Fine-tuning Pre-trained Transformer Language Models for Biomedical Event Trigger Detection

Laura Zanella, Yannick Toussaint

Fine-tuning Pre-trained Transformer Language Models for Biomedical Event Trigger Detection

Laura Zanella, Yannick Toussaint

LORIA (Universite de Lorraine, CNRS, Inria), Nancy, France
{laura-alejandra.zanella-calzada, yannick.toussaint}@loria.fr,
https://www.loria.fr/

Abstract. The detection of biomedical events plays an important role in building diverse applications in the biomedical domain, such as in disease prevention, pathway curation and epigenetics. Biomedical event trigger detection is a critical sub-task in extracting events that identify the possible occurrence of an event. Pre-trained transformer language models, such as BERT and its variants, have obtained the state-of-the-art performance in event extraction using different biomedical annotated corpus. However, a comparison between these models for this task has not yet been done. This paper proposes to analyze the differences between the performance of BERT and four of its variants tested on seven merged annotated biomedical corpus. BioBERT emerged as the best model from the evaluation presented here, showing that using a transformer model that is pre-trained from the original model, BERT, and uses biomedical data for its training is useful for recognizing biomedical event triggers, if the training is done for enough number of epochs.

1 Introduction

Biomedical event extraction is a complex information extraction task particularly dedicated to biomedical text, that plays a role of bridging the gap between Natural Language Processing (NLP) formulations and the expression of knowledge nuggets. The extraction of biomedical events allows to identify key information from large sets of textual data for further applications, such as pathway curation, study of biomolecular mechanisms of infectious diseases or epigenetic changes. A biomedical event contains an event trigger and one or more arguments. Event triggers generally refer to nouns or verbs that express a circumstance or eventuality, while arguments refer to biomedical entities or other events. If an event is part of the arguments of another event, then it is considered a nested event. As shown in Fig. 1, the example sentence contains an event of type 'Binding' that is constructed from the trigger word 'associations', and presents as argument a biomedical entity of the type 'Simple chemical', that plays the role of 'Theme' in the event. The sentence contains another event of the type 'Regulation', which

one argument is the biomedical entity of type 'Gene or gene product' with the role of 'Cause' and another argument is the nested event 'Binding', with the role 'Theme' [1].

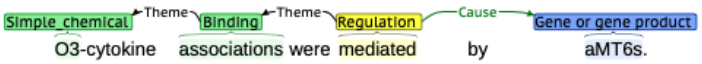

FIG. 1 – *Event extraction example: Binding event nested as argument to Regulation event.*

The extraction of biomedical events can be divided into two main sub-tasks. The first sub-task is event detection, which identifies and classifies the event triggers into a set of pre-defined types of events, and the second sub-task is argument identification, which identifies and classifies the corresponding event arguments with their respective roles (Shen et al., 2019). Event detection plays a critical role in building events, since the triggers are the targets that allow us to know that an event can exist (Cui et al., 2020). From previous works, it has been shown that more that 60 % of biomedical event extraction errors occur during the process of detecting triggers (Wang et al., 2016). This step is challenging as the same event can be represented in the form of different expressions, as single words or multi-words, and present non-conventional linguistic features, such as specialized language or consist of discontinuous spans of tokens. Also, they might be classified as different events in different contexts.

Event detection is usually considered as a multi-category classification problem. Neural network models have been widely adopted for solving this task, since they do not require the effort of experts for the design of features or use extra tools for their training. These models use word embeddings and language models as a distributed representation of the words, that transform the input text into a machine-readable language under a vectorized format. Language models are pre-trained in specific tasks using large datasets, providing the initial weights or checkpoints to the architecture of the neural model. Then, the neural model is trained in the new task by a process of fine-tuning, updating the checkpoints initially given to be able to fit a solution to the task.

