mgsohrab at WNUT 2020 Shared Task-1: Neural Exhaustive Approach for Entity and Relation Recognition Over Wet Lab Protocols

Mohammad Golam Sohrab[†], **Khoa N. A. Duong**[†],
Makoto Miwa[†, ‡], and **Hiroya Takamura**[†]
[†]Artificial Intelligence Research Center (AIRC)
National Institute of Advanced Industrial Science and Technology (AIST),
2-4-7 Aomi, Koto-ku, Tokyo, 135-0064, Japan
[‡]Toyota Technological Institute, Japan
{sohrab.mohammad, khoa.duong, takamura.hiroya}@aist.go.jp,
makoto-miwa@toyota-ti.ac.jp

Abstract

We present a neural exhaustive approach that addresses named entity recognition (NER) and relation recognition (RE), for the entity and relation recognition over the wet-lab protocols shared task. We introduce BERT-based neural exhaustive approach that enumerates all possible spans as potential entity mentions and classifies them into entity types or no entity with deep neural networks to address NER. To solve relation extraction task, based on the NER predictions or given gold mentions we create all possible trigger-argument pairs and classify them into relation types or no relation. In NER task, we achieved 76.60% in terms of F-score as third rank system among the participated systems. In relation extraction task, we achieved 80.46% in terms of F-score as the top system in the relation extraction or recognition task. Besides we compare our model based on the wet lab protocols corpus (WLPC) with the WLPC baseline and dynamic graph-based information extraction (DyGIE) systems.

1 Introduction

The entity and relation recognition over wet-lab protocol (Tabassum et al., 2020) shared task[1] is an open challenge that allows participants to use any methodology and knowledge sources for the wet lab protocols that specify the steps in performing a lab procedure. The task aims at two sub-tasks in wet lab protocols domain: named entity recognition (NER), and relation recognition or extraction (RE). In NER, the task is to detect mentions and classify them into entity types or no entity. NER has drawn considerable attentions as the first step towards many natural language processing (NLP) applications including relation extraction (Miwa and Bansal, 2016), event extraction (Feng et al., 2016), and co-reference resolution (Fragkou, 2017). In contrast, relation extraction (RE) is a task to identify relation types between known or predicted entity mentions in a sentence.

In this paper, we present a BERT-based neural exhaustive approach that addresses both NER and RE tasks. We employ a neural exhaustive model (Sohrab and Miwa, 2018; Sohrab et al., 2019b) for NER and the extended model that addresses RE task. The model detects flat and nested entities by reasoning over all the spans within a specified maximum span length. Unlike the existing models that rely on token-level labels, our model directly employs an entity type as the label of a span. The spans with the representations are classified into their entity types or non-entity. With the mentions predicted by the NER module, we then feed the detected or known mentions to the RE layer that enumerates all trigger-argument pairs as trigger-trigger or trigger-entity pairs and assigns a role type or no role type to each pair.

The best run for each sub-task achieved the F-score of 76.60% on entity recognition task that stands third rank system and the F-scores of 80.46% on relation extraction task as the top system. Besides, we also compare our model with the state-of-the-art models over the wet lab protocols corpus (WLPC). We compare the WLPC baseline model based on LSTM-CRF and maximum-entropy-based approaches to address NER and RE tasks respectfully. We also compare our model with dynamic graph-based information extraction (DyGIE) system. Our model outperforms by 4.81% for NER and 7.79% for RE over the WLPC baseline and 3.61% for NER over the DyGIE system.

2 Related Work

Most NER work focus on flat entities. Lample et al. (2016) proposed a LSTM-CRF (conditional ran-

[1] http://noisy-text.github.io/2020/wlp-task.html

dom fields) model and this has been widely used and extended for the flat NER, e.g., Akbik et al. (2018). In recent studies of neural network based flat NER, Gungor et al. (2018, 2019) have shown that morphological analysis using additional word representations based on linguistic properties of the words, especially for morphologically rich languages such as Turkish and Finnish, improves the NER performances further compared with using only representations based on the surface forms of words.

