BOUN-REX at CLEF-2020 ChEMU Task 2: Evaluating Pretrained Transformers for Event Extraction

Hilal Dönmez[1]⋆, Abdullatif Köksal[1]⋆, Elif Ozkirimli[2,3], and Arzucan Özgür[1]

[1] Department of Computer Engineering, Boğaziçi University, Turkey
[2] Department of Chemical Engineering, Boğaziçi University, Turkey
[3] Pharma International Informatics Data and Analytics Chapter, F. Hoffmann-La Roche AG, Switzerland
{hilal.donmez,abdullatif.koksal,elif.ozkirimli,arzucan.ozgur}@boun.edu.tr

Abstract. In this paper, we describe our models and results designated for CLEF-2020 ChEMU Task 2 [4], event extraction for chemical patent documents. We make use of the recent advances in pretrained transformer architectures such as BERT and BioBERT. We compare several transformers with different settings in order to improve performance. Our best performing model with BioBERT transformer architecture and AdamW optimizer achieves 0.7234 exact F1 score on the test dataset.

1 Introduction

Chemical information in patents is an essential resource for researchers working on chemical exploration and reactions. As the number of patents grows rapidly, Natural Language Processing (NLP) approaches are widely used to extract chemical information from patents so as to reduce the time and effort spent. Most previous studies on chemical information extraction focus on chemical named entity recognition (NER) [6] thanks to publicly available annotated corpora. On the other hand, there is a limited number of studies on chemical event extraction from patents.

Event extraction from patents contains detection of event trigger word, event trigger type, and event type. Figure 1 illustrates an example sentence of the event extraction task in the dataset released by Cheminformatics Elsevier Melbourne University (ChEMU). In this example, *room temperature* and *30 minutes* are given as entities with their corresponding types: *TEMPERATURE* and *TIME*. After *stirred* is detected as a trigger word for both entities, two event types (both of type ARGM) are determined separately according to the relevant entity type.

⋆ Equal contributions.

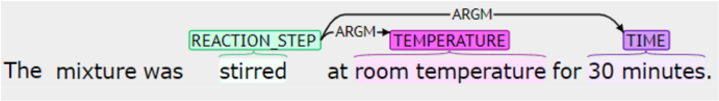

Fig. 1. An event extraction example from the ChEMU Dataset

In this work, we investigated the impact of various transformer architectures with different parameters on event extraction from patents by conducting several experiments. We also explored the effects of the pretraining corpus of transformers by comparing BERT [2] and BioBERT [7]. Besides, we investigated the significance of different optimizers such as Adam, AdamW, and SGD for the finetuning of transformers for this task.

2 Related Work

Determining the semantic relation between entities is an important scientific problem in various domains such as biomedical text, digital text, and governmental documents. Recently, deep neural networks have been widely used to identify the relations between entities. Previous research studies that use deep learning for relation extraction make use of CNN [17] and RNN [20] models by taking sentence representations with word vectors such as Word2Vec [11] and GloVe [13] in order to extract features automatically instead of hand-crafted features [5]. Recent studies on relation extraction have been based on the transformer architecture [15] trained on large amounts of unlabeled data to improve the state-of-the-art on several natural language processing tasks. In [19], a pretrained transformer model is utilized to extract efficient relation representations from text.

In event extraction, earlier neural network models enhanced CNN [12] and RNN [18] with different kinds of word representations to determine the locations and types of trigger words. In addition, structured information benefiting from dependency trees [8] and knowledge bases [9] is exploited by neural networks to improve event extraction performance. Lately, pretrained transformer based models have gained popularity for event extraction. In [16], trigger and argument extractor models obtain feature representations using BERT, a pretrained transformer model.

3 Methodology and Data

3.1 Data

We use the dataset released for the ChEMU tasks on information extraction from chemical patents. The dataset contains chemical patent documents with annotation files for training, development, and test sets. Entities with their types and relations between these entities are included in the annotation files. There are 10 different types of entity annotations for the Event Extraction Task in ChEMU. Table 1 shows the annotated types of entities in the dataset.

Table 1. Annotated Entity Types in Chemical Patents

Entity Types	
REACTION_PRODUCT	STARTING_MATERIAL
REAGENT_CATALYST	SOLVENT
OTHER_COMPOUND	EXAMPLE_LABEL
TEMPERATURE	TIME
YIELD_PERCENT	YIELD_OTHER

The event extraction problem focuses on event trigger word detection, trigger type detection, and event type prediction. Event trigger words whose types are REACTION_STEP or WORKUP are identified and the chemical entity arguments of the events are determined. The relation between an argument and a trigger word is labeled as a semantic argument role label, which is Arg1 or ArgM. The relation between a trigger word and a temperature, time or yield entity is labeled as ArgM, whereas the relation between a trigger word and an entity having one of the other entity types is labeled as Arg1. Table 2 contains the statistics of the ChEMU Dataset.

