Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 1211–1216
DOI: 10.15439/2023F2076
ISSN 2300-5963 ACSIS, Vol. 35

Adding Linguistic Information to Transformer Models Improves Biomedical Event Detection?

1st Laura Zanella
LORIA (Université de Lorraine, CNRS, Inria)
Nancy, France
laura-alejandra.zanella-calzada@loria.fr

2nd Yannick Toussaint
LORIA (Université de Lorraine, CNRS, Inria)
Nancy, France
yannick.toussaint@loria.fr

***Abstract*—Biomedical event detection is an essential subtask of event extraction that identifies and classifies event triggers, indicating the possible construction of events. In this work we propose the comparison of BERT and four of its variants for the detection of biomedical events to evaluate and analyze the differences in their performance. The models are learned using seven manually annotated corpora in different biomedical subdomains and fine-tuned by adding a linear layer and a Bi-LSTM layer on top of the models. The evaluation is done by comparing the behavior of the original models and by adding a lexical and a syntactic features. SciBERT emerged as the highest performing model when the fine-tuning is done using a Bi-LSTM layer, without need of extra features. This result suggests that the use of a transformer model that is pretrained from scratch and uses biomedical and general data for its pretraining, allows to detect event triggers in the biomedical domain covering different subdomains.**

***Index Terms*—Biomedical Event Extraction, Event Detection, Transformer Language Models, Named Entity Recognition**

I. Introduction

BIOMEDICAL event extraction is a complex information extraction task that identifies key information from large sets of textual data for further applications, such as the study of biomolecular mechanisms or epigenetic changes. A biomedical event is constructed from an event trigger and one or more arguments that orbit around the trigger. Event triggers generally refer to nouns or verbs that express an action, circumstance or eventuality, while the arguments refer either to biomedical entities or to other events, called nested events. Fig. 1 shows the example of a sentence annotated with two biomedical events, '-Reg' (which stands for 'Negative regulation') and 'Locl' (which stands for 'Localization'). The event 'Locl' (the event is given the same type as the trigger) that is constructed from the trigger word 'excretion' presents as argument the biomedical entity of the type 'D/C' (which stands for 'Drug or compound'), who plays the role 'Th' (which stands for 'Theme'). This role allows answering the question 'What is excreted?'. While the event '-Reg', constructed from the trigger word 'reduces', presents two arguments. The first argument is a biomedical entity of the type 'Drug or compound', who plays the role 'Cause'. This role allows answering the question 'What causes the reduction?'. The second argument is the nested event 'Locl' described before, who plays the role 'Theme', answering the question 'What is reduced?'.

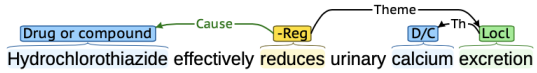

Fig. 1. Example of event extraction; the '-Reg' (negative regulation) event has the 'Locl' (localization) nested event as argument.

Event extraction is usually divided into three main sub-tasks, event detection, argument identification and event construction. Event detection identifies and classifies the trigger words into a set of predefined types of event triggers, while argument identification identifies and classifies the corresponding event arguments and their respective roles [1]. Event construction refers to the merging of the relations that correspond to the same event. This work focuses on event detection, which has a fundamental role in the construction of events, since the triggers are the targets that allow to know that an event may exist [2]. Difficulty for trigger detection comes from the sensitivity to the domain or subdomain (text can present specialized language), linguistic forms (triggers can be single words, multi-words, discontinuous markers) and ambiguity on the trigger class (a trigger word can be given different trigger classes) [3]. According to different works, such as in [1], solutions to address these issues may include additional features to provide lexical, syntactic and semantic information about text, which have proven to be useful for detecting event triggers. Transformers models have been adopted for event detection due to their positive achievements in performance for solving different Natural Language Processing (NLP) tasks [4], [5]. BERT [6], which stands for Bidirectional Encoder Representations from Transformers, is pretrained to generate bidirectional representations of the words, taking into account the semantics by considering both left and right directions of the text. From this pretraining, BERT can be fine-tuned by including additional layers on top of the model to solve new specific tasks. Furthermore, a series of variants from BERT have been developed for specific domains by being trained on large corpus with the same context, such as the biomedical domain.