Recently, nested NER has been widely interested in NLP. Zhou et al. (2004) detected nested entities in a bottom-up way. They detected the innermost flat entities and then found other NEs containing the flat entities as sub-strings using rules on the detected entities. The authors reported an improvement of around 3% in the F-score under certain conditions on the GENIA data set (Collier et al., 1999). Recent studies show that the conditional random fields (CRFs) can produce significantly higher tagging accuracy in flat or nested (stacking flat NER to nested representation) NERs (Son and Minh, 2017). Ju et al. (2018) proposed a novel neural model to address nested entities by dynamically stacking flat NER layers until no outer entities are extracted. A cascaded CRF layer is used after the LSTM output in each flat layer. The authors reported that the model outperforms state-of-the-art results by achieving 74.5% in F-score on the GENIA data set. Sohrab and Miwa (2018) proposed a neural model that detects nested entities using exhaustive approach that outperforms the state-of-the-art results in terms of F-score on the GENIA data set. Sohrab et al. (2019b) further extended the span representations for entity recognition and addressed sensitive span detection tasks in the MEDDOCAN (MEDical DOCument ANonymization) shared task[2], and the system achieved 93.12% and 93.52% in terms of F-score for NER and sensitive span detection, respectively.

Recent successes in neural networks have shown impressive performance on coupling information extraction (IE) tasks as in joint modeling of entities and relations (Miwa and Bansal, 2016). Yi et al. (2019) proposed a dynamic graph information extraction (DyGIE) system for coupling multiple IE tasks, a multi-task learning approach to entity, relation, and coreference extraction. DyGIE uses dynamic graph propagation to explicitly incorporate rich contextual information into the span representations, and the system achieved significant F1 score improvement on the different datasets. Kulkarni et al. (2018) establised a baseline for IE on the wet lab protocols corpus (WLPC). They employ an LSTM-CRF for entity recognition approach. For relation extraction, they assume the presence of gold entities and train a maximum-entropy classifier using features from the labeled entities.

3 Neural Exhaustive Approach for NER and Relation Extraction

Our BERT-based neural exhaustive approach is built upon a pipeline approach of two modules:

- Named entity recognition that uses a contextual neural exhaustive approach
- Relation extraction that aims to predict relations from detected/given mentions.

To solve entity and relation recognition tasks, the pipeline approach can be presented as three layers: BERT layer, entity recognition layer, and relation recognition layer. Figure1 shows the system architecture of entity and relation recognition.

3.1 BERT Layer

For a given sequence, the BERT layer receives sub-word sequences and assigns contextual representations to the sub-words via BERT. We assume each sentence S has n words and the i-th word, represented by S_i, is split into sub-words. This layer assigns a vector $v_{i,j}$ to the j-th sub-word of the i-th word. It also produces the representation v_S as a local context for the sentence S, which corresponds to the embedding of [CLS] token.

3.2 Entity Recognition layer

We build mention detection layer, a.k.a named entity recognition (NER) on top of the BERT. This layer assigns entity or trigger types to overlapping text spans, or word sequences, in a sentence. We firstly generate mention candidates based on the same idea as the span-based model (Lee et al., 2017; Sohrab and Miwa, 2018; Sohrab et al., 2019a), in which all continuous word sequences are generated given a maximum span length L_x. Since BERT layer works only on sub-words, we choose the embedding of the first sub-word $v_{i,1}$ as word embedding v_i of i-th word. The representation $\boldsymbol{x}_{b,e} \in R^{d_x}$ for the span from the b-th word to the e-th word in a sentence is calculated from the

