and ϵ = 1e-8. Training is conducted on four Nvidia GeForce RTX 2080 Ti GPUs. Our maximum sequence length for the input data is 384 tokens. To deal with longer sequences than the maximum sequence length, we duplicate the beginning and the end of intermediate sequences so that they form overlapping windows with a length of 64 tokens. To decide between two differently predicted tags for the same token in two adjacent windows, we choose the tag of the token which has the larger context window. We enable apex[4] fp16 16-bit mixed precision for improved computation efficiency.

Hyperparameters to choose are the batch size and the number of epochs when to stop training. During our model development, a batch size of 16 has proven to work well together with 16 epochs after which the validation loss usually does not improve anymore. The whole training process during fine-tuning is relatively fast and the training time ranges from half an hour to an hour on Pathway Curation to around two hours on GENIA depending on hyperparameter choice. Performance in evaluation fluctuates over few percentage in F1-score depending on the initial seed during neural network initialization. As we mainly compare to Björne and Salakoski (2018), we adopt their evaluation strategy and report the results of the seed with the best performance on the validation set.

[4] https://github.com/NVIDIA/apex

3.5 Corpora

We evaluate our approach to BEE on two corpora used widely in biomedical NLP research, namely the Pathway Curation corpus (PC) from the BioNLP13 challenge (Ohta et al., 2013) and the Genia 11 corpus (GENIA) from the BioNLP11 challenge (Kim et al., 2011). These corpora consist of annotated PubMed abstracts and full texts. The PC dataset focuses on pathway reactions whereas GENIA aims to cover molecular biology in general. GENIA contains 14,958 sentences and PC 5,040 sentences. GENIA distinguishes seven different event types and six different argument types, whereas PC distinguishes 24 different event types and nine argument types. Both corpora include common biochemical event types, such as *phosphorylation*, *gene expression*, *binding* or *positive (negative) regulation*. PC further distinguishes multiple conversion types, such as *dephosphorylation*, *acetylation*, *ubiquitination* etc. and it adds *activation* and *inactivation* to the class of regulation events. PC also annotates event modifiers, i.e., *speculation* and *negation*, and allows for events without a theme. The latter two types of event components currently are not addressed by our work, but could be included by adding further turns and questions to our question template. A closer breakdown of the events and their components in the two corpora can be found in Table 2.

Table 2: Statistics of our question answering training datasets built from the gold event annotations.

	GENIA11		Pathway Curation	
Question type	#questions	#gold answers	#questions	#gold answers
Simple Events				
Triggers	6,392	6,549	4,316	3,857
Arguments	6,263	1,486	3,242	2,389
Nested Events				
Triggers	10,564	3,523	5,012	1,708
Arguments	4,303	1,096	1,775	1,440
Total	27,522	12,654	14,345	9,394

3.6 Evaluation Tasks

We evaluate our model for two different tasks: Knowledge Base Population (KBP) and the standard BioNLP a* setting. In both cases, gold-standard entity annotations are provided with the corpus whereas event annotations have to be predicted.

Knowledge Base Population

Following Kim et al. (2015), we evaluate the models' capability to answer a set of predefined queries, such as finding all pairs of proteins that bind to each other. An overview of the different knowledge base queries is found in Table 3. The first four queries can be directly answered from our question answering model while the remaining three require event merging, which we perform as described in Section 3.3. As usual in KBP settings, the extracted event structures are compared on a document-level, so a same event occurring twice in a single document is counted once only in this format.

BioNLP .a* evaluation

The .a* evaluation format is the standard evaluation format provided by the GENIA and PC shared tasks. PC is conducted in a strict matching evaluation mode, where the extracted triggers, all event arguments, and their text spans must exactly coincide. The approximate span and approximate recursive matching mode for GENIA is more lenient as the text spans and positions may differ up to one word from the gold-standard annotations and nested regulation events only need to coincide in their theme arguments.

4 Results and Discussion

4.1 Knowledge Base Population

We use TEES SVM (Björne and Salakoski, 2011) and TEES CNN (Björne and Salakoski, 2018) [5] as baselines for knowledge base population. Both provide result files and models online[6]. We compare the result of our single homogeneous QA multi-turn model to the individual models of these approaches.

