Biomedical Event Extraction with Generative Language Models

Fabio Barth[1], Leon Weber-Genzel[2] and Ulf Leser[1]

[1]*Humboldt-Universität zu Berlin, Unter den Linden 6, 10099 Berlin, Germany*
[2]*Ludwig-Maximilians-Universität München, Geschwister-Scholl-Platz 1, 80539 München, Germany*

Abstract

The recent success of sequence-to-sequence language models like T5 [1] brought noticeable improvements for many general domain natural language processing (NLP) tasks. However, so far these models found little application for complex domain-specific problems such as biomedical event extraction (BEE). BEE is a challenging task with the goal of extracting complex structures that describe relationships between multiple molecular entities from scientific texts. The structure of biomedical events is similar to a graph which makes it non-trivial to decode structured predictions from a single sequence.

Paolini et al. [2] presented a framework called 'Translation between Augmented Natural Languages' (TANL) to solve such structured prediction tasks using sequence-to-sequence language models. It generates sentences for solving tasks like relation extraction, named entity recognition, and event extraction. In this paper, we investigate the effectiveness of the TANL framework for BEE. We designed a natural language description to solve BEE in-sequence using TANL and evaluated it, based on the T5 language model, on two data sets. Our results show that a generative model can perform the BEE task. However, our approach does not outperform the baseline on any tested data set. We find that our model struggles especially with the argument detection step of BEE.

Keywords

biomedical event extraction, information extraction, generative language models

1. Introduction

The database PubMed contains more than 34 million citations and abstracts of biomedical literature [3]. The indexed scientific papers contain much information regarding medical and biomedical research, for instance about protein interactions and the usage of drugs for specific diseases [4]. The main goal in natural language processing of biomedical text is to extract the most useful information from such corpora and provide it for others in a useful format [5]. This, for example, can be done by structuring the information as a graph which is called biomedical event extraction (BEE).

Paolini et al. [2] presented a framework called 'Translation between Augmented Natural Languages' (TANL) to solve structured extraction tasks in a sequence-to-sequence setting. Therefore, a generative language model generates an output sequence from a mark-up enriched

LWDA'23: Lernen, Wissen, Daten, Analysen. October 09–11, 2023, Marburg, Germany

barthfab@informatik.hu-berlin.de (F. Barth); leonweber@cis.lmu.de (L. Weber-Genzel); leser@informatik.hu-berlin.de (U. Leser)
https://leonweber.me/ (L. Weber-Genzel); https://www2.informatik.hu-berlin.de/~leser/ (U. Leser)
0000-0002-2499-472X (L. Weber-Genzel); 0000-0003-2166-9582 (U. Leser)

version of the input that encodes the predicted structured information. The information can be extracted from the generated output and structured as a graph. By transforming tasks like relation extraction, named entity recognition, and even event extraction into translation tasks and then fine-tuning a large generative language model on them, Paolini et al. [2] document state-of-the-art results of all mentioned tasks but using only general English texts. The benefit of using this framework for biomedical event extraction (besides the promising results in various other structured prediction tasks) is that it can be easily adapted for multitask or few-shot learning because of the simple architecture.

In this paper, we used the TANL framework for BEE and analyzed the performance. We extended the natural language description proposed by Paolini et al. [2] to perform BEE. We experiment on two corpora and compare our method to three state-of-the-art baselines. Our results indicate an unfavorable outcome because BEE is more difficult than previous tasks solved by TANL. The rest of the paper is structured as follows: We first discuss the preliminary background. Second, we explain our approach and the experimental setup. We then present and analyze our results and finally give a conclusion.

2. Background

2.1. The Biomedical Event Extraction (BEE) Task

In event extraction, the goal is to extract textual events from a given text that usually already comes with annotated entity mentions. A textual event is a complex combination of relations linked between a set of empirical observations from texts. For a correct event extraction, the model needs to identify an event trigger, its type, and the arguments with the corresponding type [6]. The task can also be split into two sub-tasks: event detection and argument detection. Event detection describes the task of identifying the event trigger and its event type and argument detection describes the task of identifying the arguments and their types given an event. It is important to note, that an argument of an event e_i could also be another event e_j. That is why an event representation has more of a graph-like structure and is more complex than a simple relation between two given entities [6].