Language models pre-trained on transformers architectures have become commonly used for solving different type of NLP tasks due to their positive achievements in performance. BERT (Devlin et al., 2018), which stands for Bidirectional Encoder Representations from Transformers, is a state-of-the-art (SOTA) language model designed to pre-train bidirectional representations of words, taking into account the context by considering both left and right directions of the text. From this pre-training, BERT can be fine-tuned by adding additional layers on top of the neural model to solve new tasks. Additionally, a series of variants from BERT have been developed for specific domains by being trained in large corpus with the same context, such as the biomedical domain.

The contribution of this work is the comparison of a set of transformer language models for detecting biomedical event triggers to analyze their performance and identify which is the most appropriate for tackling this task. For this purpose, seven annotated biomedical corpus are merged and used as input for the training of the models. The two focal points of this evaluation refer to (1) identify whether using a transformer model pre-trained on a biomedical domain

1. Sentence annotated with the BioNLP 2013 CG Corpus (Nédellec et al., 2013). Annotation visualization with the BRAT annotation tool: `https://brat.nlplab.org/`.

language presents advantages in the performance, and (2) analyze whether using the different biomedical corpus together for the models' training can improve the detection of event triggers.

Five transformers models are used for comparing their performance in detecting biomedical event triggers; BERT, BioBERT, SciBERT, PubMedBERT and BioMedRoBERTa. The model that achieved the top performance is BioBERT when it is trained for 100 epochs, showing that a model that was pre-trained using biomedical domain language data and initialized its pre-training from the BERT weights, is useful for identifying biomedical event triggers. Also, using different corpus as input data can improve the event detection for the trigger classes that overlap among the different corpus, since they provide enough samples to train the model. However, having a high number of samples does not ensure a significant classification performance, since the samples can present the same word triggers for different trigger types, which can be confusing for the model. A possible solution for this problem would be to include extra features for training the model to enrich the text context and better differentiate the triggers types.

2 Related Work

This section summarizes the techniques proposed from previous works for solving the event detection task.

The current SOTA systems for event detection use neural network models for their strong event extraction capabilities. In the model of (Wang et al., 2016) the information of dependency trees obtained from biomedical abstracts is used for training a model to acquire the word embeddings. The limitation of this system is the dependency on external tools of the dependency parsing, which can be a source of error propagation. Overcoming this limitation, (Jagannatha and Yu, 2016) explore two types of Bi-directional Recurrent Neural Networks (Bi-RNNs), Long Short Term Memory (LSTM) and Gated Recurrent Units (GRU), to extract trigger events from Electronic Health Records (EHR). Word embedding uses a skip-gram language model trained on biomedical data trough a shallow neural network. They compare the performance of their system and the one obtained by using CRF, showing an improvement in the recall. (Rahul et al., 2017) use RNNs to extract higher level features through the hidden state of the network. They also use the word and the entity type embeddings as features, without using any hand-design features, demonstrating that their system achieves the SOTA performance in the MLEE corpus.

Convolutional Neural Networks (CNNs) are another type of neural models that achieves high performance in detecting events. These models have shown good capacity for extracting features from the underlying structures of the k-grams in the sentences. One limitation of CNNs is that the modeling of k-grams is done in a consecutive way, ignoring the non-consecutive k-grams that can be important for event detection. In (Nguyen and Grishman, 2016) work, this problem is overcome by proposing the non-consecutive convolution, demonstrating the effectiveness of the general setting and the domain adaption by improving the performance of the SOTA works. The model uses pre-trained word embeddings for representing windows of text. As an extension of CNNs models, (Nguyen and Grishman, 2018) present a Graph Convolution Network (GCN) model to exploit syntactic dependency relations. They use dependency trees to link words to their informative context for event detection, demonstrating a performance that achieves the SOTA. (Yan et al., 2019) also propose a GCN model,

integrating aggregative attention to model and aggregate multi-order syntactic representations of the sentences, while in the case of (Cui et al., 2020), the extend the GCN by adding the relation aware concept, which exploits the syntactic relation labels and models the relation between words.

