Exploring BERT Architectures for Relation Extraction of Chemical Patents

Lea Cleary

E-mail: lcleary@berkeley.edu

Abstract

Novel chemistries developed outside of academia are often only published in patents, but patents are written for the purpose of protecting intellectual property, so they can be challenging to parse for relevant information. This project focuses on the relation extraction subtask of extracting the synthesis process of new chemical compounds from patent snippets. It explores different combinations of input and output representations for BERT models in order to understand which architectures are most effective for classifying relations from chemical patents.

1. Introduction

Chemical patent literature is a vital source of information about new chemical compounds, especially in industries like drug discovery. Novel chemistries developed outside of academia are often published first or only in patents. Automated information extraction from chemical patents is becoming more necessary since the ever-increasing volume of existing patents is no longer tractable to process manually.

While text mining techniques have been developed for scientific literature and clinical texts, these techniques are not straightforward to transfer to chemical patent literature. Patents are written for the purpose of protecting intellectual property and contain different scope and language from scientific literature. Thus, information extraction techniques need to be developed with the specific focus of chemical patent literature in mind.

2. Background

The Cheminformatics Elsevier Melbourne University (ChEMU) evaluation lab was introduced in 2020 as a series of challenges aimed at developing information extraction tools for chemical patents. ChEMU 2020 is the first running of the series which focused on extracting the synthesis process of new chemical compounds from patent snippets (He et al., 2020). This first challenge consisted of two key information extraction tasks:

1) Chemical named entity recognition (NER), which aims to identify chemical compounds and their specific roles in a reaction, and
2) Event extraction, which identifies the reaction steps and relates the compounds involved in the chemical reaction.

This project presents a modified version of the second task. Participants of the challenge normally break apart the second task into two steps: identification of the trigger word indicating a reaction step, followed by relation extraction. Since trigger word identification can be considered an extension of the chemical NER task, this project focuses solely on on relation extraction and assumes that all the entities, including the trigger words, are given at the beginning of the task. This project will also explore the application of BERT transformer models to complete the task.

Five groups of participants tackled the event extraction task in ChEMU 2020, two of which used BERT-based methods. BOUN-REX (Köksal, et al., 2020) posted a best model F1-score of 0.7234. Their approach was to split each patent snippet into individual sentences, mark the entities from the NER step, then fine-tuned BioBERT with a combined trigger word detection and relation extraction loss function to complete the event extraction task. Melaxtech (Wang et al., 2020) posted the best F1-score of the competition with a 0.9548. Their approach also used single sentences, but they replaced the entity with its semantic type, and fine-tuned a BioBERT model they pre-trained on patent

[Step 3] Synthesis of N-(3-chloro-4-fluorophenyl)-N-(2-fluoro-4-(hydrazinecarbonyl)benzyl)tetrahydro-2H-thiopyran-4-carboxamide 1,1-dioxide
Methyl 4-((N-(3-chloro-4-fluorophenyl)-1,1-dioxidotetrahydro-2H-thiopyran-4-carboxamido)methyl)-3-fluorobenzoate (1.240 g, 2.628 mmol), synthesized in step 2, and hydrazine monohydrate (2.554 mL, 52.554 mmol) were dissolved in ethanol (20 mL)/water (5 mL) at room temperature, and the solution was stirred at 80°C for 5 hours, and then cooled to room temperature to terminate the reaction. The reaction mixture was concentrated under reduced pressure to remove the solvent. The title compound was used without further purification (1.180 g, 95.2%) as yellow solid.