Table 2. Statistics of the ChEMU Dataset. Chemical patent documents are split into sentences via GENIA Sentence Splitter [13]. The gold standard labels for the test set are not available at this time. Therefore, the corresponding entries in the table are marked with '-'.

	Train Set	**Development Set**	**Test Set**
# of documents	900[4]	225	9999
# of entities	16343	3843	4575980
# of trigger words	6867	1605	-
# of relations	14310	3332	-
# of Arg1 relations	9703	2247	-
# of ArgM relations	4607	1085	-
# of sentences	5974	1418	3942870

[4] We were able to use 713 out of the 900 documents in the train set due to a problem during the downloading process.

3.2 Preprocessing

Our preprocessing steps involve sentence splitting and adding entity markers. For simplicity, we consider the relations that are present in single sentences, and we split the documents into sentences via the GENIA Sentence Splitter [14]. For each entity in a sentence, we construct sentence-entity pairs and predict events and trigger words from these pairs. On the other hand, there are 121 entities that have relations with more than one trigger word in our training set. We ignore these kinds of entities for event trigger word detection.

We need to explicitly identify an entity to find the corresponding relation and trigger word in a sentence. Therefore, we add specific markers called <E> and </E> before and after the entities for the model to identify the entities by following the discussion in [1]. Moreover, we create different representations for each sentence having more than one entity by applying the marker method. Hence, the sentence representation is distinct for each entity in the same sentence having more than one entities. The following examples show that there are two different representations for *hexanes* and *silica*, which are located in the same sentence.

- The solvent was removed in vacuo, and the crude product was purified by flash chromatography (silica, 100% **<E> hexanes </E>** to 9:1 hexanes/EtOAc) to give a pale-yellow viscous oil (3.83 g, 86%).
- The solvent was removed in vacuo, and the crude product was purified by flash chromatography (**<E> silica </E>**, 100% hexanes to 9:1 hexanes/EtOAc) to give a pale-yellow viscous oil (3.83 g, 86%).

3.3 Model

Problem Definition: For a given sentence S with an entity e_t with type t, the objectives are to find the trigger word in S, its type including *None*, and the relation between the trigger word and e_t from a set of predefined event types. As event types are determined according to entity types, we do not make a model for event type detection. Hence, we focus on trigger type and trigger word detection. If there is a trigger word for an entity in a given sentence, the event type is found by simple rules.

Table 3. Lookup table for event types

Entity Types	**Event Type**
REACTION_PRODUCT STARTING_MATERIAL REAGENT_CATALYST SOLVENT OTHER_COMPOUND EXAMPLE_LABEL	Arg1
TEMPERATURE TIME YIELD_PERCENT YIELD_OTHER	ArgM

Two objectives are selected to address this problem. Our base model is a transformer-based pretrained architecture, which extracts a fixed-length sentence representation and token representations from an input sentence with entity markers indicating the entity's location. The fixed-length sentence representation is utilized to detect the type of the trigger word in the sentence with a given annotated entity. If there is a trigger word in the sentence, the event type is determined by the type of the given entity from a simple lookup table shown in Table 3.

We propose an approach similar to question answering methods [2] to find the span of the trigger word. Our trigger word span model predicts probabilities of *start* and *end* tags with the token representations which are produced by the transformer-based pretrained architecture. Trigger word span is the sequence between tokens with the highest start and end probabilities.

Our proposed architecture is jointly trained, as shown in Figure 2. Different pretrained transformers with several optimizers, learning rates, and weight decays are evaluated on the development set by exact F1 scores. The considered settings are summarized below. The configuration for our best model is shown in bold.

- Transformer Architectures: **BioBERT**[5], $\text{BERT}_{\text{Large}}$[6], $\text{BioBERT}_{\text{Large}}$[7]
- Optimizer: **AdamW**, Adam, SGD
- Learning Rate: $\mathbf{1e-5}$, $1e-6$, $1e-4$, $1e-3$
- Weight Decay: **0**, 0.1, 0.01