In this work we compare BERT and four of its variants pretrained in the biomedical domain for the detection of biomedical event triggers to analyze their performance and identify which model is the most appropriate to address this task. For this purpose, BERT, BioBERT, SciBERT, Pub-

MedBERT, and BioMedRoBERTa are fine-tuned using two different classifiers, a linear layer and a Bidirectional Long Short Term Memory (Bi-LSTM) layer, to detect biomedical event triggers. These BERT variants have been chosen for comparison because they share the same BERT architecture but have previously been pretrained using different data in the biomedical and/or general domain [7]–[9]. The models are learned using seven manually annotated data sets merged together. These corpora were originally developed for the event extraction task in different biomedical subdomains. In addition to these data, two features are included as lexical and syntactical extra-information to the models, the stems and the parts-of-speech (POS) tags, respectively. SciBERT presented the highest performance when the fine-tuning is done using a Bi-LSTM classifier without adding any extra-features. This result suggests that using a transformer model that is pretrained from scratch using biomedical and general domain data, allows to detect biomedical event triggers addressing different biomedical subdomains.

Our main contributions refer to the (1) comparison of the capability of different pretrained transformer models to detect biomedical events, (2) evaluation of the performance of two different classifiers for the fine-tuning of event detection, (3) analysis of the impact of manually annotated corpora on different biomedical subdomains to detect event triggers, and (4) assessment of whether adding lexical and syntactic information improves biomedical event detection.

II. Related Work

Current SOTA systems for event detection use neural network models due to their robust event extraction capabilities.

P. V. Rahul et al. [10] used Recurrent Neural Networks (RNN) to extract higher level features through the hidden state of the network to identify biomedical event triggers. They also used the word and the entity type embeddings as features, demonstrating positive results in the MLEE [11] corpus. S. Duan et al. [12] and Y. Zhao et al. [13] explored an augmentation of the semantic information by integrating the full document representation. Both proposed the use of RNNs to extract cross-sentence features without the use of external resources. T. H. Nguyen and R. Grishman [14] presented a Graph Convolution Network (GCN) model to exploit syntactic dependency relations. They used dependency trees to link words to their informative context for event detection. H. Yan et al. [15] also proposed a GCN model, integrating aggregative attention to model and aggregate multi-order syntactic representations of the sentences, while in the case of S. Cui et al. [2], they extended the GCN by adding the relation aware concept, which exploits the syntactic relation labels and models the relation between words. DeepEventMine [16] is an end-to-end system for event extraction that consists on four main modules; BERT model, trigger and entity detection and classification, relation extraction and event identification. For each of the modules, BERT is used as base model and a linear layer is added. One of the main objectives of this system is improving the extraction of nested events, where it has achieved the new SOTA performance on seven biomedical nested event extraction tasks. B. Portelli et al. [17] compared BERT and five of its variants for the identification of Adverse Drugs and Events (ADEs). They showed that span-based pretraining, from spanBERT, provides an improvement in the recognition of ADEs, and that the pretraining of the models in the specific domain is particularly useful in comparison to train the models from scratch. A. Ramponi et al. [18] developed BEESL, a neural network model based on a sequence labeling system for the extraction of events. The system converts the event structures into a format of sequence labeling, and uses BERT as language model. Y. Chen [19] proposed the Multi-Source Transfer Learning-based Trigger Recognizer system, which is an extension on transfer learning using multiple source domains. All the datasets from the different domains are used for jointly train the neural network, achieving a higher recognition performance on the biomedical domain, having a wide coverage of events.

According to these works, transformer architectures have achieved positive results for detecting event triggers, and the use of pretrained language models has shown an improvement in the performance of this task. However, these works have been developed in a specific biomedical subdomain or in the general domain, not allowing a generalization to different biomedical subdomains. This may present a limitation in the detection of biomedical triggers because the language in biomedical texts is usually specialized and very specific. In addition, an analysis on how the pretrained language models used were selected over the other existing models is not described. Besides, according to A. Ramponi et al. [18], the detection of triggers continues to be the most important source of errors in event extraction, where around 31 % of the errors correspond to non-detection of triggers and 28 % to over-detection of triggers.