Figure 1 Example of one patent snippet in the ChEMU 2020 corpus

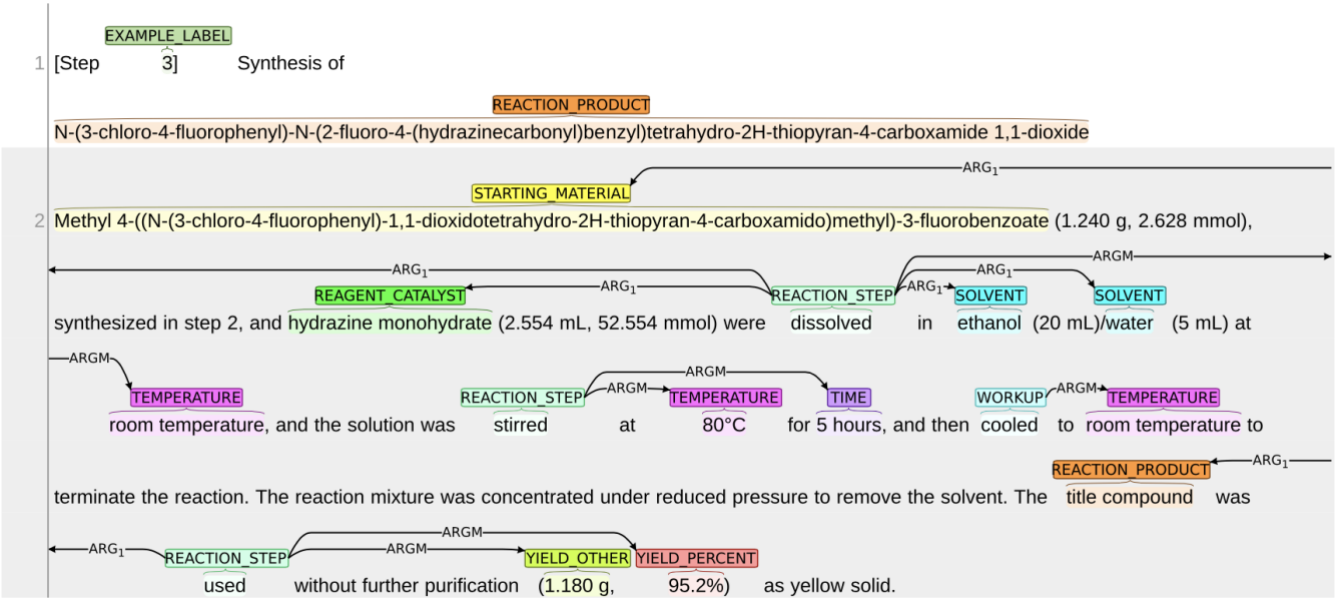

Figure 2 Visualization of the annotations in the snippet in **Figure 1**.

literature, which they called Patent_BioBERT. Afterward, they applied some rule-based post-processing to retrieve any relations they missed from the single-sentence approach.

The approach for this project is to explore different combinations of input and output representations for BERT models in order to understand which architectures are most effective for classifying relations from chemical patents.

3. Data

The ChEMU 2020 corpus provided for the challenge is a benchmark annotated dataset consisting of 1500 patent snippets sampled from 170 patent documents from both the US and European patent offices (Vespoor et al., 2020). An example of a patent snippet and its annotations are given in **Figure 1** and **Figure 2**, respectively. The corpus is provided pre-split into train, dev, and test sets with 900, 225, and 375 snippets, respectively. There is also a sample dataset consisting of only 50 snippets.

For the event extraction task, two types of trigger words indicate a relation: WORKUP, which is an event step where a chemical compound is isolated or purified; and REACTION_STEP, which is an event step involved in converting starting material to product. The relation extraction task consists of two labels: ARG1, which is a relation between an event trigger word and a chemical compound; and ARGM, which is a relation between a trigger word and conditions or yields for a reaction.

3.1 Data Processing

The ChEMU 2020 corpus is a multi-relation extraction (MRE) problem. MRE with neural network architectures is problematic because neural networks require pre-defined sizes for inputs, but the number of relations per document in MRE can vary. A viable option for using BERT architecture on the ChEMU 2020 corpus is to run a multi-pass single relation extraction (SRE) approach instead. This would require generating a separate input for every relation in each snippet.

Patent snippets can be longer than the BERT base token limit of 512, and some are even longer than the BERT large token limit of 1024. This is the prime motivation for splitting patent snippets into individual sentences. However, relations can span across multiple sentences, so splitting by individual sentences alone will result in missing relations. Thus, instead of splitting the patents into single sentences, the data pre-processing keeps just the sentences that contain both relation entities. If the truncated snippet cannot fit the BERT base token limit of 512, the entry is discarded. Applying this pre-processing allowed for a maximum length of 300 tokens for the BERT input and no discarded entries across all train, dev, and test sets.

To convert an MRE to an SRE problem means handling negative relations. Given a patent snippet with multiple entities identified, only a fraction of the pairings are actual relations of interest. The accompanying annotations provide the positive relations, but given unseen data, all possible combinations of trigger words and entities would need to be considered. However, the patent snippets contain an average of 25 entities and 8 unique trigger words each, resulting in an intractable number of pairs to consider.