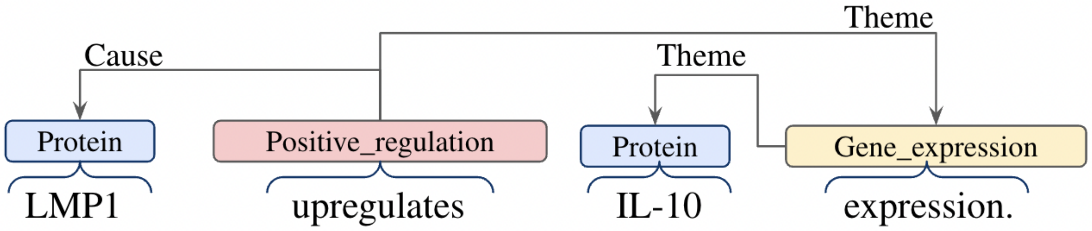


Figure 1: Sentence containing two events. The first event is a gene expression event (yellow) with the protein entity (blue) ”IL-10” as theme argument. The second event is a positive regulation event with the protein entity (blue) ”LMP1” as cause argument and the gene expression event as theme argument.

Consider the sentence from Figure 1 containing two different events. In this example, there are two biomedical events: positive regulation event and gene expression event. The gene expression event has as event trigger the word ”expression” and contains a single argument. This argument is a protein entity and its argument type is considered a ”Theme”. However, the positive regulation event with the event trigger ”upregulates” has two arguments, the entity ”LMP1” and the gene expression event with the event trigger ”expression”. These arguments have two different argument types, ”Cause” and ”Theme”, respectively.

There are three different equivalence criteria for correct event extraction: Strict matching, approximate span matching, and approximate recursive matching [7]. Strict matching has the strongest equivalence criteria where no mismatches with the gold events are allowed for a correct prediction. In approximate span matching, only small deviations from the gold standard are allowed [7]. For instance, in approximate span matching a predicted event would still pass the equivalence criteria if a gold event trigger is entirely contained in the predicted event trigger but the prediction is extended by one or two words before or after the gold trigger. This would not hold for strict matching. With approximate recursive matching, an event can match the gold event, even if it is only partially correctly predicted [7]. In approximate recursive matching the predicted spans do not have to be equal to the gold spans for passing the equivalence criteria. Also, only Theme arguments are considered for approximate recursive matching.

Next, we discuss the prominent approaches to BEE. Biomedical event extraction has a long-standing and enduring presence within the field with neural approaches reporting the most successful results [8, 9, 10, 11, 12]. In early approaches joint learning was used by stacking multiple models on top of one another to extract the different event components (event trigger, event type, arguments) [8]. In this respect, other early approaches use a multi-layered processing pipeline [9]. Here, every step of the processing pipeline is dedicated to extracting an event component.

Recent work leverages transformer-based language models like BERT [13] for mitigating propagation errors [11, 14]. Hai-Long Trieu et al. [10] propose a single neural model named DeepEventMine that takes the BERT model and processes events in an end-to-end manner. However, the model still consists of multiple layers focusing on different event components. Hai-Long Trieu et al. [10] claim state-of-the-art results for multiple BEE data sets on strict and approximate span matching.

There are three recent approaches closest to our approach: BERT QA [15], BeeSL [11], and the text-to-graph framework by Frisoni et al. [14]. In Bert QA, Wang et al. [15] model BEE as an iterative question-answering (QA) process and train a single model on answering predefined questions. In their approach, the BEE task is transformed into a different task where the event graph has to be decoded from the language model output. Wang et al. [15] also proposed BEEDS, an extension of the QA approach using distantly supervised learning [12]. Ramponi et al. [11] introduce BEE as sequence labeling (BeeSL). They employ a BERT-based neural model by using multi-labeling and multi-task learning. However, they enable it at the token-level for multi-label sequence labeling [11].