III. Materials and Methods

Fig. 2 shows the approach followed in this work. The annotated data is given as input to the pretrained transformer models to calculate the embeddings. The models used are BERT and four of its variants, who have achieved state-of-the-art performance in different NLP tasks without requiring major architectural modifications according to the specific tasks. In addition, the embeddings of a lexical and a syntactic features are also calculated. Then, a classification layer is added on top of the models for fine-tuning to detect event triggers.

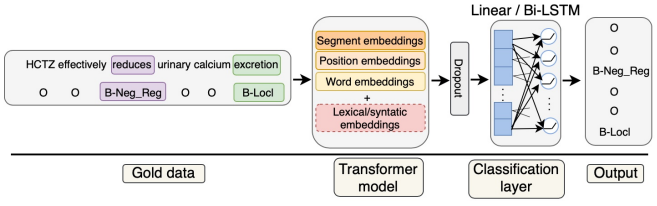

Fig. 2. Overview of the approach proposed to detect event triggers.

A. Transformer Model: BERT

BERT [6] is a contextualized word representation model based on a masked language model pretrained with bidirectional transformers [7]. In BERT, the sequence of input tokens (words or sub-words) is constituted with initial vectors that are the combination of the token embeddings, the (token) position embeddings and the segment embeddings (text segment to which the token corresponds) through element-wise summation. The embeddings of extra features can be computed and included in this summation, such as the POS embeddings (token function in meaning and grammar within the sentence), which has demonstrated to be helpful in detecting event triggers [1]. The embeddings are then passed to a set of layers of transformer modules. Each transformer layer generates a contextual representation of every token by summing the non-linear transformation of the tokens' representations from the previous layer. This representation is weighted by the attentions calculated using the representations of the previous layer as query. The last layer generates the contextual representations for all the tokens, where the information of the whole text span is combined [20]. Following the BERT principle, other transformer models have been developed being pretrained with data from specific domains, e.g. biomedical data, presenting better adaptation for solving in-domain tasks. BioBERT [7] and BioMedRoBERTa [21] are some examples of BERT variants pretrained in the biomedical domain.

B. Fine-Tuning Transformer Models for Event Detection

Various downstream text mining tasks can be performed by making minimal modifications to the BERT architecture through a process of fine-tuning. Here, the transformer models are fine-tuned following the Named Entity Recognition (NER) task. NER is one of the main tasks of biomedical text mining, which aims to recognize domain-specific nouns in a biomedical corpus by giving each word s_i in a sentence $S = s_1, s_2, ..., s_n$ (n refers to the number of words in the sentence) a predefined class $l \in L$ (where L refers to the predefined collection of entity types including the no-entity class). In this work, NER is adapted to identify triggers, which implies not only identifying nouns, but also verbs and in some cases adjectives. Two different classification layers, a linear layer and a Bi-LSTM layer, are used separately for comparison. The output labels are obtained following the IOB (Inside-Outside-Beginning) tagging to identify and classify the triggers into the predefined trigger categories (in the case of the I and B tags).

IV. Experimental Settings

A. Corpus

Table I presents the seven datasets [1] (all publicly available) used for fine-tuning the transformer models. These corpora were manually or semi-manually annotated by experts and released to be used in the development and improvement of event extraction models.

TABLE I
STATISTICS OF THE CORPUS USED

Dataset	No. Triggers	Trig Classes	Documents	Train/Dev/Test
CG 2013	9,790	35	PubMed abstracts	300/100/200
EPI 2011	2,035	14	PubMed abstracts	600/200/400
GENIA 2011	10,210	10	MEDLINE abstracts	1,000 (total)
GENIA 2013	4,676	12	PMC full-text	34 (total)
ID 2011	2,155	10	PMC full-text	15/5/10
PC 2013	6,220	22	PubMed abstracts	260/90/175
MLEE	5,554	15	PubMed abstracts	131/44/87

For the development of the experiments, the training and development datasets of all the corpora are initially merged into one single dataset and split into sentences, obtaining a total of 24,819 sentences. The original test sets are not used since the annotation are not released. Then, a random data partition into 80/20 is applied to obtain the training and testing sets, containing 19,855 and 4,964 sentences, respectively. Each sentence is further split into words by spaces and then, each word into sub-words or tokens following the setting of the BERT tokenization. These tokens are then given as input to the transformer model. All the trigger classes from each corpus are considered for the final trigger classification, presenting a final set of 58 classes (some classes overlap among the different corpora).