[5] https://huggingface.co/monologg/biobert_v1.1_pubmed

[6] https://github.com/google-research/bert

[7] https://huggingface.co/trisongz/biobert_large_cased

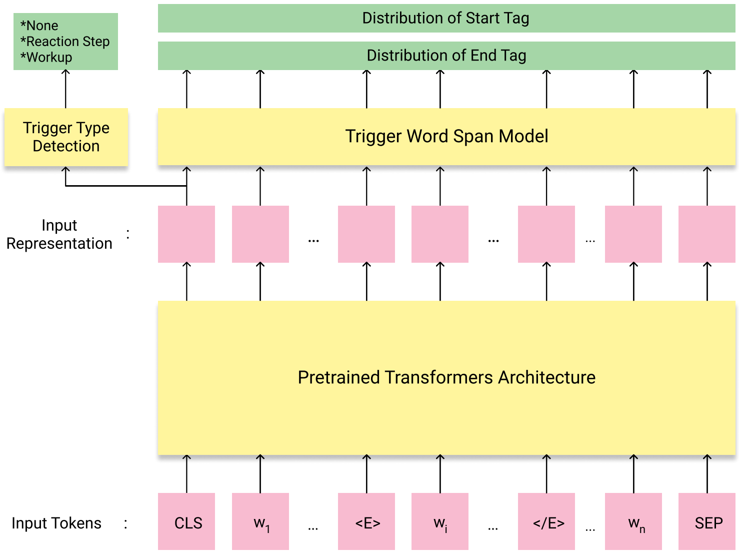

Fig. 2. Our final model with trigger type detection and trigger word span model. w_i's represent wordpieces constructed from the tokenizer of the pretrained transformer. *CLS* and *SEP* are special tokens used as a fixed-length sentence representation and as a separator, respectively. Trigger type detection is used to classify the type of the trigger word. Trigger word span model finds the span of the trigger word in the sentence.

4 Results

We compare our results with pretrained transformer architectures with different settings. We evaluate the final performance of our model with exact F1 and relax F1 scores given in [4]. Furthermore, the F1 score for trigger type detection and the accuracy for trigger word span detection are also presented. We report our results by taking the average scores of 10 runs to decrease the effect of high variance in transformers architectures, as stated in [3].

Table 4. Exact F1 scores of pretrained transformer architectures with different optimizers on the development set.

Optimizer	**BioBERT**	**BioBERT$_{\textbf{Large}}$**	**BERT$_{\textbf{Large}}$**
AdamW	**0.7367**	0.7332	0.7329
Adam	0.7351	0.7227	0.7042
SGD	0.7292	0.7177	0.6755

As shown in Table 4, the best performing pretrained transformer model is BioBERT with AdamW optimizer, even though the complexities of $\text{BERT}_{\text{Large}}$ and $\text{BioBERT}_{\text{Large}}$ are higher than BioBERT. $\text{BERT}_{\text{Large}}$ and $\text{BioBERT}_{\text{Large}}$ have 24 layers, 16 heads, and 340 million parameters while BioBERT has 12 layers, 12 heads, and 110 million parameters. Besides, while $\text{BERT}_{\text{Large}}$ is pretrained on English Wikipedia and Book Corpus, BioBERT and $\text{BioBERT}_{\text{Large}}$ are pretrained on additional resources, i.e., Pubmed Abstracts and PMC full-text articles. Table 4 shows that BioBERT and $\text{BioBERT}_{\text{Large}}$ perform better than $\text{BERT}_{\text{Large}}$. Our results suggest that the domain similarity between chemical patent documents and the pretraining corpus of BioBERT and $\text{BioBERT}_{\text{Large}}$ leads to better performance. In [10], it is shown that the generalization capability of the AdamW optimizer is better than the Adam and SGD optimizers and our results support this claim.

Table 5. *F1 score* for trigger type detection and *accuracy* for trigger word span detection on the development set.

Module	**Class**	**BioBERT+AdamW**
Trigger Type Detection	All	0.9848
	None	0.9735
	Reaction Step	0.9885
	Workup	0.9822
Trigger Word Span	Both	0.9524
	Start	0.9591
	End	0.9567

There are two different objectives, namely trigger type detection and trigger word span detection, in our final architecture. Table 5 contains the results of the two objectives separately on the development set. The trigger type detection model achieves 0.9848 F1 score, whereas the accuracy of our trigger word span model is 0.9524.

Table 6. Our final model's precision, recall, and F1 scores on the development and test sets.

	Precision		**Recall**		**F1**	
	Exact	Relax	Exact	Relax	Exact	Relax
Development Set	0.7690	0.7700	0.7069	0.7072	0.7367	0.7372
Test Set	0.7610	0.7610	0.6893	0.6893	0.7234	0.7234

Our final model's performance is summarized in Table 6 for all objectives: trigger word, trigger type and event type detections. It achieves 0.7407 and

0.7234 in the main metric (exact F1) on the development and test sets, consecutively.

5 Conclusion and Future Work

In this paper, we introduce a transformer based approach for event extraction in chemical patent documents. We compare several pretrained transformer models with different settings and show that BioBERT's performance with the AdamW optimizer is better than both $\text{BERT}_{\text{Large}}$ and $\text{BioBERT}_{\text{Large}}$ for this task. Finally, we report our best model's performance separately on the trigger type and trigger word span detection tasks. Our best model, BioBERT, achieves 0.7234 exact F1 score on the test set.