[2] http://temu.bsc.es/meddocan/

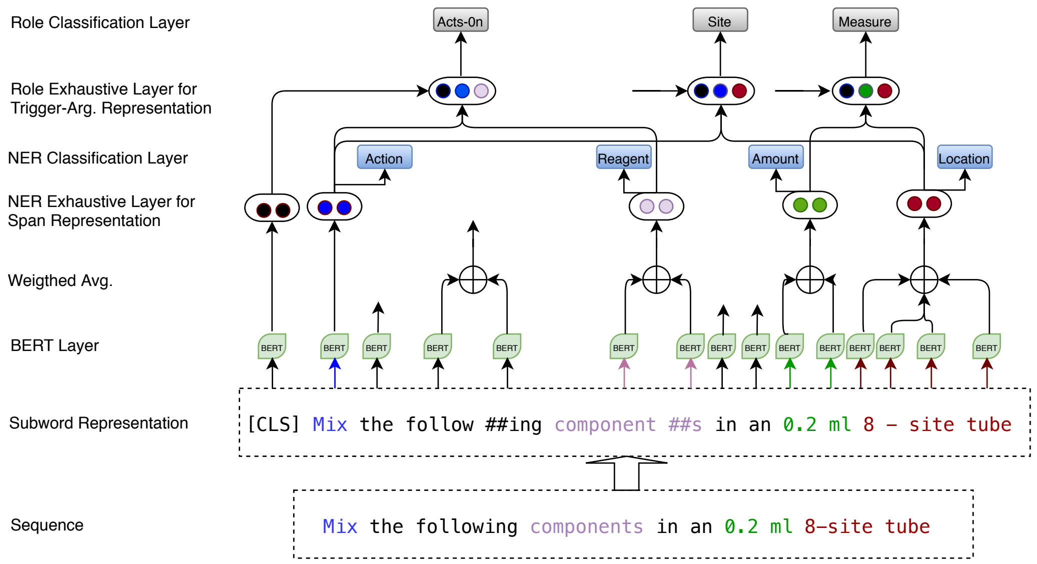
Role Classification Layer
Role Exhaustive Layer for Trigger-Arg. Representation
NER Classification Layer
NER Exhaustive Layer for Span Representation
Weigthed Avg.
BERT Layer
Subword Representation
Sequence
Acts-On
Site
Measure
Action
Reagent
Amount
Location
BERT
[CLS] Mix the follow ##ing component ##s in an 0.2 ml 8 – site tube
Mix the following components in an 0.2 ml 8-site tube

Figure 1: System Architecture for Neural Exhaustive Approach for NER and Relation Extraction. The example sequence is taken from Wet Lab Protocols Data set

embeddings of the first word, the last word, and the weighted average of all words in the span as follows:

$$\boldsymbol{x}_{b,e} = \left[\boldsymbol{v}_b; \sum_{i=b}^{e} \boldsymbol{\alpha}_{b,e,i} \boldsymbol{v}_i; \boldsymbol{v}_e \right], \qquad (1)$$

where $\boldsymbol{\alpha}_{b,e,i}$ denotes the attention value of the i-th word in a span from the b-th word to the e-th word, and $[;;]$ denotes concatenation.

3.3 Relation Recognition Layer

The relation recognition layer enumerates all trigger-argument pairs (trigger-trigger and trigger-entity pairs) given triggers and entities detected by the entity recognition layer and assigns a role type or no role to each pair. We generate relation representation based on the same idea as the deep event extraction system (Trieu et al., 2020).

Since each role is constructed by a trigger and an argument, we firstly compute representations of all triggers and arguments detected by the entity recognition layer. The representations of a trigger and an argument are calculated in the same way. A trigger t ranging from the starting t_s-th word to the ending t_e-th word is represented with the concatenation of its span representation x_t (from Equation 1) and a 300-dimensional entity type embedding s_t, as follows:

$$\boldsymbol{v}_t = [\boldsymbol{x}_t; \boldsymbol{s}_t], \qquad (2)$$

Similarly, the representation of an argument a can be calculated as

$$\boldsymbol{v}_a = [\boldsymbol{x}_a; \boldsymbol{s}_a]. \qquad (3)$$

The representation $r_i \in R^{d_r}$ for a relation pair i is then calculated from its trigger representation v_t , argument representation v_a, and the context representation v_S which is obtained from the sentence representation of the BERT layer:

$$\boldsymbol{r}_i = GELU\left(\boldsymbol{w}_r\left[\boldsymbol{v}_t; \boldsymbol{v}_a; \boldsymbol{v}_S\right] + \boldsymbol{b}_r\right), \qquad (4)$$

where W_r and b_r are learnable weights and biases respectively and GELU is the Gaussian Error Linear Unit activation function. After obtaining the pair representation r_i, we classify it with a softmax function to predict the corresponding role type.