B. Pretrained Transformer Models

The transformer model, BERT [6], and four BERT variants pretrained in the biomedical domain, BioBERT [7], SciBERT [8], PubMedBERT [20], and BioMedRoBERTa [21], are used and compared for the detection of event triggers. These models differ from each other by the corpora in which they were pretrained (all in English), the type of pretraining and the size of the vocabulary. SciBERT and PubMedBERT, were pretrained from scratch, meaning that they use a unique vocabulary on the pretraining corpus and include embeddings that are specific for in-domain words. BioBERT and BioMedRoBERTa were pretrained starting from the BERT checkpoints, which means that the vocabularies are built with general-domain texts (similar to BERT) as well as the initialization of the embeddings.

C. Lexical and Syntactic features

The embeddings of stems and POS tags are also computed and added as extra-features. Stems provide lexical information that correspond to the words reduced to their word roots, without needing to be an existing word in the dictionary. Stems are obtained by applying a set of rules to remove attached suffixes and prefixes (affixes) from terms without considering the POS or the context of the word occurrence [27]. POS tags represent syntactic information that provides the categorical differences of the words according to their functions in meaning and grammatically within the sentence. POS tagging consists on automatically obtaining the POS tag of each word among the different POS categories corresponding to their

[1]Cancer Genetics (CG) 2013 [22], Epigenetics and Post-translational Modifications (EPI) 2011 [23], GENIA 2011 [24], GENIA 2013 [25], Infectious Diseases (ID) 2011 [26], Pathway Curation (PC) 2013 [22], Multi-Level Event Extraction (MLEE) [11]

syntactical role [28]. For this work, the stems of the words are obtained using the 'Snowball Stemmer' module from NLTK-3.4.5 [2], while the POS were obtained using spaCy-3.0.0 [3], using 'en_core_web_sm', a pipeline developed for biomedical data. The embeddings of the stems and POS tags are summed to the rest of the embeddings (token, position and segment) calculated by the transformer models.

D. Parameters Settings

All the experiments are done with PyTorch, using the Transformers [4] library and the models were taken from Hugging Face [5]. The transformer models are trained using the original parameters from BERT, presenting a dropout probability for the attention heads and hidden layers of 0.1, a hidden size of 768, an initializer range of 0.02, a max position embeddings of 512 and an intermediate size of 3,072. The number of attention heads and hidden layers was 12 for both. 'Adam' was used as optimizer and 'gelu' as activation function. The training parameters of the classification layers, both linear and Bi-LSTM, were set as follows; batch size of training and testing sets of 16, learning rate of 1e-05 and max gradient norm of 10, since gradient clipping was included. The maximum length of the sentences was set to 256. All the models were trained during 100 epochs on the training set without applying early stopping, and evaluated by measuring the precision (P), recall (R) and F1-score.