In addition to the current neural network models, pre-trained language models based on transformers are often involved in the detection of events since they have shown to improve the performance of the current systems. DeepEventMine (Trieu et al., 2020) 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, a linear layer is added in the neural model, having at the top the BERT model. One of the main objectives of this system is improving the extraction of nested events, where it was achieved the new SOTA performance on seven biomedical nested event extraction tasks. (Portelli et al., 2021) propose a comparison between transformers models, i.e. BERT and five of its variants, for the identification of Adverse Drugs and Events (ADEs). They show that span-based pre-training, from spanBERT, provides an improvement in the recognition of ADEs, and that the pre-training of the models in the specific domain is particularly useful in comparison to train the models from scratch. (Ramponi et al., 2020) developed BEESL, a neural network model based on 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. Finally, (Chen, 2021) propose 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.

Based on this analysis, neural networks models have the advantage of not requiring extra tools for extracting features or the need to hand-design features. According to the results obtained from these models, they have been positioned as the SOTA for extracting biomedical event triggers, where the use of pre-trained language models based on transformers architectures has shown an improvement in the performance of this task.

3 Methodology

This section briefly describes the biomedical event trigger detection task. Then, the proposed model, composed by the transformers models and the classification layer, and its training details, are introduced.

The aim of this proposal is to compare a set of pre-trained transformer language models to recognize and categorize biomedical event triggers. Event trigger detection is treated as a multi-classification problem, where for each word s_i in a sentence $S = s_1, s_2, ..., s_n$, the neural network model learns its vectorized representation, and predict its trigger class $l \in L$. Here, n refers to the number of words in the sentence and L to the collection of trigger types.

3.1 Model

The neural network model, used for training, consists of two main modules; a transformer model and a linear classification layer, as shown in Figure 2. A brief description of these

modules is presented below.

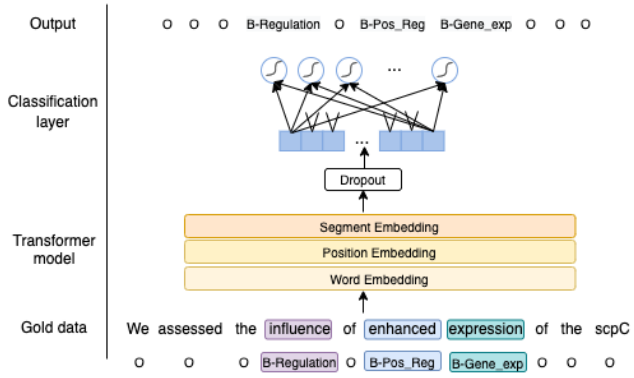

FIG. 2 – *Overview of the model used for detecting biomedical event triggers.*

3.1.1 Transformer Model

Most of SOTA transformer models follow the recent success of the BERT architecture. This model consists of a multi-layer, multi-head self-attention mechanism, which has shown performance advantages in leveraging GPU-based parallel computation and modeling long-path dependencies from the text. The sequence of input tokens (words or sub-words) is firstly processed using a lexical encoder. In this step are combined the token embedding, the (token) position embedding and the segment embedding (text segment to which the token corresponds) are combined through element-wise summation. Then, this embedding layer is 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 (Gu et al., 2021).

3.1.2 Linear classification Layer

After obtaining the contextual representation of the tokens, a linear classification layer is employed to classify the vectors into the event trigger classes. In this step, the high dimensional input vectors $x_1, x_2, ..., x_N \in X$ are submitted to a linear transformation $t_j = x_i w_j + b_j$, to be classified into one pre-defined category label $t_1, t_2, ..., t_M$ of the biomedical event triggers. Where M is the number of trigger categories, $w_j = (w_{j1}, w_{j2}, ..., w_{jD})$ represents the input weights that connect the input node and the jth node of the hidden layers, b_j represents the bias.

The output labels are calculated using the IOB (inside-outside-beginning) tagging for identifying the triggers and then, classifying them into the trigger types (in the case of the I and B tags). The final output is a sequence of IOB labels at the word-level. An example of this output is shown below, where words in bold in the sentence are the triggers identified and classified

in the trigger types presented in the labels below the words, while the rest of the words in the sentence are labeled as not being a trigger.