4 Experimental Settings

We provide empirical evidence on the effectiveness of the pipeline architecture in both NER and relation extraction over the wet lab protocols[3] task of the W-NUT 2020[4]. The wet lab protocols corpus with eighteen entity types[5] and fifteen relation types[6] are randomly split into four subsets: train,

[3] http://noisy-text.github.io/2020/wlp-task.html

[4] http://noisy-text.github.io/2020/

[5] Entity Type: Action, Seal, Numerical, Concentration, Size, Modifier, Measure-Type, Generic-Measure, Time, Speed, Action, Location, Method, Temperature, Mention, pH, Device, Amount, Reagent

[6] Relation Type: Coreference-Link, Measure, Site, Meronym, Measure-Type-Link, Product, Commands, Mod-Link, Count, Acts-on, Using, Creates, Setting, Of-Type, Or

development, test, and test release (unlabeled) sets, which contain 370, 122, 123 and 111 lab protocols respectively. In our experiments, we merge the train and development as train-set, test-set use as development-set, and predict the annotations for test release set which is used as test-set.

Our model is implemented in the PyTorch[7] framework. We employed the official wet lab protocols evaluation script for NER[8] and relation extraction[9] to evaluate our system's performances on both tasks.

4.1 Data Preprocessing

Each text and the corresponding annotation file were preprocessed by several simple rules[10] only for tokenization[11]. After tokenization, each text with mapping annotation files were directly passed to the deep neural approach for mention detection and relation extraction. Note that the offsets were restored to the original offsets in evaluation.

4.2 Training Settings

We train the model in a pipeline manner based on the pre-trained BERT model. We employed the pre-trained PubmedBERT (Gu et al., 2020) model which is an uncased BERT Base model that was pretrained over PubMed abstracts and full PubMed central articles. Besides, we also employed SciBERT (Beltagy et al., 2019) model that is pre-trained based on large-scale biomedical text. Moreover, we also employed original pre-trained BERT (Devlin et al., 2019) model which is a uncased BERT base model to judge the performances of our model among the PubmedBERT, SciBERT and BERT.

According to our investigation, we choose 10 as the maximum span length of mention candidates. We also truncate every sentences at 256 sub-words without losing any gold entities or relations (we maintain a 100% recall of gold entities and relations in the training set).

NER and RE models are trained on 100 epochs with learning rate of 0.00003.

5 Results and Discussions

In order to evaluate the performance of NER, we conduct experiments on different sets of BERT-based learning representations, including PubmedBERT with merging training- and dev-set (PubmedBERT-Merge), PubmedBERT along with training (PubmedBERT-Train), SciBERT with merging training- and dev-set (SciBERT-Merge), and SciBERT along with training (SciBERT-Train).

In contrast to relation extraction, as based on our primary results of NER with PubmedBERT and SciBERT where PubmedBERT is outperforming to SciBERT. Therefore, we conduct all our relation extraction experiments using PubmedBERT. For relation extraction task, we learn our model on two data scenarios. First, we perform a clustering approach on training- and dev-set to find the similar or duplicate text files in wet-lab data set. We found that many similar text files with inconsistent annotations exist in the train- and dev-set. The similarity approach with a setting threshold is applied on the train and dev-set to cluster the similar or duplicate text protocols. We then eliminate those text and its corresponding annotation files which appear in the training set to avoid model learning confusion and data leakage. We also applied the predefined relation rules (Kulkarni et al., 2018) to filter out any invalid relations appearing in the system output. We conduct experiments on different sets of PubmedBERT-based learning representations, including PubmedBERT using finetune with filtering approach (PubmedBERT-Finetune-Filter), PubmedBERT along with finetune (PubmedBERT-Finetune), PubmedBERT along with filter approach (PubmedBERT-Filter) and PubmedBERT without finetune and filtering approaches (PubmedBERT).

In second data scenario, we learn our model by keeping all the original training set, development set, and test set. Based on original data setting, the PubmedBERT-based learning representations are PubmedBERT-Original-Finetune-Filter, PubmedBERT-Original-Finetune, PubmedBERT-Original-Filter and PubmedBERT-Original.

We also report the result of ensemble learning that combines the predictions using different span representations to reduce the variance of predictions and the generalization error.

5.1 NER Performances

Table 1 shows the results of NER on the dev- and test-set. Here, the PubmedBERT-Merge, SciBERT-

[7] https://pytorch.org

[8] https://github.com/jeniyat/WNUT_2020_NER/tree/master/code/eval

[9] https://github.com/jeniyat/WNUT_2020_RE/blob/master/code/evaluation.py

[10] We also published our preprocessing script at https://github.com/dnanhkhoa/WNUT-2020

[11] Unlike the traditional NER models, our model is independent from traditional 'BIO' tagging scheme, where 'B', 'I', and 'O' stand for 'Begin', 'Inside', and 'Outside' of named entities respectively, so we do not need to assign such tags to the tokens.