Exploratory analysis of the corpus revealed that over 99% of relations occur within 5 entities of a trigger word. To process the test and dev set, all the positive relations provided by the annotation file were included, and negative relations that are within 5 entities of each trigger word were added. To process the test set, all relations within 5 entities of each trigger word were considered without consulting the annotation file for positive relations. Applying this processing resulted in 25 missed positive relations in the test set, which is a negligible amount representing less than one percent of the total positive relations in that set.

The overall statistics for the pre-processed train, dev, and test data are summarized in **Table 1**. The negative relations were labeled as NONE, and the problem becomes a multi-class classification. Typical of MRE, there is an imbalance between negative and positive relations, which amount to a baseline accuracy of 0.688 if the model only guesses the majority class.

4. Methods

This project consists of a matrix of nine BERT models with the same high-level architecture, but varies in the structure of the inputs into and the outputs from the BERT layer (**Figure 3**). Most of the architectures included in the matrix experiment are borrowed from the work of Soares et al., 2019.

The high-level architecture consists of fine-tuning a BERT model with the train and dev sets to obtain embedded representations for each relation. These representations are then fed into a dense layer with linear activation, followed by a dropout layer, and finally into a softmax layer for classification. All models use an Adam optimizer with learning rate of 3e-5, batch size of 32, and trained over 10 epochs. All models are evaluated with an F1-score obtained by inference using the test set.

Table 1 Data statistics for the pre-processed ChEMU 2020 corpus

	Snippets	Relations	Negative	Positive	ARG1	ARGM
Train	900	45805	68.8%	31.2%	21.2%	10.0%
Dev	225	10673	68.8%	31.2%	21.1%	10.2%
Test	375	18488	68.8%	31.2%	21.1%	10.2%

4.1 Types of BERT Inputs

Three different options or marker types were used for conveying information about the relation entities into the BERT encoder. The first type, referred to as Standard or Std, does not provide explicit identification of the entities. The second type, referred to as Entity Marker or EM, augments each entity with four reserved word pieces ([E1], [/E1], [E2], [/E2]) to mark the beginning and end of each entity mention.

Finally, the third type, referred to as NER Label or NER, also augments each entity with beginning and end markers. However, instead of using markers that will translate to unknown tokens, the beginning marker is replaced with a special token that encodes the entity's NER label, as described by Han et al., 2020. These special tokens (capital Greek letters in this case) are known by BERT but are rarely used in English text.

4.2 Types of BERT Output Representations

Four separate methods of extracting a fixed relation representation to be used in the classification layer are employed. The first method, referred to as [CLS], uses BERT's reserved token as the fixed relation representation. This is the common method used when using BERT for relation extraction and is the method used by both participants in the ChEMU 2020 challenge who used BERT.

The second method, referred to as Entity Start or Start, uses a concatenation of the embeddings of just the start tokens for each entity. This method can only be applied to Entity Marker and NER Label type inputs, since the Standard type contains no markers.

The third method, referred to as Mention Pool or Pool, takes all embeddings corresponding to the word pieces in the entity mention, takes the max-pool of each, and concatenates them to produce the relation representation. Finally, the