'PTHrP	**drives**	breast	tumor	**initial**	**progression**	and	**metastasis** ...'
'O'	*'B-Regulation'*	*'O'*	*'O'*	*'B-Development'*	*'I-Development'*	*'O'*	*'B-Metastasis'*

4 Experimental settings

This section introduces the corpus, experimental parameters setting and evaluation metrics used for comparing the pre-trained transformer models in the event detection task.

4.1 Corpus

The seven datasets used in this work for training the models are mentioned in Table 1, together with the number of triggers and events that they contain, the type of documents and the train, development and test dataset sizes (referring to the number of documents). A brief description of these datasets is included below. Cancer Genetics (CG) 2013 (Nédellec et al., 2013), contains physiological and pathological processes at various levels of biological organization. Epigenetics and Post-translational Modifications (EPI) 2011 (Ohta et al., 2011), contains the representations of proteins and DNA modification events and the catalysis of these reactions. GENIA 2011 (Kim et al., 2011), contains biomolecular events, as well as GENIA 2013 (Kim et al., 2013), but this last updated with more recent papers. Infectious Diseases (ID) 2011 (Pyysalo et al., 2011), contains biomolecular mechanisms of infectious diseases, virulence and resistance. Pathway Curation (PC) 2013 (Nédellec et al., 2013), contains targets reactions relevant to the development of biomolecular pathway models. Multi-Level Event Extraction (MLEE) (Pyysalo et al., 2012), contains events of different levels of biological organization ranging from the subcellular to the organism level.

Dataset	No. Triggers	No. Events	Documents	Train/Dev/Test
CG 2013	9,790	17,248	PubMed abstracts	300/100/200
EPI 2011	2,035	2,453	PubMed abstracts	600/200/400
GENIA 2011	10,210	13,560	MEDLINE abstracts	1,000 (total)
GENIA 2013	4,676	6,016	PMC full-text	34 (total)
ID 2011	2,155	2,779	PMC full-text	15/5/10
PC 2013	6,220	8,121	PubMed abstracts	260/90/175
MLEE	5,554	6,677	PubMed abstracts	131/44/87

TAB. 1 – *Statistics of the corpus used.*

The training and development datasets of all the corpus were initially merged into one single dataset and split into sentences, obtaining a total of 24,819 sentences. Then, a random data partition into 80/20 was applied for obtaining 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 following the setting of the BERT tokenization, which is a prerequisite for the input of BERT. The sentences split into sub-words are then given as input to the BERT layer.

All the different trigger types from each dataset were considered for the final trigger classification, presenting a final set of 58 trigger classes.

4.2 Pre-trained Transformer Models

Together with the original BERT model, four BERT variants were used for comparison, BioBERT (Lee et al., 2020), SciBERT (Beltagy et al., 2019), PubMedBERT (abstracts + full text) (Gu et al., 2020), and BioMedRoBERTa (Gururangan et al., 2020). These models differ from each other for the corpus they were trained on and their type of pre-training. Details about the models are presented in Table 2, where can be noticed that two of the BERT variants (SciBERT and PubMedBERT) were pre-trained from scratch, meaning that they use a unique vocabulary on their pre-training corpus and include embeddings that are specific for in-domain words. BioBERT and BioMedRoBERTa were pre-trained starting from the BERT checkpoints, which means that their vocabularies are built with general-domain texts (similar to BERT) as well as the initialization of the embeddings.

Model	Version	Pre-training	Corpus	Text size
BERT	base uncased	from scratch	WikiPedia + BookCorpus	3.3B words/16 GB
BioBERT	base v1.1	from BERT	PubMed	4.5B words
SciBERT	scivocab cased	from scratch	PMC + Semantic scholar	3.2B words
PubMedBERT	base uncased	from scratch	PMC + PubMed	3.1B words/21 GB
BioMedRoBERTa	base	from BERT	Semantic scholar	7.55B tokens/47GB

TAB. 2 – *Pre-trained language models based on transformers used for comparison.*

4.3 Parameter Settings

Experiments were developed with PyTorch and the models were taken from the Transformers repository [2]. All the transformer models use 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 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 vocab size varies for each model, where BERT presents 30,522; SciBERT, 31,116; BioBERT, 28,996; PubMedBERT, 30,522 and BioMedRoBERTa, 50,265.