V. Results and Discussion

The evaluation results of the fine-tuning of the models for event detection are shown in Table II. The approximate time in hours for the fine-tuning of each model is presented in the last column of the table. The highest results obtained in epochs 10, 30 and 100 are presented in bold, and the highest overall results of all epochs are presented in bold and underlined. First, we observe that SciBERT, which was pretrained from scratch using biomedical and general data, obtained the best results for each number of epochs and overall, in P, R and F1. It presented higher values when Bi-LSTM was used as classifier, especially when extra features were not added or when the lexical feature is added in the case of the training for 10 epochs. When the training was done for more than 10 epochs, the performance between SciBERT+POS (syntactic feature) and SciBERT+stem (lexical feature) was very similar. When the fine tuning was done using a linear classifier, SciBERT+POS achieved the best results, having a difference of around 10 % to when the lexical feature (SciBERT+stem) is added. PubMedBERT, a model pretrained from scratch using biomedical data, achieved the second best performance, being below SciBERT by 4 % when the training is done for 30 epochs, using Bi-LSTM as classifier and no adding extra-features (which was the best overall result of SciBERT). When PubMedBERT used Bi-LSTM as classifier, the results were very similar between adding the syntactic or lexical features and not adding them. These results were also similar to when a linear classifier was used and the extra features are added, noticing that the result was worse when no features were added. In the case of BERT, which was trained from scratch using data from the general domain, it presented lower results than PubMedBERT by around 5 %. The best results of BERT were obtained using a linear classifier and not adding extra features, noticing that the results of BERT+POS and BERT+stem were slightly lower and very similar between each other. This same behavior can be noticed when Bi-LSTM was used as classifier. These three last transformer models, SciBERT, PubMedBERT and BERT, presented some similarities in that they were trained from scratch, used very comparable text sizes for their pretraining and had similar vocabulary sizes. The two models that presented the lowest performance are BioBERT and BioMedRoBERTa, both pretrained from the BERT weights, using biomedical and, biomedical and general data, respectively, presenting the largest text sizes of all the models. BioBERT used the smallest vocabulary for its pretraining, while BioMedRoBERTa used the largest in comparison to the rest of the models. In both models it was observed that there was not significant change when adding the extra features, although there was an improvement of around 7 % when using a Bi-LSTM classifier compared to a linear classifier. In general, what can be noticed in all the models is that adding the syntactic and lexical features does not improve the performance for detecting biomedical events.

Fig. 3 shows the performance of fine-tuning SciBERT during 30 epochs using a Bi-LSTM classifier on the seven datasets separately. The F1-scores obtained using EPI, CG, ID, GE'13 and PC were similar between each other, obtaining values between 0.70 and 0.80. When GE'11 was used, the F1-score reached a value of around 0.65 and when MLEE was used, the model completely failed the detection of triggers. In Fig. 4 it is observed the effect of fine-tuning SciBERT over 30 epochs using a Bi-LSTM classifier without adding extra-features by cumulatively adding each corpus one by one. Below each corpus is shown the total number of classes by adding each corpus. Recall was improved when CG and EPI were used together, and then reduced as the rest of the corpora were added. Precision was affected when EPI and GE'11 were added. The behavior of recall and precision varied differently depending on the added corpus, although when GE'13 was added both values were comparable, and as might be expected according to the observed on Fig. 3, when MLEE was added the values were negatively affected. This behavior may be due to the fact that when adding a new corpus for the fine-tuning of the models, some classes may overlap between the corpora while other classes do not, causing to probably have less samples in the new classes and, therefore, affecting the balance of the data. In addition, the context of the different biomedical subdomains may also affect the performance, since BERT and its variants compute embeddings considering the semantics.

[2]https://www.nltk.org/_modules/nltk/stem/snowball.html
[3]https://spacy.io/
[4]https://github.com/huggingface/transformers
[5]https://huggingface.co/