	Dev: NER			Test: NER		
Learning Approach	P	R	F	P	R	F(%)
Ensemble	**83.14**	83.28	**83.21**	83.69	70.62	76.60
PubmedBERT-Merge (Train+dev)	82.04	**83.51**	82.77	80.59	71.57	75.81
SciBERT-Merge (Train+Dev)	82.47	82.80	82.64	80.79	70.40	75.24
PubmedBERT-Train	82.46	79.23	80.81	83.66	69.59	75.98
SciBERT-Train	80.68	80.46	80.57	80.78	71.70	75.97

Table 1: Performance of NER on the dev- and test-set

	Exact Match			Partial Match		
Team Name	P	R	F	P	R	F(%)
BITEM	**84.73**	72.25	**77.99**	**88.72**	75.66	81.67
PublishInCovid19	81.36	**74.12**	77.57	85.74	**78.11**	**81.75**
mgsohrab	83.69	70.62	76.60	87.95	74.22	80.50
Kabir	78.79	72.20	75.35	83.73	76.73	80.08
IITKGP	77.00	72.93	74.91	81.76	77.43	79.54
BIO-BIO	78.49	71.06	74.59	83.16	75.29	79.03
Fancy Man Launches Zippo	76.21	71.76	73.92	81.15	76.41	78.71
SudeshnaTCS	74.99	71.43	73.16	79.73	75.95	77.80
B-NLP	77.95	63.93	70.25	84.85	69.59	76.46
KaushikAcharya	73.68	63.98	68.48	79.31	68.87	73.73
IBS	74.26	62.55	67.90	79.72	67.15	72.89
DSC-IITISM	64.20	57.07	60.42	68.52	60.90	64.49
mahab	50.19	52.96	51.54	55.09	58.14	56.57

Table 2: Team performances of NER on the test-set

	Dev: RE			Test: RE		
Learning Approach	P	R	F	P	R	F(%)
Ensemble	88.16	**86.91**	**87.53**	80.86	**80.07**	**80.46**
PubmedBERT-Finetune-Filter	**88.59**	85.47	87.00	**83.03**	77.35	80.09
PubmedBERT-Finetune	88.55	85.47	86.99	82.93	77.36	80.05
PubmedBERT-Filter	88.54	84.84	86.65	81.96	75.96	78.84
PubmedBERT	88.50	84.84	86.63	81.92	75.97	78.83
PubmedBERT-Original-Finetune-Filter	87.85	86.36	87.10	78.67	79.03	78.85
PubmedBERT-Original-Finetune	87.85	86.36	87.10	78.59	79.03	78.81
PubmedBERT-Original-Filter	88.09	85.15	86.60	80.36	77.48	78.89
PubmedBERT-Original	88.04	85.15	86.57	80.30	77.48	78.87

Table 3: Performance of relation extraction (RE) on the dev- and test-set

	Relation Extraction		
Team Name	P	R	F(%)
mgsohrab	**80.86**	80.07	**80.46**
Big Green	45.42	**86.54**	59.57

Table 4: Team performances of relation extraction (RE) on the test-set

Merge, PubmedBERT-Train, and SciBERT-Train are used for ensemble approach. In this table, it is shown that the ensemble approach using maximum voting of all the approaches is effective to improve the NER system performance with achieving 83.21% and 76.60% in terms of F-score over the dev- and test-set respectfully. In contrast, the PubmedBERT-Merge shows the best performance as an individual learning on NER with achieving

	NER Test-set			RE Test-set		
Entity Level	P	R	F	P	R	F(%)
All	82.70	71.25	76.55	80.96	80.05	80.50
Single-token	**85.43**	**72.05**	**78.17**	**82.95**	78.97	**80.91**
Multi-token	77.73	69.70	73.50	79.01	**81.20**	80.09

Table 5: Performances of NER and RE of our model on different entity level on the test-set

	NER			RE		
Model	P	R	F	P	R	F(%)
WLPC Baseline (Kulkarni et al., 2018)	–	–	78.30	80.98	77.04	78.96
DyGIE (Yi et al., 2019)	–	–	79.50	–	–	∗64.10
Our Model	82.83	83.40	**83.11**	**88.75**	**84.86**	**86.75**

Table 6: Performance comparison of NER and RE based on different models on the wet lab protocols dataset. '-' denotes results are not reported in the original paper. '*' indicates the performance of relation extraction system is based on predicted entity boundary as input.