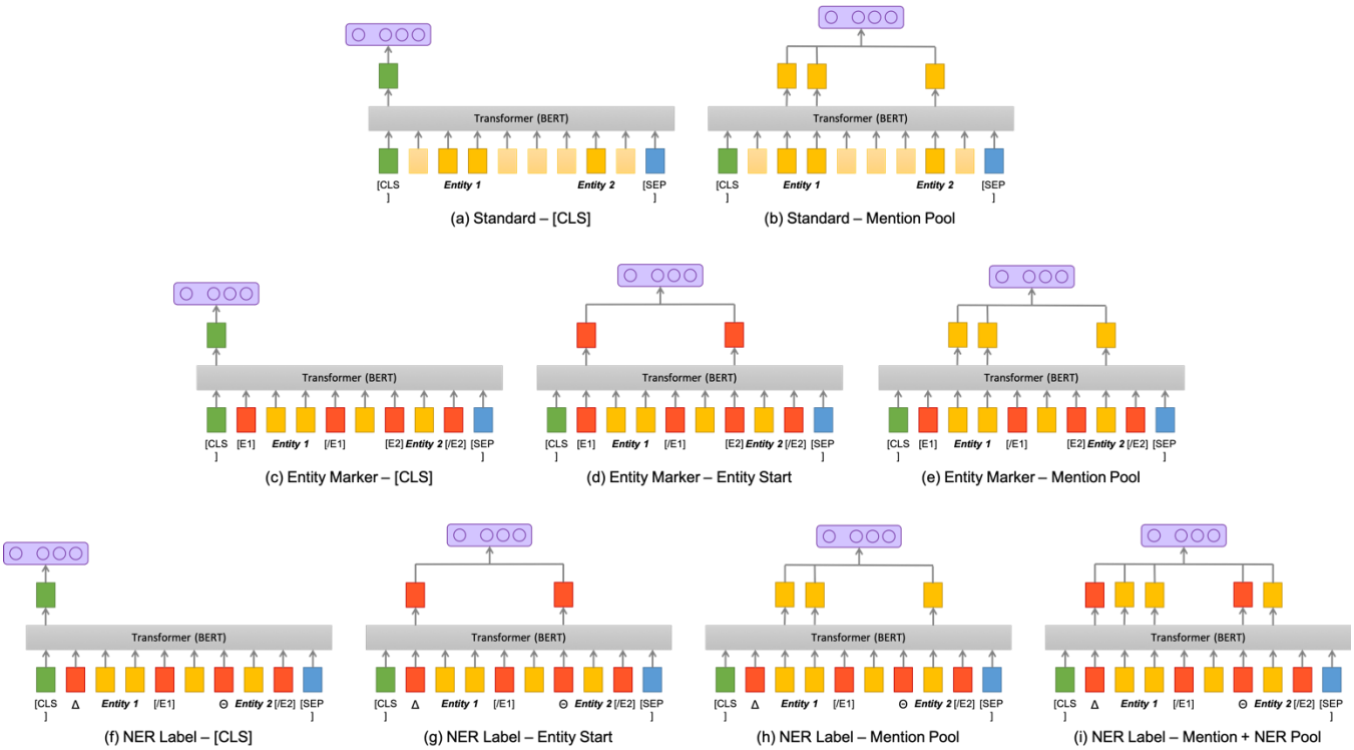

Figure 3 Diagram of input and output configurations tested in the matrix experiment

fourth method, referred to as Mention + NER Pool or Pool+, can only be applied to an NER label input and includes the NER label token before the max-pooling step in Mention Pool.

4.3 BERT Size

Despite truncating each snippet entry to just the sentences containing the relations, the inputs to BERT are still large at 300 tokens long. As a result, an inordinate amount of time is needed to run the full matrix experiment using BERT base, even on a TPU. Moreover, securing reliable and affordable TPU resources was a challenge.

The models were run using BERT-Tiny on a 16-node CPU VM (N1-standard) on Google Cloud Platform (GCP) in order to complete the experiment in a more timely and economical manner. BERT-Tiny is part of a release of 24 smaller BERT models intended for use on environments with restricted computational resources, which can be fine-tuned in the same manner as the original BERT models (Turc et al., 2019). **Table 2** summarizes the validation vs test F1-scores for matrix experiment run using BERT-Tiny. All the results are comparable to the scores obtained by the participants of the ChEMU 2020 challenge, and good correspondence between the validation and test scores indicates a good fit.

For comparison, a series of runs fined tuned using the sample data (1977 snippets) and evaluated with a subset of the test data (5961 snippets) was performed on the various sizes of BERT on a v3-8 TPU node on GCP. All runs used an NER Label – Mention + NER Pool architecture. Ten epochs were completed, and the average time per epoch was recorded along with the test F1-score in **Figure 4**. While these results show a noticeable drop in performance in BERT-Tiny compared to BERT base, it also highlights how the run time increases with model size. For reference, the average run time for the matrix experiments run on BERT-Tiny on the 16-node CPU VM with the full dataset averaged about 1250 seconds per epoch.