All approaches mentioned so far do not use generative language models for solving BEE. Recently, Frisoni et al. [14] trained two different generative language models with a similar setup as we do and claim to achieve remarkably good results [14]. For one, they also fine-tuned a T5-base language model as well as BART-base model [16] but train it on text-to-graph generation instead of translation between augmented languages. Frisoni et al. [14] claim that TANL’s event annotation does not consider nested or overlapping events, which is why they chose to generate the event graph directly from the model and not regenerate the input sequence at all. This causes that the generated events are not accompanied by spans and the results are therefore only evaluated on approximate recursive matching. Note that in our approach we redesigned the original NLD proposed by Paolini et. al [2] to include nested or overlapping events.

2.2. TANL

Recent work has shown that sequence-to-sequence models like T5 can solve a variety of NLP tasks better than the baselines [1]. Translation between Augmented Natural Languages (TANL) is a new approach proposed by Paolini et al. [2] of translating natural language into an augmented language description which can be decoded into a structured object by a rule-based post-processing step [17]. The structured object could, for instance, be a graph that represents relations between entities inside the input text.

By framing the event extraction task as a translation task, the TANL framework can be used to identify relations, events, and entities inside of text. Paolini et al. [2] used the pre-trained generative language model T5 to perform a variety of NLP tasks. The results show that TANL outperforms many state-of-the-art frameworks on RE and NER tasks. The results on event extraction were comparable to other frameworks [17]. However, no biomedical data was used in any of the experiments.

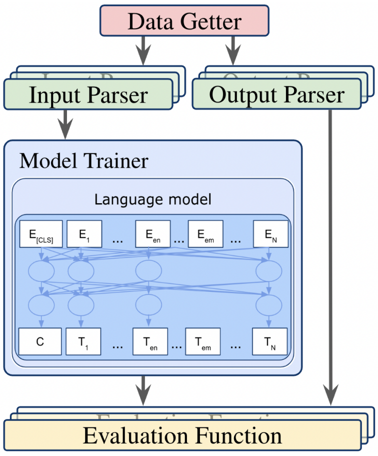
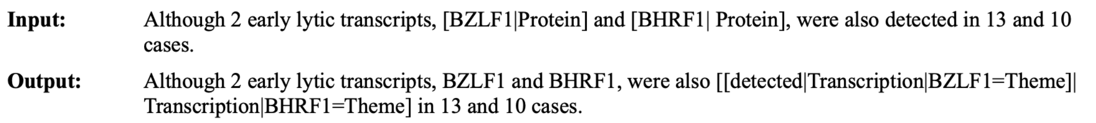


Figure 2: The processing pipeline of TANL. The first component is the data getter which is used for downloading and preprocessing various NLP data sets for tasks like event extraction (EE), relation extraction (RE), or named entity recognition (NER). The second and third components are the input and output parsers which generate input or target sequences based on a pre-defined NLD. The fourth component is the model trainer. The fifth component is the evaluation function for decoding the output sequence and evaluating the results.

The processing pipeline of the TANL framework is split into a data getter, an input parser, an output parser, a model trainer, and an evaluation function (See Figure 2). All parts can be adapted and modified for any NLP task or data set. The input and output parser generate the input and target sequence respectively. The evaluation function extracts the information from generated sequence and builds the information graph.

2.3. Language Description Design

In order to facilitate the sequence-to-sequence model, the inputs and targets of a given task have to be encoded into sequences in TANL. This natural language description is based on a mark-up language where task-specific information is highlighted with square brackets (see Figure 3). It follows a strict pattern to later be able to extract the information easily. The framework provides multiple adaptations to solve a variety of information extraction tasks. See Figure 3 for an example.