As future work, we plan to extend our study to enable the detection of multiple trigger words in a sentence by using a sequence labeling setup with the BIO encoding. Thus, we will consider entities having relations with more than one trigger word. In addition, we will design a two-stage model that firstly detects the trigger word span and then classifies the trigger type as an alternative to our jointly trained model.

6 Acknowledgements

GEBIP Award of the Turkish Academy of Sciences (to A.O.) is gratefully acknowledged.

References

1. Baldini Soares, L., FitzGerald, N., Ling, J., Kwiatkowski, T.: Matching the blanks: Distributional similarity for relation learning. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. pp. 2895–2905. Association for Computational Linguistics, Florence, Italy (Jul 2019). https://doi.org/10.18653/v1/P19-1279, https://www.aclweb.org/anthology/P19-1279
2. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: 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). pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019). https://doi.org/10.18653/v1/N19-1423, https://www.aclweb.org/anthology/N19-1423
3. Dodge, J., Ilharco, G., Schwartz, R., Farhadi, A., Hajishirzi, H., Smith, N.: Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. arXiv preprint arXiv:2002.06305 (2020)

4. He, J., Nguyen, D.Q., Akhondi, S.A., Druckenbrodt, C., Thorne, C., Hoessel, R., Afzal, Z., Zhai, Z., Fang, B., Yoshikawa, H., Albahem, A., Cavedon, L., Cohn, T., Baldwin, T., Verspoor, K.: Overview of chemu 2020: Named entity recognition and event extraction of chemical reactions from patents. In: Arampatzis, A., Kanoulas, E., Tsikrika, T., Vrochidis, S., Joho, H., Lioma, C., Eickhoff, C., Névéol, A., Cappellato, L., Ferro, N. (eds.) Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Eleventh International Conference of the CLEF Association (CLEF 2020), vol. 12260. Lecture Notes in Computer Science (2020)
5. Kambhatla, N.: Combining lexical, syntactic, and semantic features with maximum entropy models for information extraction. In: Proceedings of the ACL Interactive Poster and Demonstration Sessions. pp. 178–181 (2004)
6. Krallinger, M., Rabal, O., Leitner, F., Vazquez, M., Salgado, D., Lu, Z., Leaman, R., Lu, Y., Ji, D., Lowe, D.M., et al.: The chemdner corpus of chemicals and drugs and its annotation principles. Journal of cheminformatics **7**(1), 1–17 (2015)
7. Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C.H., Kang, J.: Biobert: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics **36**(4), 1234–1240 (2020)
8. Li, D., Huang, L., Ji, H., Han, J.: Biomedical event extraction based on knowledge-driven tree-lstm. 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). pp. 1421–1430 (2019)
9. Liu, S., Chen, Y., He, S., Liu, K., Zhao, J.: Leveraging framenet to improve automatic event detection. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 2134–2143 (2016)
10. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: International Conference on Learning Representations (2018)
11. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
12. Nguyen, T.H., Grishman, R.: Event detection and domain adaptation with convolutional neural networks. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers). pp. 365–371 (2015)
13. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Empirical Methods in Natural Language Processing (EMNLP). pp. 1532–1543 (2014), `http://www.aclweb.org/anthology/D14-1162`
14. Sætre, R., Yoshida, K., Yakushiji, A., Miyao, Y., Matsubayashi, Y., Ohta, T.: Akane system: protein-protein interaction pairs in biocreative2 challenge, ppi-ips subtask. In: Proceedings of the second biocreative challenge workshop. vol. 209, p. 212. Madrid (2007)
15. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. ArXiv **abs/1706.03762** (2017)
16. Yang, S., Feng, D., Qiao, L., Kan, Z., Li, D.: Exploring pre-trained language models for event extraction and generation. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. pp. 5284–5294 (2019)
17. Zeng, D., Liu, K., Lai, S., Zhou, G., Zhao, J.: Relation classification via convolutional deep neural network. In: Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers. pp. 2335–2344 (2014)

18. Zhang, W., Ding, X., Liu, T.: Learning target-dependent sentence representations for chinese event detection. In: China Conference on Information Retrieval. pp. 251–262. Springer (2018)
19. Zhao, Y., Wan, H., Gao, J., Lin, Y.: Improving relation classification by entity pair graph. In: Asian Conference on Machine Learning. pp. 1156–1171 (2019)
20. Zhou, P., Shi, W., Tian, J., Qi, Z., Li, B., Hao, H., Xu, B.: Attention-based bidirectional long short-term memory networks for relation classification. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). pp. 207–212 (2016)