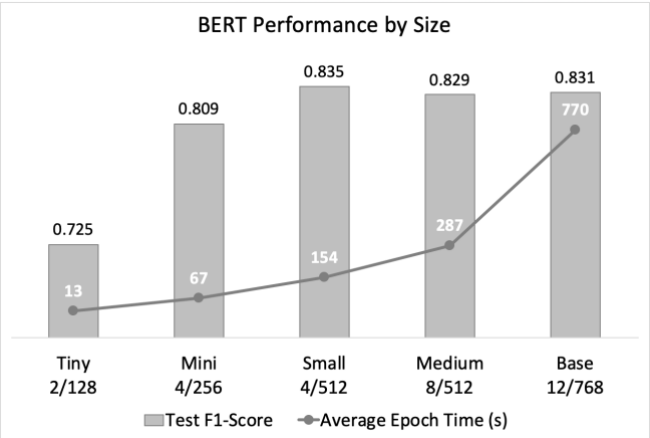

Figure 4 NER-Pool+ model results on various BERT sizes run on a v3-8 TPU node. The numbers below the models represent the transformer layer size / hidden embedding size

Table 2 Results summary for the matrix experiment

		VALIDATION				TEST			
Marker Type	**Head Type**	**Accuracy**	**Recall**	**Precision**	**F1-Score**	**Accuracy**	**Recall**	**Precision**	**F1-Score**
Standard	[CLS]	0.749	0.690	0.799	0.741	0.748	0.694	0.795	0.741
Entity Marker	[CLS]	0.757	0.708	0.800	0.751	0.751	0.702	0.792	0.744
NER Label	[CLS]	0.757	0.695	0.800	0.744	0.754	0.703	0.796	0.746
Entity Marker	Entity Start	0.829	0.812	0.844	0.828	0.826	0.807	0.840	0.823
NER Label	Entity Start	0.831	0.815	0.848	0.831	0.825	0.808	0.841	0.824
Standard	Mention Pool	0.909	0.907	0.910	0.909	0.908	0.906	0.909	0.908
Entity Marker	Mention Pool	0.930	0.928	0.931	0.929	0.927	0.925	0.929	0.927
NER Label	Mention Pool	0.932	0.931	0.933	0.932	0.929	0.928	0.931	0.929
NER Label	Mention + NER Pool	0.927	0.926	0.927	0.927	0.924	0.923	0.924	0.924

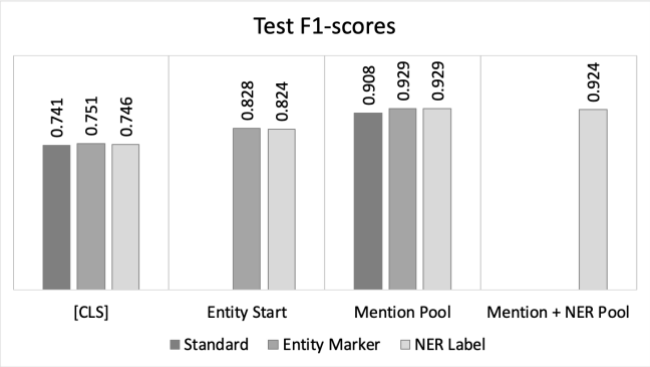

Figure 5 Results of the matrix experiment

5. Results and Discussion

5.1 Marker-Representation Matrix Experiment

The results of the marker-representation matrix experiment summarized in **Figure 5** show a clear trend of better performance for representations that incorporate more token embeddings. Adding a second token embedding to the representation afforded around 0.08 increase in the test F1-score for the models, while adding several more pooled embeddings increased the test F1-score by an even larger amount of around 0.1. The validation loss by epoch shown in **Figure 6** indicate a similar trend, with the pooled representations achieving significantly lower losses than either [CLS] or entity start type representations.

On the other hand, the type of marker used for input into the transformer layer does not exhibit as distinct of an effect. **Figure 5** indicates a slight improvement in using some form of marker for the entities, especially for the pooled representations. The test F1-score increases by about 0.01 and 0.02 for [CLS] and pooled representations, respectively, when using some form of entity marker. However, using generic markers vs specific NER labels to indicate the entities do not seem to make much difference in both test F1-score performance and validation loss. Furthermore, the added token embedding of the NER label in the pooled representation did not confer any added benefit to the performance of the model.

Figure 7 displays the fraction of each label correctly identified by each model in the matrix, which provides some insight into the trend in model performance. The one-token [CLS] representation appear to be biased towards predicting the majority class; it identifies the negative relations very well, but struggles with the positive relations. The two-token entity start representation shows similar performance with negative relations, but improves on identifying the positive relations by about 20%. Finally, the pooled representations identify the positive relations almost as well as the negative relations.

5.2 Reducing Tuning Data Size

Figure 8 explores the effect of reducing the amount of data available to tune the transformer model. Only the NER label type of marker is used to capture all four different representation types in the matrix in a comparable manner. The train and dev datasets were reduced by randomly picking a fraction of the files to process into input snippets, while the results were evaluated using the same test dataset as the matrix experiment.