Label	P	R	F(%)	Prediction	Annotation	Correct
Action	90.71	92.92	91.80	4239	4138	3845
Concentration	85.45	85.61	85.53	536	535	458
Reagent	85.20	86.43	85.81	4053	3995	3453
Amount	91.78	91.93	91.86	1192	1190	1094
Location	79.27	78.43	78.85	1312	1326	1040
Method	65.36	54.56	59.47	485	581	317
Time	91.77	91.03	91.40	863	870	792
Temperature	93.63	91.17	92.38	518	532	485
Device	69.92	70.51	70.21	472	468	330
Modifier	64.68	66.58	65.62	1648	1601	1066
Size	75.70	71.68	73.64	107	113	81
Mention	64.29	80.36	71.43	70	56	45
Ph	83.64	74.19	78.63	55	62	46
Numerical	65.06	70.13	67.50	249	231	162
Seal	70.00	65.62	67.74	60	64	42
Measure-type	65.53	56.62	60.75	235	272	154
Speed	90.75	94.01	92.35	173	167	157
Generic-measure	41.90	30.77	35.48	105	143	44
Overall (micro)	83.14	83.28	83.21	16372	16344	13611

Table 7: Categorical performances of NER on the dev-set

82.77% in terms of F-score.

Table 2 shows the NER task results on the participated teams. In this table results are listed in descending order in terms of exact match-based F-score. The top system achieves 77.99% where our team achieves 76.60% in terms of F-score for NER task.

5.2 Relation Extraction Performances

Table 3 shows the results of relation extraction task on the dev- and test-set. Here, all the reported learning approaches in this table are used for ensemble approach. In this table, it is shown that the ensemble approach using maximum voting of all the approaches is also effective to improve the relation extraction system performance with achieving 87.53% and 80.46% in terms of F-score over the dev- and test-set respectfully. In contrast, the PubmedBERT-Original-Finetune-Filter and PubmedBERT-Finetune-Filter are showing the best performances as an individual learning on relation extraction with achieving 87.10% and 80.09%

Label	P	R	F(%)	Prediction	Annotation	Correct
Coreference-Link	69.64	46.43	55.71	56	84	39
Measure	92.66	91.20	91.93	1880	1910	1742
Site	81.61	87.82	84.61	1202	1117	981
Meronym	73.97	65.98	69.74	388	435	287
Measure-Type-Link	87.90	88.62	88.26	124	123	109
Product	43.18	20.00	27.34	44	95	19
Commands	07.14	05.00	05.88	14	20	1
Mod-Link	92.52	91.85	92.18	1510	1521	1397
Count	87.37	83.84	85.57	95	99	830
Acts-On	91.44	89.82	90.63	3050	3105	2789
Using	77.04	75.50	76.26	832	849	641
Creates	00.00	00.00	00.00	0	0	0
Setting	91.13	92.48	91.80	1713	1688	1561
Of-Type	75.00	54.55	63.16	16	22	12
Or	68.99	62.64	65.66	158	174	109
Overall (micro)	88.16	86.91	87.53	11082	11242	9770

Table 8: Categorical performances of relation extraction (RE) on the dev-set

	RE		
Training Strategy	P	R	F(%)
Not pre-finetune NER layer	**88.09**	85.15	86.60
Pre-finetune NER layer using gold entities	87.85	**86.36**	**87.10**

Table 9: Performance of relation extraction (RE) using different training strategies on the dev-set

	NER			RE		
Entity	P	R	F	P	R	F(%)
BERT-base-uncased	84.18	83.19	83.68	87.84	85.55	86.83
PubmedBERT-full-uncased	**84.50**	**83.70**	**84.10**	**87.85**	**86.36**	**87.10**

Table 10: Performance of NER and relation extraction (RE) using different BERT-based learning on the dev-set

Span Length	P	R	F(%)
8	82.65	82.55	82.60
10	**82.83**	82.89	**82.86**
12	81.95	**83.71**	82.82

Table 11: Performance of our model with different spans on the dev-set

in terms of F-score over the dev- and test-set respectively.