TABLE II
RESULTS OF THE MODELS' FINE-TUNING FOR EVENT DETECTION

Classifier	Model	10 epochs			30 epochs			100 epochs			Time
		P	R	F1	P	R	F1	P	R	F1	(h)
Linear	BERT	0.57	0.67	0.62	0.60	0.68	0.64	0.62	0.68	0.65	13
	BERT+POS	0.58	0.61	0.59	0.62	0.63	0.62	0.64	0.64	0.64	14
	BERT+stem	0.62	0.58	0.59	0.67	0.57	0.61	0.66	0.62	0.63	18
Bi-LSTM	BERT	0.59	0.57	0.57	0.67	0.58	0.62	0.65	0.64	0.64	19
	BERT+POS	0.46	0.59	0.51	0.58	0.62	0.60	0.61	0.63	0.62	21
	BERT+stem	0.57	0.59	0.57	0.63	0.61	0.62	0.67	0.60	0.63	15
Linear	BioBERT	0.49	0.49	0.48	0.52	0.50	0.50	0.56	0.49	0.51	19
	BioBERT+POS	0.54	0.44	0.47	0.49	0.51	0.49	0.51	0.51	0.51	16
	BioBERT+stem	0.48	0.50	0.47	0.52	0.46	0.49	0.53	0.48	0.50	18
Bi-LSTM	BioBERT	0.60	0.39	0.45	0.60	0.56	0.58	0.64	0.56	0.59	14
	BioBERT+POS	0.57	0.39	0.44	0.59	0.55	0.57	0.61	0.55	0.58	15
	BioBERT+stem	0.54	0.50	0.50	0.61	0.52	0.56	0.59	0.57	0.58	20
Linear	SciBERT	0.59	0.64	0.61	0.61	0.65	0.63	0.70	0.70	0.70	11
	SciBERT+POS	**0.67**	**0.72**	**0.69**	0.69	0.71	0.70	0.72	**0.73**	**0.72**	16
	SciBERT+stem	0.56	0.62	0.58	0.61	0.62	0.61	0.64	0.62	0.63	13
Bi-LSTM	SciBERT	0.65	0.71	0.68	0.71	**0.73**	**0.72**	0.74	0.71	**0.72**	19
	SciBERT+POS	0.55	0.56	0.54	0.70	0.71	0.70	0.73	0.70	0.71	22
	SciBERT+stem	**0.67**	0.68	0.67	**0.72**	0.68	0.70	**0.75**	0.68	0.71	16
Linear	PubMedBERT	0.49	0.61	0.54	0.58	0.66	0.61	0.58	0.62	0.60	14
	PubMedBERT+POS	0.63	0.68	0.65	0.64	0.68	0.66	0.68	0.67	0.67	16
	PubMedBERT+stem	0.62	0.66	0.64	0.66	0.67	0.66	0.70	0.67	0.68	18
Bi-LSTM	PubMedBERT	0.57	0.65	0.61	0.66	0.69	0.67	0.67	0.69	0.68	19
	PubMedBERT+POS	0.58	0.65	0.61	0.67	0.66	0.66	0.69	0.67	0.68	17
	PubMedBERT+stem	0.59	0.66	0.61	0.66	0.69	0.67	0.70	0.66	0.68	18
Linear	BioMedRoBERTa	0.48	0.49	0.47	0.52	0.52	0.51	0.55	0.50	0.52	14
	BioMedRoBERTa+POS	0.52	0.56	0.53	0.55	0.51	0.52	0.55	0.53	0.54	13
	BioMedRoBERTa+stem	0.50	0.53	0.51	0.51	0.51	0.51	0.53	0.54	0.53	18
Bi-LSTM	BioMedRoBERTa	0.58	0.50	0.53	0.60	0.57	0.58	0.69	0.53	0.59	19
	BioMedRoBERTa+POS	0.51	0.56	0.52	0.61	0.53	0.56	0.62	0.56	0.58	15
	BioMedRoBERTa+stem	0.51	0.54	0.52	0.57	0.59	0.57	0.60	0.59	0.59	15

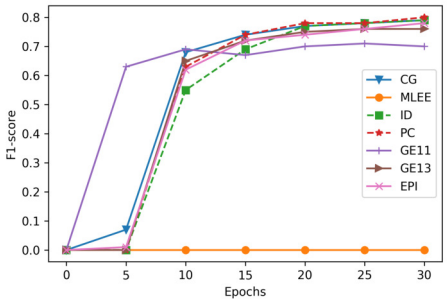

Fig. 3. Fine-tuning SciBERT on the different corpus (Bi-LSTM classifier).

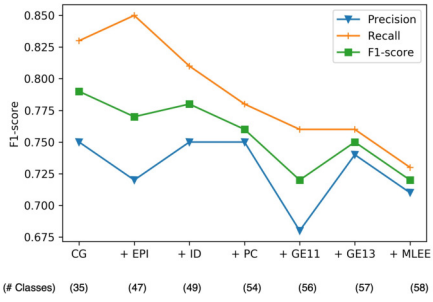

Fig. 4. Fine-tuning SciBERT by cumulatively adding the corpus one by one (Bi-LSTM classifier).