Input: Although 2 early lytic transcripts, [BZLF1|Protein] and [BHRF1| Protein], were also detected in 13 and 10 cases.

Output: Although 2 early lytic transcripts, BZLF1 and BHRF1, were also [[detected|Transcription|BZLF1=Theme]| Transcription|BHRF1=Theme] in 13 and 10 cases.

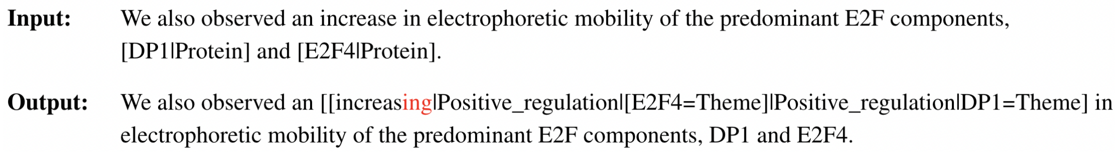
Figure 3: An example sequence from [4] parsed into the an input and an output sequence. The entities and events are described in natural words inside the bracket special tokens [17].

Paolini et al. [2] also propose an adaptation for event extraction that covers data sets with a small number of event types and flat target structures. However, this adaptation does not support nested event extraction and higher-order event extraction. Nested events are events that have the same event trigger but different event types or arguments. Events with other events as an argument are called higher-order events. Both types are more difficult to extract because the model also needs to consider knowledge about other events in a sequence. However, typical BEE data sets contain deep event structures, which means nested or higher-order events are very common. We therefore extended the NLD for the BEE task to support nested event extraction as well as higher-order event extraction.

3. TANL for BEE

We modified the input and output of TANL to generate task-specific input and target sequences (see Figure 3). For the input sentence, we tagged the given gold entities in the sentence with square brackets and added the corresponding entity type as context for the model inside the brackets as this is standard for BioNLP evaluation. The entity name and the type are separated by a pipe character. The target sentences are annotated by framing the event trigger in square brackets. Inside the brackets, the event type and the arguments are added and separated by a pipe. We used a nested pattern for the event extraction target sentences similar to the nested entities proposed by Paolini et al. [2]. With this modification, multiple events can be extracted from a single sequence.

To evaluate the generated sentences, we built a function to extract the tagged information and transform it into a standoff format [7]. For this, we used a built-in function of TANL which is searching for square brackets in the sentence and extracting the information inside of them. The function can handle simple and nested events. However, around 10 % of the generated output sentences have different text (disregarding the annotations) than the original sequence. Such reconstruction errors are documented over various data sets [2]. These errors happen because of typos or smaller falsely-generated tokens. Figure 4 shows an example of false-generated output tokens from our experiments.



Input: We also observed an increase in electrophoretic mobility of the predominant E2F components, [DP1|Protein] and [E2F4|Protein].

Output: We also observed an [[increasing|Positive_regulation|[E2F4=Theme]|Positive_regulation|DP1=Theme] in electrophoretic mobility of the predominant E2F components, DP1 and E2F4.

Figure 4: An input example sequence from [4] and a generated output example. The model generated ”increasing” instead of ”increase” as event trigger but the Needleman-Wunsch algorithm mitigates those errors [17].

Especially incorrectly generated triggers can cause entire events to be incorrectly predicted, reducing the accuracy of the model. To repair such cases, string alignment is implemented [2], based on the Needleman-Wunsch algorithm [18]. For the evaluation of the information extraction, the extracted arguments of an event have to be matched with the given gold entities and events. If more than one gold event name or gold entity name matches with the predicted argument name, we use the same technique as proposed in TANL by picking the matching token that is the closest relative to the event trigger in the sequence [2].

The predicted events are parsed into a standoff format to be evaluated by standardized evaluation scripts provided by the BioNLP shared task organizers from the corresponding shared task of the evaluated data set [4, 19].