Table 4 shows the relation extraction task results on the participated teams. Our relation extraction system achieves 80.56% in terms of F-score as a top system in this task. We outperformed the second best system by 20.89% in terms of F-score.

Our system is based on span-based representation, therefore we also investigate the performances for all vs single-token vs multi-token entities. Table 5 shows the break down performances of our model on different entity levels over the NER and relation extraction on the test set.

In contrast, we also compare our model with the state-of-the-art models over the wet lab protocols corpus (WLPC). Table 6 shows the comparison of our model with the WLPC baseline and DyGIE systems. In NER, our model outperforms the WLPC baseline and DyGIE systems by 4.81% and 3.61% respectively in terms of F-score. In RE, our model outperforms the WLPC baseline by 7.79% in terms of F-score. In compare the RE task for DyGIE, NER predictions are given as input in DyGIE where gold data boundary is given as input in our and WLPC baseline models. We report the DyGIE RE performance without comparing with our RE performance. In these comparisons, we use the same train-, dev-, test-set, and evaluation script that reported in the WLPC baseline (Kulkarni et al., 2018) for fair comparisons.

5.3 Ablation Study

We show the performances of different BERT-based learning models for NER and relation extraction tasks on the development column in Table 1 and Table 3 to compare the possible scenarios of the given solutions and to report the best system submissions for NER and relation extraction. For NER and relation extraction tasks, all the results in Table 1 and Table 3 in development column window show that almost all the results in different approaches are close to each other to solve the NER and relation extraction tasks.

Table 7 shows the categorical performances using ensemble learning of NER on the dev-set. In this table, we also break down the number of predicted and correct mentions among the gold annotations of each category. Here, prediction can be denoted as number of predicted entities, annotation as number of gold entities of each category, and correct as number of true positive outcomes where the model correctly predicts the positive category. In this table, it can be observed that for the frequent classes (e.g. Action, Reagent, Amount etc.), the model shows high performance because there are a reasonable number of training instances for the classes. In contrast, for the rare classes (e.g. Size, Mention, Ph, numerical etc.), the performances are also consistence. Table 8 shows the categorical performances using ensemble learning of relation extraction on the dev-set. In this table, it shows the categorical performances using ensemble learning of relation extraction on the development set. In this table, it seems that the model is well generalized to classify the relation types that leads to achieve the top system in the shared task.

Since we provided gold entities in the RE task, therefore, we also examine two different strategies for training the RE model that present in the Table 9. In this table, it shows that we can significantly boost the RE performance just by pre-finetuning the NER layer using gold entities. Table 10 shows the performances of NER and RE based on the original BERT base in compare to the PubmedBERT. The results show that PubmedBERT is outperformed both in NER and RE tasks. In Table 11, we compared our model in different span length. We chose the maximum span size from 8, 10, and 12 that covers more than 99% mentions to judge the sensitivity of our approach in different span length. In ths table, it can be observed that the performances of our model are consistence even with different span lengths.

6 Conclusion

This paper presented a BERT-based neural exhaustive approach that addresses both named entity recognition (NER) and relation extraction (RE) tasks. This neural approach consider all possible spans exhaustively, for NER which is capable to detect flat and nested entities from the generated mention candidates.

Several enhancements, namely PubmedBERT, SciBERT, BERT-base-uncased, filtering, clustering, and ensembling are investigated for the wet-lab protocol data set to enhance the system performance. In NER task, we achieved 76.60% in terms of F-score as third rank system among the participated systems. In relation extraction task, we achieved 80.46% in terms of F-score as the top system that participated in the relation extraction task. Moreover, our model outperforms by 4.81% for NER and 7.79% for RE over the WLPC baseline and 3.61% for NER over the DyGIE system.

In the future direction, we will implement a joint modeling that addresses NER and relation extraction in an end-to-end manner.

Acknowledgments

This work is based on results obtained from a project commissioned by the Public/Private R&D Investment Strategic Expansion PrograM (PRISM).

References

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *COLING 2018, 27th International Conference on Computational Linguistics*, pages 1638–1649.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: A pretrained language model for scientific text. 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)*, pages 3606–3611.