For the rest of the training parameters, the batch size of both training and testing sets were set to 16, the learning rate to 1e-05 and the maximum gradient norm to 10, since gradient clipping was included. The maximum length of the sentences was set to 256.

All the models were trained for 10, 30 and 100 epochs on the training set.

4.4 Evaluation Metrics

Three metrics are measured for the evaluation of the experimental results; precision (P), recall (R) and F1-score (F1), which can be obtained from the equations in 1. TP refers to the true positives or the positive samples correctly classified. Positive samples refer to the samples that correspond to the specific class that is being evaluated, while the rest of the samples are

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

negative. FP refers to the false positives or the negative samples incorrectly classified and FN refers to the false negatives or the positive samples incorrectly classified.

$$P = \frac{TP}{TP + FP} \quad R = \frac{TP}{TP + FN} \quad F1 = \frac{2 * P * R}{P + R} \tag{1}$$

5 Evaluation

The results performed by each transformer model are shown in Table 3, were precision, recall and F1-score are given for the different number of epochs; 10, 30 and 100. All result values represent the training using the seven biomedical annotated datasets merged. BERT model achieves the top recall and F1-scores for 10 and 30 epochs, while SciBERT achieves the second best values, being lower than BERT only by 0.01 point in the F1-score. For both number of epochs, SciBERT presents higher precision than BERT, which suggests that from the triggers identified, SciBERT classifies more of them correctly, reducing the false positives. However, since BERT achieves to correctly classify more triggers from the total data as shown with the recall, it reduces the false negatives, which is more important for the model to avoid the problem of missing triggers.

When training the models for 100 epochs, the performance of BERT improves in a very subtle way even if the number of epochs is increased more than three times. BioBERT achieves the best scores in precision, recall and F1-score for this number of epochs, surpassing the rest of the models for at least 0.02, 0.05 and 0.03 points, for each metric respectively. BioBERT also presents the most important change in performance according to the number of epochs, improving its F1-score by around 0.03 points between 10 and 30 epochs, and 0.15 points between 30 and 100 epochs. This shows that the model improves its ability to classify biomedical triggers when trained for a longer period. In the case of SciBERT, a similar behavior is observed even if the performance improvement is less important, the F1-score improves by around 0.02 points between 10 and 30 epochs, and 0.07 points between 30 and 100 epochs. For the rest of the models, the change in performance is not as remarkable as in BioBERT and SciBERT, even if they present a slight improvement when increasing the number of epochs and, in the case of PubMedBERT, the score F1-score is decreased by 0.01 points between the training with 30 and 100 epochs. The results suggest that models pre-trained from BERT with a biomedical corpus, such as BioBERT and BioMedRoberta, are useful for detecting biomedical event triggers if the training is done for a sufficient number of epochs, as in the case of BioBERT trained for 100 epochs. On the contrary case, BioMedRoBERTa, only uses a general domain corpus for its pre-training and presents the lowest performance, even if the size of its pre-training corpus is larger than for the rest of the models.

On the other hand, using a model pre-trained from scratch with general and biomedical domain corpus combined, as in the case of SciBERT, presents better capabilities to identify biomedical triggers than using exclusively a biomedical corpus for the pre-training, as in the case of PubMedBERT. In the case of this last model, it also presents lower results than the models pre-trained from scratch using only a general domain corpus, as in the case of BERT, even if the size of the corpus in both models is similar.