The results can be found in Table 4. Our approach achieves a 0.87 percentage points (pp) and a 2.47 pp better F1-score than TEES CNN and TEES SVM, respectively, on GENIA. On PC, it achieves a 2.40 pp and a 3.13 pp better F1-score. This increase can be attributed to a considerably better recall (2.35 pp for GENIA and 6.59 pp for PC, compared to TEES CNN). Its precision is 1.38 pp and 2.24 pp lower than the respective best baseline result. It shows performance gains of up to 5.16 pp F1 in the first three *Basic Event* queries which require no event merging. Results for the other type of queries are mixed: Our model achieves good results for binding and pathway pairs, yet is worse for transitive protein regulations and the combination of all conversion arguments.

Most likely, the question answering approach achieves strong performances in extraction of simple events as they rely on only one or two questions and require no complicated merging steps. The model infers binding and pathway pairs in the fifth query relatively well since we explicitly query for those in the **Arguments** question type. The worse results for the arguments of a conversion event in the sixth query are probably due to the naive heuristic of simply enumerating all valid argument combinations as output during event merging. Regulation event detection in the forth and seventh query presumably also suffer from our too-simple event merging as we match a detected event trigger cause to a whole previously discovered event structure. We also observe that error propagation negatively influences regulation detection and event detection as we immediately extract simple events after our first **Triggers** question from the (theme, trigger)-pairs, but we do not incorporate event arguments or regulations found in later question turns into a joint extraction of events.

[5] https://github.com/jbjorne/TEES

[6] https://b2share.eudat.eu/records/bee50aa63b0b404da9c76b29de4d8653

Table 3: Queries for our Knowledge Base Population evaluation, adapted from Kim et al. (2015). We conduct evaluation of found events at document level, i.e., counting unique event structures per document. The answers are denoted as tuples. Example questions and answers are given in italics.

Knowledge Base Queries on a document	
Query description	Example answer
1. Which protein appears in context of event *A*?	(EventType, ProteinTheme)
- *Which protein appears in context of a gene expression?*	- *(Gene Expression, MACS1)*
2. What is an argument of event *A* of entity *X*?	(EventType, ProteinTheme, ArgumentType, Argument)
- *What is the location of the localization of MACS1?*	- *(Localization, MACS1, ToLoc, mitochondrial matrix)*
3. Is the simple event *A* part of a regulation?	(SimpleEvent, Boolean)
- *Is the transport of hydroxyl part of a regulation?*	- *((Transport, hydroxyl), yes)*
4. What regulates the simple event *A*?	(SimpleEvent, Cause)
- *What regulates the transport of hydroxyl?*	- *((Transport, hydroxyl), amiloride)*
5. What is the site of the conversion event of *A* with cause *B*?	(EventType, ProteinTheme, ProteinCause, ProteinSite)
- *What is the site for the acetylation of H3 by Asf1?*	- *(Acetylation, H3, Asf1, K56)*
6. What binds to protein *A*?	(Protein1, Protein2)
- *What binds to Na+?*	- *(Na+, H+)*
7. What regulates *A* transitively?	(ProteinTheme, ProteinCause)
- *What regulates NF-kappaB?*	- *(NF-kappaB, TLR2)*

Table 4: Results for Knowledge Base Population on the development sets, compared to TEES SVM and TEES CNN. Semantics for each individual question are found in Table 3. The answers of the first four queries (Simple Events) can be derived by our model without event merging. The two lower sections show only F1 scores. The best value in each partial column is marked in bold.

	GENIA				Pathway Curation			
Metric/Question type	TEES SVM	TEES CNN	QA with BERT	Support	TEES SVM	TEES CNN	QA with BERT	Support
F1 (Total)	59.78	61.38	**62.25**	3625	56.06	56.79	**59.19**	3141
Precision (Total)	68.80	**69.68**	68.30	3625	**60.57**	60.52	58.33	3141
Recall (Total)	52.86	54.84	**57.19**	3625	52.18	53.49	**60.08**	3141
1. Theme Trigger Pairs	73.07	75.23	**79.41**	1301	69.21	69.34	**74.50**	866
2. Event Arguments	**49.17**	46.76	47.36	568	45.84	46.94	**49.31**	648
3. Nested Regulation Events	63.61	66.40	**71.08**	585	66.14	64.43	**71.05**	339
4. Nested Regulation Causes	39.71	**44.21**	36.03	384	**46.19**	43.78	44.44	419
Basic Events (Total)	63.24	64.38	**66.53**	2838	58.21	57.84	**61.27**	2272
5. Full Conversion Events	-	-	-	0	38.89	**61.11**	56.25	16
6. Binding/Pathway Pairs	55.14	46.03	**60.18**	126	56.84	53.64	**60.59**	138
7. Transitive Regulations	42.14	**50.05**	38.64	660	48.72	**53.79**	51.14	715
Merged Events (Total)	44.58	**49.38**	42.80	787	50.00	**53.93**	53.20	869