While it is expected that performance will decrease with the amount of tuning data, **Figure 8** indicates that it happens at different rates depending on the representation type of the model. Models using the [CLS] representation does not seem to degrade as much as the pooled representations.

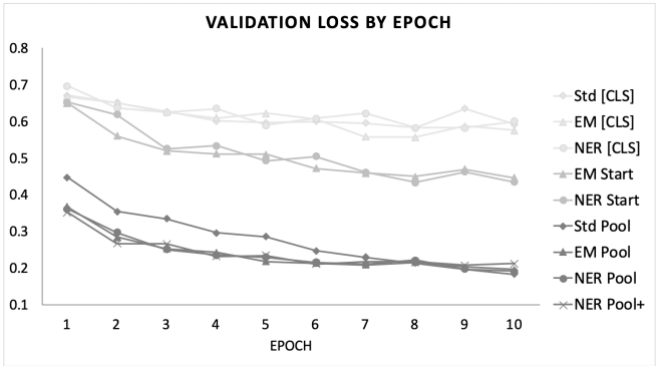

Figure 6 Validation loss by epoch for the matrix experiment

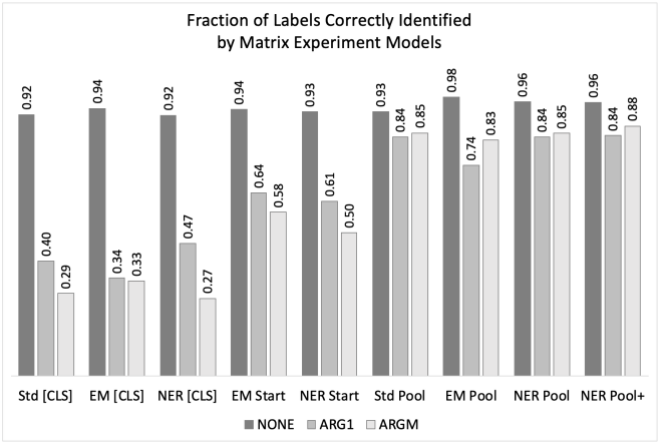

Figure 7 Error analysis for the matrix experiment

This trend can be explained by examining what happens to the predictive power of the models as the tuning data is reduced. **Figure 9** displays the fraction of each label correctly identified by each size of the Mention Pool model. As the amount of data decreases, the ability of the model to identify positive relations decrease, while its ability to identify negative relations remain fairly constant. Eventually, at just 1% of the tuning data, the model reverts to close to the baseline of only predicting the majority class. Thus, as the model using [CLS] representations was most biased toward predicting negative relations at the start, it follows that this model will also show the least change approaching the baseline.

5.3 Subsampling Negative Relations

To address the imbalance between positive and negative relations that can occur in multiple-relation extraction, Han et al., 2020 performed random subsampling of negative relations in their training set. **Figure 10** summarizes the results of applying the same subsampling to this dataset. As with the exploration of smaller models, only the NER label type of marker was used for this experiment.

Surprisingly, subsampling does not appear to help the performance of any of the models. However, the error analysis summarized in **Figure 11** provides a key insight: while subsampling slightly increases the model's ability to identify the minority classes, it simultaneously reduces model performance with the majority class. Since the majority class represents more relations, this performance reduction has the effect of reducing the overall performance of the model.

6. Conclusion

This project explores how different BERT architectures can be applied to the task of relation extraction of chemical patents. A matrix of different combinations of input and output representations for BERT models were tested. The best performances observed involved some manner of entity marker for the input along with mention pooling for the output.

The models were run economically with BERT-Tiny, which is a good option for exploratory work. Further experiments on the models indicated that reducing the size of the tuning data for BERT degrades the model towards the baseline of only predicting the majority class, while subsampling the negative relations counterproductively reduced the model performance for identifying the majority class.