4. Experiment Setup

4.1. Datasets

For obtaining train and evaluation data, we used the BigBio framework that enables easy programmatic access to over 120 biomedical data sets [20]. With that framework, we could access all used BEE data sets in a unified format.

We trained the T5-Base model with our TANL-based BEE formulation on two different data sets separately: The Pathway Curation (PC) task of the BioNLP Shared Task from 2013 [19] and the Genia11 Event Task of the BioNLP Shared Task from 2011 [4]. Both data sets consist of annotated PubMed abstracts and full texts.

- The **PC** data set focuses on pathway relations and consists of 525 documents with a train, validation, and test split distribution of 55% to 10% to 35%. There are 24 different event types represented in the data set and four distinct entity types. The data set distinguishes between multiple conversion types, such as degradation or dissociation, besides the common event types like (positive/negative) regulations, binding, or phosphorylation.

- The **Genia11** event task consists of 1210 abstracts and 14 full papers with a train, validation, and test split distribution of 65% to 13% to 22% for abstracts and 35% to 35% to 30% for full papers. The data set contains 16,416 sentences and nine different event types. The data set distinguishes between six different argument types. Those arguments can either be entities or other events.

Table 1
Statistics of the PC and the Genia11 corpora [4, 19].

	PC			Genia11		
Item	Train	Val.	Test	Train	Val.	Test
Documents	260	90	175	805	155	264
Words	53811	18579	35966	205729	64132	79047
Sentences	2761	951	1856	9551	3154	3711
Entities	7855	2734	5312	11625	4690	5301
Events	5992	2129	4004	10310	3250	4487
Nested Events	2412	852	-	4226	1532	-
Higher-Order Events	2262	813	-	3833	1131	-

Note that there are more than ten different BEE data sets from various shared tasks. However, we chose the PC data set because it is a relatively small data set with a high information density. The data set tests the limits of the TANL framework. The Genia11 data set is one of the largest BEE data sets containing event types represented in all other data sets. Therefore, the Genia11 data set gives a representative comparison to the baselines.

For both data sets, the gold events and entities are provided for the validation set and the train set. However, for the test set, only the entities are provided. There is an evaluation website for the Genia11 data set where it is possible to send the .a* files to a server and receive a test set evaluation.[1] Unfortunately, there is no such website for the PC data set. Therefore, we are not able to evaluate the PC data set on the test set.

4.2. Baseline Methods

As baselines, we used (1) the BERT-based framework Deep Event Mine (DEM) [10] and (2) a framework that used a BERT model with multi-turn question answering (QA) for the event extraction task [15]. Deep Event Mine is an end-to-end neural nested event extraction model with state-of-the-art results in seven biomedical data sets [10]. The QA framework transforms the event extraction task into a question-answering task to extract the event. We also compared our results to a third model called TEES CNN [21]. This framework uses a convolutional neural network for both event and relation extraction [21]. All baselines are evaluated on the PC and Genia11 validation sets and the Genia11 test set and we take results from the respective publications. Note that we did not use the text-to-graph framework by Frisoni et al. [14] as baseline as the results are conducted with a different equivalence and evaluation metric (see Section 2.1).

[1] http://bionlp-st.dbcls.jp/GE/2011/eval-test/

4.3. Hyperparameters

Paolini et al. [2] report that they used the same hyperparameters for the majority of experiments, across tasks. We, therefore, decided to select for both datasets similar hyperparameters to those suggested by Paolini et al. [2]. We tested additional hyperparameters on the Genia11 dataset but did not observe any significant improvement. We used a batch size of 64, a linear learning rate decay starting at 0.0005 with the AdamW optimizer [22], 200 warm-up steps, and a maximum input/output sequence length of 1024 tokens (longer sequences are truncated). We fine-tuned all datasets for 200 epochs, as suggested for datasets of this size [2]. We conducted our experiments on four Nvidia GeForce RTX 2080 Ti GPUs with a runtime of roughly 9 hours per experiment.