Table 5: Results on the standard .a* evaluation of BioNLP shared tasks, comparing our model with four competitors. The test set evaluation is conducted online where predictions are submitted to a server and the final results are returned. DeepEventMine (Trieu et al., 2020) represents results of very recent work. Note that our model does not account for event modifications or events without themes in the PC corpus. Dev (adjusted) denotes the results on the PC development set excluding these annotations. The best value in each partial column is marked in bold.

	GENIA11				Pathway Curation				
Task/Data set	Test Set			*Dev Set*	Test Set			*Dev Set*	*Dev (adjusted)*
	F1	Precision	Recall	*F1*	F1	Precision	Recall	*F1*	*F1*
TEES SVM (Björne and Salakoski, 2011)	53.30	57.65	49.56	*56.00*	51.10	55.78	47.15	*44.34*	*45.55*
EventMine (Miwa and Ananiadou, 2013)	57.98	63.48	53.35	-	52.84	53.48	**52.23**	-	-
TEES CNN (Björne and Salakoski, 2018)	56.80	64.86	50.53	*58.57*	52.10	58.31	47.08	*46.07*	*47.10*
DeepEventMine (Trieu et al., 2020)	**63.02**	**71.71**	56.20	***62.75***	**55.67**	**64.12**	49.19	***56.57***	-
QA with BERT	58.33	59.33	**57.37**	*56.50*	48.29	48.74	47.85	*44.60*	***47.59***

4.2 BioNLP .a* Evaluation

In Table 5, evaluation results in the BioNLP .a* challenge setting are compared to four competitors: TES CNN (Björne and Salakoski, 2018), TEES SVM (Björne and Salakoski, 2011), EventMine (Miwa and Ananiadou, 2013), and the very recent DeepEventMine (Trieu et al., 2020). On GENIA11, our proposed approach beats three competitors on the test set, but is outperformed by DeepEventMine by almost 5 pp in F1-score. The higher recall and lower precision compared to DeepEventMine might be attributed to the simple rule-based event merging step, which constructs events for all detected relations regardless of their score. In contrast, DeepEventMine models the event construction as a separate machine learning task in which errors from earlier steps can be corrected, potentially leading to a higher precision.

For the PC corpus, our results are considerably worse than those of the baselines on both the dev and the test set. This inferior performance can be attributed to the fact that the proposed model does not account for event modifications or events without themes. Accordingly, we evaluated the models again on the development set excluding such annotations. The results for this experiment can be found in the column *Pathway Curation Dev (adjusted)*. Under this setting our proposed model outperforms both TEES variants. Note that events without themes and their regulations make up to a tenth of the events in the development set of PC, among them the majority are simple pathway events only made up by an event trigger.

4.3 Error Analysis

Table 6: Error statistics of our question answering model.

	GENIA11		Pathway Curation	
Error type	# wrong answer	%	#questions	%
Wrong Trigger Spans	643	46.5	912	67.8
Wrong Trigger Label	56	4.1	60	4.4
Wrong Argument Spans	674	48.9	370	27.5
Wrong Argument Label	7	0.5	3	0.2
False Positives (Total)	1,380	100	1,345	100
Missing Trigger Spans (Question)	335	31.8	642	40.1
Missing Trigger Spans (Propagated)	122	11.6	243	15.2
Wrong Trigger Label	56	5.3	60	3.7
Missing Argument Spans (Question)	177	16.9	194	12.1
Missing Argument Spans (Propagated)	356	33.4	458	28.7
Wrong Argument Label	7	0.7	3	0.2
False Negatives (Total)	1,053	100	1,600	100

We conducted an error analysis on the dev sets of the GENIA11 and PC corpora. Results are shown in Table 6. We distinguish error types into false positives and false negatives:

- *Wrong Trigger/Argument Spans* denotes answers predicted by the model which are no gold-standard answers.
- *Wrong Trigger/Argument Label* means correctly detected text spans which have the wrong event type or wrong argument type.
- *Missing Trigger/Argument Spans (Question)* refers to questions where a trigger or an argument has not been extracted.
- *Missing Trigger/Argument Spans (Propagated)* refers to triggers or arguments which have not been extracted because the according question has not been found (i.e., the answers from a previous question have been wrong so that the subsequent question is not posed).