VI. CONCLUSIONS AND LIMITATIONS

In this work, we analyze BERT and four of its variants for biomedical event detection using corpora of different biomedical subdomains. By comparing the performance of the models and by adding a lexical and syntactic features, we found that fine-tuning SciBERT during 30 epochs using a Bi-LSTM classifier is the best strategy to detect biomedical events, especially if the additional features are not included. Furthermore, it is shown that fine-tuning the models for 10 to 30 epochs achieves most of the model learning, while training for more epochs can only achieve a slightly better result. One of the limitations of this work is the imbalance of the data. Since some classes of the different corpora overlap, the samples for those classes are increased, while the unique classes for each corpora present fewer samples. This can negatively affect the behavior of the models between the different subdomains. Also, using external tools to get POS tags and stems can lead to errors that are learned by the models and may be one of the reasons why performance without additional features achieves better results.

REFERENCES

[1] C. Shen, H. Lin, X. Fan, Y. Chu, Z. Yang, J. Wang, and S. Zhang, "Biomedical event trigger detection with convolutional highway neural network and extreme learning machine," *Applied Soft Computing*, vol. 84, p. 105661, 2019. doi: 10.1016/j.asoc.2019.105661

[2] S. Cui, B. Yu, T. Liu, Z. Zhang, X. Wang, and J. Shi, "Event detection with relation-aware graph convolutional neural networks," *arXiv e-prints*, pp. arXiv–2002, 2020.

[3] C. Zerva and S. Ananiadou, "Event extraction in pieces: Tackling the partial event identification problem on unseen corpora," in *Proceedings of BioNLP 15*, 2015. doi: 10.18653/v1/W15-3804 pp. 31–41.

[4] R. Hanslo, "Deep learning transformer architecture for named-entity recognition on low-resourced languages: State of the art results," in *2022 17th Conference on Computer Science and Intelligence Systems (FedCSIS)*. IEEE, 2022, pp. 53–60.

[5] K. Kaczmarek, J. Pokrywka, and F. Graliński, "Using transformer models for gender attribution in polish," in *2022 17th Conference on Computer Science and Intelligence Systems (FedCSIS)*. IEEE, 2022, pp. 73–77.

[6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018. doi: 10.18653/v1/n19-1423

[7] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang, "Biobert: a pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics*, vol. 36, no. 4, pp. 1234–1240, 2020. doi: 10.1093/bioinformatics/btz682

[8] I. Beltagy, K. Lo, and A. Cohan, "Scibert: A pretrained language model for scientific text," *arXiv preprint arXiv:1903.10676*, 2019. doi: 10.18653/v1/D19-1371

[9] A. Erdengasileng, Q. Han, T. Zhao, S. Tian, X. Sui, K. Li, W. Wang, J. Wang, T. Hu, F. Pan *et al.*, "Pre-trained models, data augmentation, and ensemble learning for biomedical information extraction and document classification," *Database*, vol. 2022, 2022. doi: 10.1093/database/baac066

[10] P. V. Rahul, S. K. Sahu, and A. Anand, "Biomedical event trigger identification using bidirectional recurrent neural network based models," *arXiv preprint arXiv:1705.09516*, 2017. doi: 10.18653/v1/W17-2340

[11] S. Pyysalo, T. Ohta, M. Miwa, H.-C. Cho, J. Tsujii, and S. Ananiadou, "Event extraction across multiple levels of biological organization," *Bioinformatics*, vol. 28, no. 18, pp. i575–i581, 2012. doi: 10.1093/bioinformatics/bts407

[12] S. Duan, R. He, and W. Zhao, "Exploiting document level information to improve event detection via recurrent neural networks," in *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2017, pp. 352–361.

[13] Y. Zhao, X. Jin, Y. Wang, and X. Cheng, "Document embedding enhanced event detection with hierarchical and supervised attention," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2018. doi: 10.18653/v1/P18-2066 pp. 414–419.