5. Results

We evaluated the results of the Genia11 data set on the validation and test split. However, the PC data set is evaluated on the validation split only because the gold standard events for the PC test split are not publicly accessible anymore (see Section 4.1).

Results can be found in Table 2. For the PC evaluation, we used a strict matching evaluation mode and for the Genia11 we used approximate span and approximate recursive matching mode to be comparable with prior work [15].

Table 2
Results of the TANL BEE model and three different baselines.

	Genia11 *test set*			Genia11 *dev set*			PC *dev set*		
Models	Prec.	Recall	F1	Prec.	Recall	F1	Prec.	Recall	F1
TANL BEE	61.00	46.29	52.63	59.15	46.72	52.26	49.18	44.33	45.57
DEM[10]	**71.71**	56.20	**63.02**	**70.52**	**56.52**	**62.75**	**65.94**	**49.52**	**56.57**
TEES CNN[21]	64.86	50.53	56.80	64.57	53.59	58.57	50.29	42.51	46.07
Bert QA[15]	59.33	**57.37**	58.33	56.41	56.58	56.50	45.90	43.37	44.60

The TANL-BEE model falls behind in most baselines for all used corpora. On the Genia11 test set, the TANL framework is 4.17 percent points (pp) F1 worse than TEES CNN, and on the validation set, TANL is 6.31 pp F1 worse than TEES CNN. It achieves a 0.97 pp better F1 score than BERT QA in the PC validation split but the score is 0.5 pp lower than the results of TEES CNN and 11 pp lower than Deep Event Mine. Although the model does not outperform baselines on any data sets we can see that TANL-BEE sometimes comes close to some of the baselines on event extraction.

6. Discussion

For a comprehensive evaluation of the results, we analyze event detection and argument detection separately. Table 3 shows the result from the Genia11 dev set from Section 5 for event detection only. The results show that TANL BEE has a higher precision in detecting events than DEM yet a worse recall. The F1 difference is reduced from roughly 10.5 pp to 4.26 pp compared to the complete event extraction task. Also, the results highlight that the performance of TANL decreases by 14,25% when the event extraction task is extended with argument detection. Deep Event Mine loses 6.13 fewer pp when argument matching is added which is around half of the percentage points compared to our framework [10]. We next study whether this performance decrease can be explained by the argument matching heuristic of TANL BEE.

Table 3
Event detection results on the Genia11 dev set.

Models	Precision	Recall	F1
TANL BEE	**78,29**	57,82	66.51
DEM[10]	71.23	**70.31**	**70.77**

Argument matching describes the task of finding the span of an entity or event in a sequence with multiple entities or events with the same trigger name. In the TANL framework, the references in the second part of an event annotation e_1 (after the first pipe-separator) are located by searching for the entity closest to the given event trigger t_1 of e_1. [2] show that this approximation is good enough for the tasks that the authors evaluate. However, we show that this greedy approach is not suitable for the complex task of argument matching in BEE.

We evaluated the results of the T5 model with an optimal argument linker. That means we linked the gold entities directly to the arguments when the predicted argument has the same entity name and argument type as the gold arguments (see Table 4). The T5 model with an optimal argument linker has a 4,86 pp higher score than the base model. Unfortunately, other baselines like Deep Event Mine do not report the error of false argument matching [10].

We also evaluated TANL-BEE on the Genia11 11 development set with three other evaluation settings to provide deeper insides into the complexity of argument matching in event extraction. When more than one entity was found as a possible argument:

- We took the closest entity relative to the trigger as the argument entity (T5 Base in Table 4), or
- we took the farthest argument (T5 Base with farthest arg. finder in Table 4), or
- we took a random entity from the matching entities (T5 Base with random arg. finder in Table 4).

The results show that the F1 score does not change more than 1% for these three evaluation techniques. However, roughly 10% of all events contain at least one argument with multiple possible entity triggers in their sequence and roughly 13% of all events contain at least one argument with multiple possible event triggers in their sequence. Our assumption is, that the argument detection task of event extraction is far more complex in BEE than in other event extraction data sets. A simple heuristic approach does not solve the problem of argument matching with sufficient accuracy.