We find that wrong label assignment is the cause for about five percent of false positives and false negatives. Missing propagated questions make up about one half of the false negatives during question answering in non-regulation event types. The relative amount of errors is lower in GENIA11 compared to Pathway Curation which reflects the overall better model performances in GENIA11.

In an ablation study, we examine the impact of joint training on all questions versus training only on the one question type of simple events trigger detection, i.e., only using the examples of the first **Triggers** question and examining the impact of multi-task learning in our model. We find that training the model only on the one question type results in a worse performance (1.08 pp F1-score) for answering this one specific question compared to evaluating the found triggers trained on the full questions dataset. This indicates that the shared model parameters provide a benefit for detecting the right answer to all question types.

5 Conclusion

We presented an approach for BEE in which this task is modeled as multi-turn question answering problem using BERT as underlying language model. We show that our model is able to form event structures from the answers of multiple questions. Our experiments show promising results on two corpora, especially in a Knowledge Base Population setting. In future work, we aim to improve model performance by adjusting the event merging procedure and by using further or modified question templates. It would also be worthwhile

to study the reasons of the performance gains of our model compared to TEES in more detail, for instance by replacing the CNN in TEES CNN with BERT.

References

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: Pretrained language model for scientific text. In *EMNLP*.

Jari Björne and Tapio Salakoski. 2011. Generalizing biomedical event extraction. In *Proceedings of BioNLP Shared Task 2011 Workshop*, pages 183–191.

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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Graciela H Gonzalez, Tasnia Tahsin, Britton C Goodale, Anna C Greene, and Casey S Greene. 2015. Recent advances and emerging applications in text and data mining for biomedical discovery. *Briefings in bioinformatics*, 17(1):33–42.

Jin-Dong Kim, Jung-jae Kim, Xu Han, and Dietrich Rebholz-Schuhmann. 2015. Extending the evaluation of genia event task toward knowledge base construction and comparison to gene regulation ontology task. *BMC bioinformatics*, 16(S10):S3.

Jin-Dong Kim, Yue Wang, Toshihisa Takagi, and Akinori Yonezawa. 2011. Overview of genia event task in bionlp shared task 2011. In *Proceedings of the BioNLP Shared Task 2011 Workshop*, pages 7–15. Association for Computational Linguistics.

Xiaoya Li, Fan Yin, Zijun Sun, Xiayu Li, Arianna Yuan, Duo Chai, Mingxin Zhou, and Jiwei Li. 2019. Entity-relation extraction as multi-turn question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1340–1350.

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2018. The natural language decathlon: Multitask learning as question answering. *arXiv preprint arXiv:1806.08730*.

Makoto Miwa and Sophia Ananiadou. 2013. Nactem eventmine for bionlp 2013 cg and pc tasks. In *Proceedings of the BioNLP Shared Task 2013 Workshop*, pages 94–98.

Tomoko Ohta, Sampo Pyysalo, Rafal Rak, Andrew Rowley, Hong-Woo Chun, Sung-Jae Jung, Sung-Pil Choi, Sophia Ananiadou, and Jun'ichi Tsujii. 2013. Overview of the pathway curation (pc) task of bionlp shared task 2013. In *Proceedings of the BioNLP Shared Task 2013 Workshop*, pages 67–75.

Pontus Stenetorp, Goran Topić, Sampo Pyysalo, Tomoko Ohta, Jin-Dong Kim, and Jun'ichi Tsujii. 2011. Bionlp shared task 2011: Supporting resources. In *Proceedings of BioNLP Shared Task 2011 Workshop*, pages 112–120, Portland, Oregon, USA. Association for Computational Linguistics.

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*.

Marco A Valenzuela-Escárcega, Gus Hahn-Powell, Mihai Surdeanu, and Thomas Hicks. 2015. A domain-independent rule-based framework for event extraction. In *Proceedings of ACL-IJCNLP 2015 System Demonstrations*, pages 127–132.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Lvxing Zhu and Haoran Zheng. 2020. Biomedical event extraction with a novel combination strategy based on hybrid deep neural networks. *BMC bioinformatics*, 21(1):47.