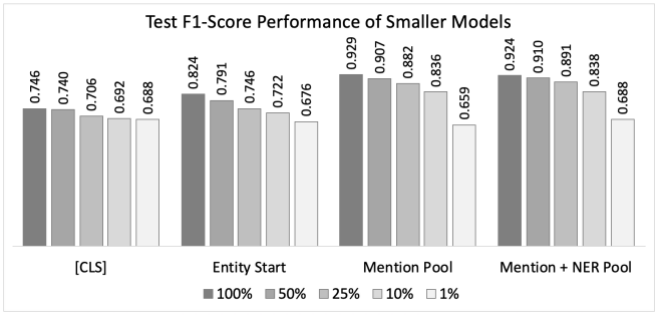

Figure 8 Effect of reducing tuning data size on test F1-score

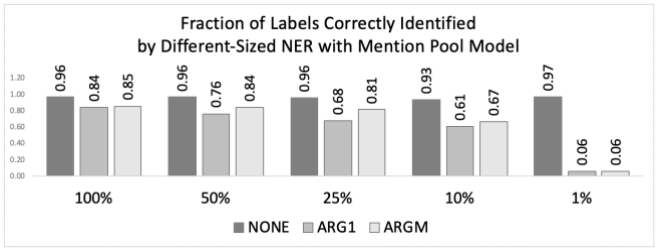

Figure 9 Error analysis for reducing tuning data experiment

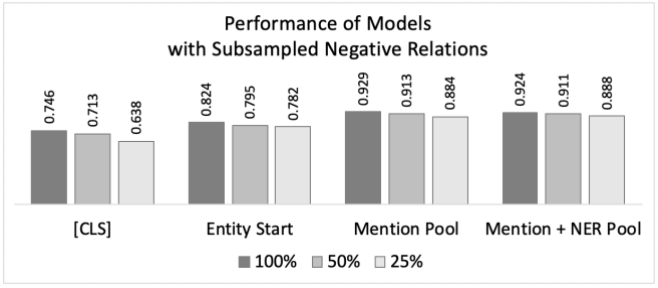

Figure 10 Effect of subsampling the negative relations on NER Label models

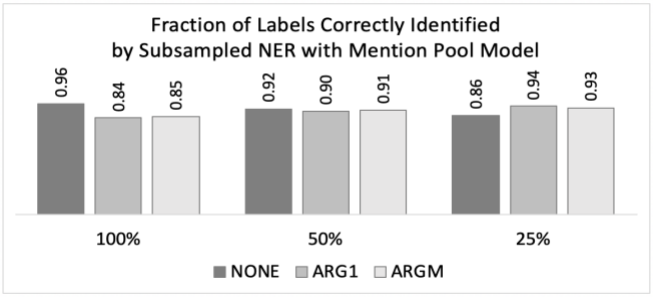

Figure 11 Error analysis for negative subsampling experiment

Acknowledgements

Code for the implementation of the model architectures by was largely derived and modified from jvasilakes (https://github.com/jvasilakes/BERT-RE). The BERT models were obtained from https://github.com/google-research/bert.

References

[1] Han X, Wang L. 2020 "A Novel Document-Level Relation Extraction Method Based on BERT and Entity Information" *IEEE Access* **8**. DOI: 10.1109/ACCESS.2020.2996642

[2] He J, Nguyen DQ, Akhondi S, Druckenbrodt C, et al. 2021 "ChEMU 2020: Natural Language Processing Methods Are Effective for Information Extraction from Chemical Patents" *Front. Res. Metr. Anal.* DOI: 10.3389/frma.2021.654438

[3] Köksal A, Hilal D, Özkirimli E, Aruzcan, Ö. 2020 "BOUN-REX at CLEF-2020 ChEMU task 2: evaluating pretrained trasnsformers for event extraction" *Working Notes of CLEF 2020-Conference and Labs of the Evaluation Forum* **2696** http://ceur-ws.org/Vol-2696/paper_206.pdf

[4] Soares L, FitzGerald N, Ling J, Kwiatkowski T. 2019 "Matching the Blanks: Distributional Similarity for Relation Learning" arXiv:1906.03158

[5] Turc I, Chang MW, Lee K, Toutanova K. 2019 "Well-Read Students Learn Better: On the Importance of Pre-training Compact Models" arXiv:1908.08962

[6] Vespoor K, Nguyen DQ, Akhondi S, Druckenbrodt C, Thorne, C, Hoessel R, He J, Zhai Z. 2020 "ChEMU dataset for information extraction from chemical patents." DOI: 10.17632/wy6745bjfj.2

[7] Wang J, Ren Y, Zhang Z, Zhang Y. 2020 "Melaxtech: A report for CLEF 2020 – ChEMU Task of Chemical Reaction Extraction from Patent" *Working Notes of CLEF 2020-Conference and Labs of the Evaluation Forum* **2696** http://ceur-ws.org/Vol-2696/paper_238.pdf