Frisoni et al. [14] did not use any heuristic post-processing step to map the spans to the triggers because they hypothesized that the high type heterogeneity overhead and sequences with multiple identical event or argument triggers with different spans would lower the performance [14]. This assumption can be supported by our result on argument matching.

Table 4
Results on Genia11 *dev set* with four different TANL setups.

TANL Models	Precision	Recall	F1
T5 Base	59.15	46.72	52.26
T5 Base with random arg. finder	58.02	46.00	51.32
T5 Base with furthest arg. finder	57.95	46.65	51.69
T5 Base with optimal arg. finder	**64.04**	**51.56**	**57.12**

7. Conclusion

For this paper, we analyzed the performance of a generative language model for BEE by transforming the BEE task into a translation task. We modified the original NLD by Paolini et al. [2] to support nested event extraction, modifiers detection, or event overlapping and still fulfill the requirements for the BioNLP shared task evaluation. Our results show no improvement compared to the state-of-the-art. We outline the major drawbacks of the model affecting its performance. The major drawback is the natural language description itself and especially the argument matching of entities and events. We think that the information extraction from the sentences could be improved by adapting the natural language description for the output prompts for better argument-to-entity linking.

For a more robust TANL-BEE two components have to be improved: the NLD design and the model size. We argue that the information extraction from the sentences might be improved by adapting the output prompts for a better argument to entity linking. We hypothesize that a bigger model than T5-Base could improve the accuracy. We also show in preliminary experiments that a biomedical language model like SciFive [23] did not improve the score.

For future work, we introduce a new approach to assess the performance of generative large language models on biomedical event extraction in a few-shot learning setting. Specifically, we train a generative large language model using only a few examples (ten to twenty-five) within in-context prompts.

References

[1] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li and Peter J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, in: Journal of Machine Learning Research, volume 21, Massachusetts, USA, 2020.

[2] Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, Rishita Anubhai, Cicero Nogueira dos Santos, Bing Xiang and Stefano Soatto, Structured prediction as translation between augmented natural languages, in: International Conference on Learning Representations, Seattle, USA, 2021.

[3] Esther Landhuis, Scientific literature: Information overload, Nature 535 (2016) 457–458.

[4] Jin-Dong Kim, Yue Wang, Toshihisa Takagi, and Akinori Yonezawa, Overview of genia event task in bionlp shared task 2011, in: Proceedings of the BioNLP Shared Task 2011

Workshop, BioNLP Shared Task ’11, Portland, USA, 2011, p. 7–15.

[5] Kevin Bretonnel Cohen and Dina Demner-Fushman, Biomedical natural language processing, volume 11, John Benjamins Publishing Company, 2014.

[6] George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel and Ralph Weischedel, The automatic content extraction (ACE) program – tasks, data, and evaluation, in: Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC’04), European Language Resources Association (ELRA), Lisbon, Portugal, 2004.

[7] Jin-Dong Kim, Tomoko Ohta, Sampo Pyysalo, Yoshinobu Kano, and Jun’ichi Tsujii, Extracting bio-molecular events from literature—the bionlp’09 shared task, in: Computational Intelligence, volume 27, Wiley Online Library, Sanya, Hainan, China, 2011, pp. 513–540.

[8] Sebastian Riedel, David McClosky, Mihai Surdeanu, Andrew McCallum and Christopher D. Manning, Model combination for event extraction in BioNLP 2011, in: Proceedings of BioNLP Shared Task 2011 Workshop, Association for Computational Linguistics, Portland, Oregon, USA, 2011, pp. 51–55.

[9] J. Björne, T. Salakoski, Tees 2.2: Biomedical event extraction for diverse corpora, BMC bioinformatics 16 (2015) S4. doi:10.1186/1471-2105-16-S16-S4.