[14] T. H. Nguyen and R. Grishman, "Graph convolutional networks with argument-aware pooling for event detection," in *Thirty-second AAAI conference on artificial intelligence*, 2018. doi: 10.1609/aaai.v32i1.12039

[15] H. Yan, X. Jin, X. Meng, J. Guo, and X. Cheng, "Event detection with multi-order graph convolution and aggregated attention," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019. doi: 10.18653/v1/D19-1582 pp. 5766–5770.

[16] H.-L. Trieu, T. T. Tran, K. N. Duong, A. Nguyen, M. Miwa, and S. Ananiadou, "Deepeventmine: end-to-end neural nested event extraction from biomedical texts," *Bioinformatics*, vol. 36, no. 19, pp. 4910–4917, 2020. doi: 10.1093/bioinformatics/btaa540

[17] B. Portelli, E. Lenzi, E. Chersoni, G. Serra, and E. Santus, "Bert prescriptions to avoid unwanted headaches: A comparison of transformer architectures for adverse drug event detection," in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 2021. doi: 10.18653/v1/2021.eacl-main.149 pp. 1740–1747.

[18] A. Ramponi, R. van der Goot, R. Lombardo, and B. Plank, "Biomedical event extraction as sequence labeling," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020. doi: 10.18653/v1/2020.emnlp-main.431 pp. 5357–5367.

[19] Y. Chen, "A transfer learning model with multi-source domains for biomedical event trigger extraction," *BMC genomics*, vol. 22, no. 1, pp. 1–18, 2021. doi: 10.1186/s12864-020-07315-1

[20] Y. Gu, R. Tinn, H. Cheng, M. Lucas, N. Usuyama, X. Liu, T. Naumann, J. Gao, and H. Poon, "Domain-specific language model pretraining for biomedical natural language processing," *ACM Transactions on Computing for Healthcare (HEALTH)*, vol. 3, no. 1, pp. 1–23, 2021. doi: 10.1145/3458754

[21] S. Gururangan, A. Marasović, S. Swayamdipta, K. Lo, I. Beltagy, D. Downey, and N. A. Smith, "Don't stop pretraining: Adapt language models to domains and tasks," in *Proceedings of ACL*, 2020. doi: 10.18653/v1/2020.acl-main.740

[22] C. Nédellec, R. Bossy, J.-D. Kim, J.-J. Kim, T. Ohta, S. Pyysalo, and P. Zweigenbaum, "Overview of bionlp shared task 2013," in *Proceedings of the BioNLP shared task 2013 workshop*, 2013, pp. 1–7.

[23] T. Ohta, S. Pyysalo, and J. Tsujii, "Overview of the epigenetics and post-translational modifications (epi) task of bionlp shared task 2011," in *Proceedings of BioNLP Shared Task 2011 Workshop*, 2011, pp. 16–25.

[24] J.-D. Kim, Y. Wang, T. Takagi, and A. Yonezawa, "Overview of genia event task in bionlp shared task 2011," in *Proceedings of BioNLP shared task 2011 workshop*, 2011, pp. 7–15.

[25] J.-D. Kim, Y. Wang, and Y. Yasunori, "The genia event extraction shared task, 2013 edition-overview," in *Proceedings of the BioNLP Shared Task 2013 Workshop*, 2013, pp. 8–15.

[26] S. Pyysalo, T. Ohta, R. Rak, D. Sullivan, C. Mao, C. Wang, B. Sobral, J. Tsujii, and S. Ananiadou, "Overview of the infectious diseases (id) task of bionlp shared task 2011," in *Proceedings of BioNLP Shared Task 2011 Workshop*, 2011, pp. 26–35.

[27] A. G. Jivani *et al.*, "A comparative study of stemming algorithms," *Int. J. Comp. Tech. Appl*, vol. 2, no. 6, pp. 1930–1938, 2011.

[28] S. Petrov, D. Das, and R. McDonald, "A universal part-of-speech tagset," *arXiv preprint arXiv:1104.2086*, 2011.