N. Collier, H. S. Park, N. Ogata, Y. Tateisi, C. Nobata, T. Ohta, T. Sekimizu, H. Imai, K. Ibushi, and

Jun'ichi Tsujii. 1999. The GENIA Project: Corpus-based Knowledge Acquisition and Information Extraction from Genome Research Papers. In *Proceedings of EACL*, pages 171–172. ACL.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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)*, pages 4171–4186.

Xiaocheng Feng, Lifu Huang, Duyu Tang, Heng Ji, Bing Qin, and Ting Liu. 2016. A Language-Independent Neural Network for Event Detection. In *Proceedings of the 54th Annual Meeting of the ACL (Volume 2: Short Papers)*, pages 66—71, Berlin, Germany.

Pavlina Fragkou. 2017. Applying named entity recognition and co-reference resolution for segmenting english texts. *Progress in Artificial Intelligence*, 6(4):325—346.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2020. Domain-specific language model pretraining for biomedical natural language processing. *arXiv preprint arXiv:2007.15779*.

Onur Gungor, Tunga Gungor, and Suzan Uskudarli. 2019. The effect of morphology in named entity recognition with sequence tagging. *Natural Language Engineering*, 25(1):147–169.

Onur Gungor, Suzan Uskudarli, and Tunga Gungor. 2018. Improving named entity recognition by jointly learning to disambiguate morphological tags. In *COLING 2018, 27th International Conference on Computational Linguistics*, pages 2082–2092.

Meizhi Ju, Makoto Miwa, and Sophia Ananiadou. 2018. A Neural Layered Model for Nested Named Entity Recognition. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1446—1459, New Orleans, Louisiana. ACL.

Chaitanya Kulkarni, Wei Xu, Alan Ritter, and Raghu MachirajuYi. 2018. An annotated corpus for machine reading of instructions in wet lab protocols. In *Proceedings of NAACL-HLT 2018*.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural Architectures for Named Entity Recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the ACL: Human Language Technologies. ACL*, volume 1, pages 260—-270, San Diego, California. ACL.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 188–197, Copenhagen, Denmark. Association for Computational Linguistics.

Makoto Miwa and Mohit Bansal. 2016. End-to-End Relation Extraction using LSTMs on Sequences and Tree Structures. In *Proceedings of the 54th Annual Meeting of the ACL*, pages 1105—1116, Berlin, Germany. ACL.

Mohammad Golam Sohrab and Makoto Miwa. 2018. Deep exhaustive model for nested named entity recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2843–2849, Brussels, Belgium. Association for Computational Linguistics.

Mohammad Golam Sohrab, Minh Thang Pham, Makoto Miwa, and Hiroya Takamura. 2019a. A neural pipeline approach for the pharmaconer shared task using contextual exhaustive models. In *Proceedings of The 5th Workshop on BioNLP Open Shared Tasks*, pages 47–55.

Mohammad Golam Sohrab, Pham Minh Thang, and Makoto Miwa. 2019b. A generic neural exhaustive approach for entity recognition and sensitive span detect. In *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2019)*, pages 735–743, Span. IberLEF 2019.

Nguyen Truong Son and Nguyen Le Minh. 2017. Nested Named Entity Recognition Using Multilayer Recurrent Neural Networks. In *Proceedings of PACLING 2017*, pages 16–18, Sedona Hotel, Yangon, Myanmar.

Jeniya Tabassum, Sydney Lee, Wei Xu, and Alan Ritter. 2020. WNUT-2020 Task 1 Overview: Extracting Entities and Relations from Wet Lab Protocols. In *Proceedings of EMNLP 2020 Workshop on Noisy User-generated Text (WNUT)*.

Hai-Long Trieu, Thy Tran, Khoa N A Duong, Anh Nguyen, Makoto Miwa, and Sophia Ananiadou. 2020. DeepEventMine: End-to-end Neural Nested Event Extraction from Biomedical Texts. *Bioinformatics*. Btaa540.

Luan Yi, Wadden Dave, He Luheng, Shah Amy, Ostendorf Mari, and Hajishirzi Hannaneh. 2019. A general framework for information extraction using dynamic span graphs. In *Proceedings of NAACL-HLT 2019*.

Guodong Zhou, Jie Zhang, Jian Su, Dan Shen, and Chewlim Tan. 2004. Recognizing Names in Biomedical Texts: a Machine Learning Approach. *Bioinformatics*, 20(7):1178—1190.