[10] Hai-Long Trieu, Thy Thy Tran, Khoa N. A. Duong, Anh Nguyen, Makoto Miwa and Sophia Ananiadou, Deepeventmine: end-to-end neural nested event extraction from biomedical texts, in: Bioinformatics, volume 36, Oxford, England, 2020, pp. 4910–4917. doi:10.1093/bioinformatics/btaa540.

[11] Alan Ramponi, Rob van der Goot, Rosario Lombardo and Barbara Plank, Biomedical event extraction as sequence labeling, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Online, 2020, pp. 5357–5367. doi:10.18653/v1/2020.emnlp-main.431.

[12] U. L. Xing D. Wang, L. Weber, BEEDS: Large-scale biomedical event extraction using distant supervision and question answering, in: Proceedings of the 21st Workshop on Biomedical Language Processing, Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 298–309. doi:10.18653/v1/2022.bionlp-1.28.

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee and Kristina Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186. doi:10.18653/v1/N19-1423.

[14] Giacomo Frisoni, Gianluca Moro, and Lorenzo Balzani, Text-to-text extraction and verbalization of biomedical event graphs, in: Proceedings of the 29th International Conference on Computational Linguistics, International Committee on Computational Linguistics, Gyeongju, Republic of Korea, 2022, pp. 2692–2710.

[15] Xing D. Wang, Leon Weber and Ulf Leser, Biomedical event extraction as multi-turn question answering, in: Proceedings of the 11th International Workshop on Health Text Mining and Information Analysis, Association for Computational Linguistics, Online, 2020, pp. 88–96. doi:10.18653/v1/2020.louhi-1.10.

[16] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed,

Omer Levy, Ves Stoyanov and Luke Zettlemoyer, BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 7871–7880. doi:10.18653/v1/2020.acl-main.703.

[17] Hiroaki Ozaki, Gaku Morio, Yuta Koreeda, Terufumi Morishita, and Toshinori Miyoshi, Hitachi at MRP 2020: Text-to-graph-notation transducer, in: Proceedings of the CoNLL 2020 Shared Task: Cross-Framework Meaning Representation Parsing, Association for Computational Linguistics, Online, 2020, pp. 40–52. doi:10.18653/v1/2020.conll-shared.4.

[18] Maroš Čavojský and Martin Drozda and Zoltán Balogh, Analysis and experimental evaluation of the needleman-wunsch algorithm for trajectory comparison, in: Expert Systems with Applications, volume 165, Amsterdam, Niederlande, 2021. doi:https://doi.org/10.1016/j.eswa.2020.114068.

[19] Tomoko Ohta, Sampo Pyysalo, Rafal Rak, Andrew Rowley, Hong-Woo Chun, Sung-Jae Jung, Sung-Pil Choi, Sophia Ananiadou and Jun'ichi Tsujii, Overview of the pathway curation (PC) task of BioNLP shared task 2013, in: Proceedings of the BioNLP Shared Task 2013 Workshop, Sofia, Bulgaria, 2013, pp. 67–75.

[20] Jason Fries et al., Bigbio: A framework for data-centric biomedical natural language processing, in: Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track, volume 35, Curran Associates, Inc., New Orleans, USA, 2022, pp. 25792–25806.

[21] Jari Björne and Tapio Salakoski, Biomedical event extraction using convolutional neural networks and dependency parsing, in: Proceedings of the BioNLP 2018 workshop, Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 98–108. doi:10.18653/v1/W18-2311.

[22] Diederik P. Kingma and Jimmy Lei Ba, Adam: A method for stochastic optimization, in: Y. Bengio, Y. LeCun (Eds.), Conference Track Proceedings of the 3rd International Conference on Learning Representations ICLR, San Diego, CA, USA, 2015.

[23] L. N. Phan, J. T. Anibal, H. Tran, S. Chanana, E. Bahadroglu, A. Peltekian, G. Altan-Bonnet, Scifive: a text-to-text transformer model for biomedical literature (2021). arXiv:preprint arXiv:2106.03598, 2021.