Model	10 epochs			30 epochs			100 epochs		
	P	R	F1	P	R	F1	P	R	F1
BERT	**0.57**	**0.67**	**0.62**	**0.60**	**0.68**	**0.64**	0.62	0.68	0.65
BioBERT	0.51	0.61	0.55	0.57	0.59	0.58	**0.72**	**0.75**	**0.73**
SciBERT	0.59	0.64	0.61	0.61	0.65	0.63	0.70	0.70	0.70
PubMedBERT	0.49	0.61	0.54	0.58	0.66	0.61	0.58	0.62	0.60
BioMedRoBERTa	0.48	0.49	0.47	0.52	0.52	0.51	0.55	0.50	0.52

TAB. 3 – *Results of the pre-trained language models trained during 10, 30 and 100 epochs.*

5.1 Category grained performance analysis

The precision, recall and F1-score values for each of the trigger types are shown in Table 4, in descending order according to the F1-score value. The support or the number of occurrences of each trigger type in the data is also included. These results were obtained from BioBERT, trained for 100 epochs, since it presented the best performance from all the experiments.

From these results, we observe that the support of each type of trigger does not necessarily influence the ability of the model to classify the event triggers. Most of the trigger types with low support (≤ 5), have also a low or zero F1-score (last six trigger types in the lower right of the table). Trigger types with high support (≥ 100) do not exceed an F1-score of 0.78, except for $Deglycosylation$, $Process$ and $Gene\_expression$, which present F1-scores of 0.91, 0.90 and 0.85, respectively. The event trigger type with the highest support is $Positive\_regulation$, with a F1-score of 0.70, which is significantly lower in comparison to the two trigger types with the highest F1-score, $Amino\_acid\_catabolism$ and $Glycolysis$, that obtained F1-score of 1.00 and 0.95, respectively, even if they presented a support of 1 and 10. Analyzing the corpus, the trigger word for the $Amino\_acid\_catabolism$ type, is always 'glutaminolysis' in the different sentences of the training and testing sets, which facilitates its detection. For the $Glycolysis$ type, the situation is similar, having always as trigger the word itself included: 'glycolysis', 'glycolysis pathway', 'aerobic glycolysis', or a variant of the word: 'glycolytic'.

Similar to $Positive\_regulation$, $Negative\_regulation$ and $Regulation$ are two of the trigger types that have high support (586 and 556, respectively) but relatively lower F1-score (0.75 and 0.61, respectively). From this, it can be observed that even having large number of occurrences, the model presents problems for its classification. This may be because the triggers that correspond to these categories are usually similar words that are modified by a negation, in the case of $Negative\_regulation$, or that depend on the context to know if they belong to $Regulation$ or $Positive\_regulation$, which may be difficult for the model to identify.

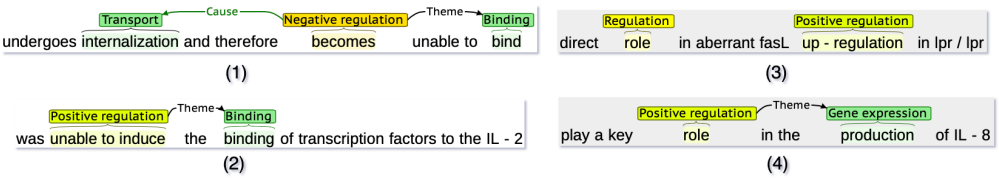

FIG. 3 – *Examples of sentences with annotated event triggers.*

Figure 3 shows in sentence (1) an example of a $Negative\_regulation$ trigger, where the trigger 'becomes' was incorrectly classified by the model as $Regulation$. The word that provides the information about a negation in the sentence is 'unable'; however, it is not annotated as part of the trigger, which can provide confusion to the model to differentiate from other type of regulation trigger. On the other hand, in sentence (2), the $Positive\_regulation$ trigger 'unable to induce' was incorrectly classified as $Negative\_regulation$. This can be due to the fact that the word 'unable' is part of the trigger, and it represents a negative action, which can also creates confusion to the model. A possible solution to this problem is including a module for negation detection, in order to identify negations even if they are not part of the annotated triggers.

In sentence (3) is shown an example of ambiguous information, with the $Regulation$ trigger and, in sentence (4) with the $Positive\_regulation$ trigger, where for both cases the trigger is 'role'. This can create confusion to the model, as the same word is annotated with a different trigger type. Using extra-features to enrich the data given to the model, such as the parts-of-speech (POS) tags, can be a possible solution to this problem, since POS information has demonstrated to be helpful in detecting event triggers (Shen et al., 2019).

Trigger type	P	R	F1	Support	Trigger type	P	R	F1	Support
Amino_acid_catabolism	1.00	1.00	1.00	1	Entity	0.63	0.74	0.68	398
Glycolysis	1.00	0.90	0.95	10	Degradation	0.68	0.68	0.68	19
Acetylation	0.86	0.99	0.92	82	Transcription	0.65	0.70	0.67	175
Phosphorylation	0.89	0.94	0.91	207	Synthesis	1.00	0.50	0.67	2
Deglycosylation	0.83	1.00	0.91	5	Conversion	0.55	0.75	0.64	28
Process	0.84	0.96	0.90	136	Regulation	0.66	0.57	0.61	556
Deacetylation	0.81	1.00	0.90	13	Blood_vessel_development	0.52	0.72	0.60	18
Metastasis	0.84	0.92	0.88	53	Transport	0.62	0.54	0.59	42
Methylation	0.85	0.90	0.87	73	Planned_process	0.65	0.54	0.59	104
Demethylation	0.75	1.00	0.86	3	Metabolism	0.57	0.57	0.57	7
Ubiquitination	0.82	0.90	0.86	67	Cell_death	0.56	0.58	0.57	43
Gene_expression	0.82	0.88	0.85	754	Growth	0.50	0.67	0.57	3
Hydroxylation	0.82	0.85	0.84	27	DNA_demethylation	0.40	1.00	0.57	2
Glycosylation	0.81	0.84	0.82	67	DNA_domain_or_region	0.57	0.57	0.57	7
DNA_methylation	0.82	0.82	0.82	77	Development	0.49	0.54	0.51	39
Cell_differentiation	0.92	0.73	0.81	15	Dephosphorylation	0.33	1.00	0.50	1
Carcinogenesis	0.78	0.81	0.79	31	Deubiquitination	1.00	0.33	0.50	3
Activation	0.78	0.80	0.79	65	Inactivation	0.44	0.53	0.48	15
Protein_catabolism	0.70	0.87	0.78	30	Catalysis	0.38	0.56	0.45	16
Pathway	0.79	0.76	0.78	168	Breakdown	0.40	0.50	0.44	4
Cell_proliferation	0.77	0.73	0.75	37	Mutation	0.45	0.41	0.43	32
Binding	0.72	0.79	0.75	434	Protein_processing	0.25	1.00	0.40	1
Negative_regulation	0.71	0.79	0.75	586	Anaphora	0.23	0.14	0.18	49
Localization	0.71	0.77	0.74	164	Protein_domain_or_region	0.00	0.00	0.00	5
Infection	1.00	0.56	0.71	9	Cell_division	0.00	0.00	0.00	2
Cell_transformation	0.76	0.67	0.71	39	Catabolism	0.00	0.00	0.00	5
Positive_regulation	0.72	0.68	0.70	1,276	Remodeling	0.00	0.00	0.00	1
Dissociation	0.64	0.78	0.70	9	Translation	0.00	0.00	0.00	2
Death	0.69	0.69	0.69	16	Dehydroxylation	0.00	0.00	0.00	1

TAB. 4 – *Results of trigger categories on the test set from BioBERT trained with 100 epochs.*

6 Conclusion

In this work is presented a comparison between five pre-trained transformer models in order to identify the one that performs the best for biomedical event trigger detection. For this purpose, all models are trained using a corpus with seven merged biomedical datasets and, an analysis of the models, including the type of pre-training and the pre-training corpus is devel-

oped according to the performance obtained. One of the main goals was to identify whether using a transformer model pre-trained on a biomedical domain language presents advantages in the performance. From the results of the different models, BioBERT presented the highest performance when it is trained for 100 epochs. This model is pre-trained from BERT using a biomedical corpus, suggesting that a model pre-trained on in-domain data that does not start its pretraining from scratch is the best strategy for biomedical event trigger detection. When analyzing whether the use of the different biomedical corpus merged can improve the detection of event triggers, it is observed that some of the types of triggers that present a very small support have a low performance, since there are not enough samples for the model to learn. However, the types of triggers that present high support do not necessarily present high performance, suggesting that these types of triggers may present ambiguities between their samples, making it difficult for the model to generalize. Based of this, for the next step a possible direction would be to enrich the information given to the model by adding extra features, as the POS, in order to reduce ambiguities, and to merge the trigger types with the lowest support to other categories with similar types of events to solve the data imbalance problem.

References

Beltagy, I., K. Lo, and A. Cohan (2019). Scibert: A pretrained language model for scientific text. *arXiv preprint arXiv:1903.10676*.

Chen, Y. (2021). A transfer learning model with multi-source domains for biomedical event trigger extraction. *BMC genomics 22*(1), 1–18.

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

Devlin, J., M.-W. Chang, K. Lee, and K. Toutanova (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Gu, Y., R. Tinn, H. Cheng, M. Lucas, N. Usuyama, X. Liu, T. Naumann, J. Gao, and H. Poon (2020). Domain-specific language model pretraining for biomedical natural language processing.

Gu, Y., R. Tinn, H. Cheng, M. Lucas, N. Usuyama, X. Liu, T. Naumann, J. Gao, and H. Poon (2021). Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH) 3*(1), 1–23.

Gururangan, S., A. MarasoviÄ, S. Swayamdipta, K. Lo, I. Beltagy, D. Downey, and N. A. Smith (2020). Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of ACL*.

Jagannatha, A. N. and H. Yu (2016). Bidirectional rnn for medical event detection in electronic health records. In *Proceedings of the conference. Association for Computational Linguistics. North American Chapter. Meeting*, Volume 2016, pp. 473. NIH Public Access.

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

Kim, J.-D., Y. Wang, and Y. Yasunori (2013). The genia event extraction shared task, 2013 edition-overview. In *Proceedings of the BioNLP Shared Task 2013 Workshop*, pp. 8–15.

Lee, J., W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang (2020). Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics 36*(4), 1234–1240.

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

Nguyen, T. H. and R. Grishman (2016). Modeling skip-grams for event detection with convolutional neural networks. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 886–891.

Nguyen, T. H. and R. Grishman (2018). Graph convolutional networks with argument-aware pooling for event detection. In *Thirty-second AAAI conference on artificial intelligence*.

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

Portelli, B., E. Lenzi, E. Chersoni, G. Serra, and E. Santus (2021). 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*, pp. 1740–1747.

Pyysalo, S., T. Ohta, M. Miwa, H.-C. Cho, J. Tsujii, and S. Ananiadou (2012). Event extraction across multiple levels of biological organization. *Bioinformatics 28*(18), i575–i581.

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

Rahul, P. V., S. K. Sahu, and A. Anand (2017). Biomedical event trigger identification using bidirectional recurrent neural network based models. *arXiv preprint arXiv:1705.09516*.

Ramponi, A., R. van der Goot, R. Lombardo, and B. Plank (2020). Biomedical event extraction as sequence labeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 5357–5367.

Shen, C., H. Lin, X. Fan, Y. Chu, Z. Yang, J. Wang, and S. Zhang (2019). Biomedical event trigger detection with convolutional highway neural network and extreme learning machine. *Applied Soft Computing 84*, 105661.

Trieu, H.-L., T. T. Tran, K. N. Duong, A. Nguyen, M. Miwa, and S. Ananiadou (2020). Deepeventmine: end-to-end neural nested event extraction from biomedical texts. *Bioinformatics 36*(19), 4910–4917.

Wang, J., J. Zhang, Y. An, H. Lin, Z. Yang, Y. Zhang, and Y. Sun (2016). Biomedical event trigger detection by dependency-based word embedding. *BMC medical genomics 9*(2), 123–133.

Yan, H., X. Jin, X. Meng, J. Guo, and X. Cheng (2019). 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)*, pp. 5766–5770.