YNU-HPCC at SemEval-2021 Task 11: Using a BERT Model to Extract Contributions from NLP Scholarly Articles

Xinge Ma, Jin Wang and Xuejie Zhang
School of Information Science and Engineering
Yunnan University
Kunming, China
Contact: maxinge@mail.ynu.edu.cn, {wangjin, xjzhang}@ynu.edu.cn

Abstract

This paper describes the system we built as the YNU-HPCC team in the SemEval-2021 Task 11: *NLPContributionGraph*. This task involves first identifying sentences in the given natural language processing (NLP) scholarly articles that reflect research contributions through binary classification; then identifying the core scientific terms and their relation phrases from these contribution sentences by sequence labeling; and finally, these scientific terms and relation phrases are categorized, identified, and organized into subject-predicate-object triples to form a knowledge graph with the help of multiclass classification and multi-label classification. We developed a system for this task using a pre-trained language representation model called BERT that stands for Bidirectional Encoder Representations from Transformers, and achieved good results. The average F_1-score for Evaluation Phase 2, Part 1 was 0.4562 and ranked 7th, and the average F_1-score for Evaluation Phase 2, Part 2 was 0.6541, and also ranked 7th.

1 Introduction

As the number of research publications increases, there is a growing need for digital libraries to equip researchers with alternative knowledge representations. In addition, because scientific literature is growing at a rapid rate and researchers today are faced with a publication deluge, it is difficult to keep up with the research progress even within ones own narrow discipline. The open research knowledge graph (ORKG) (Jaradeh et al., 2019) is posited as a solution to the problem of keeping track of research progress without the cognitive overload imposed by reading dozens of full papers. To this end, the aim of this task is to build a comprehensive knowledge graph that represents the research contributions of scholarly publications per paper and also shows where the contributions are interconnected across papers (D'Souza and Auer, 2020).

The task was defined on a dataset containing natural language processing (NLP) scholarly articles with their contributions structured to be integrable within a knowledge graph infrastructure, such as the ORKG. The structured contribution annotations were provided as follows: (1) contribution sentences: a set of sentences about the contribution in the article; (2) scientific terms and relations: a set of scientific terms and relational cue phrases extracted from the contribution sentences; and (3) triples: semantic statements that pair scientific terms with a relation, modeled toward the subject-predicate-object statements for building knowledge graph. The triples were organized under three (mandatory) or more of the twelve total information units (i.e., *ResearchProblem*, *Approach*, *Model*, *Code*, *Dataset*, *ExperimentalSetup*, *Hyperparameters*, *Baselines*, *Results*, *Tasks*, *Experiments*, and *AblationAnalysis*). An illustration of this process is shown in Figure 1.

The difficulty of this task lies in text classification (Joulin et al., 2017) and sequence labeling (Ma and Hovy, 2016). Text classification refers to determining which of the two or more labels a one-dimensional linear sequence belongs to. Similarly, sequence labeling is used to tag each element in a one-dimensional linear sequence with a label from a set of labels. Before the popularity of deep learning, the common solutions to the sequence-labeling problem were all based on either the hidden Markov model (Zhou and Su, 2001) or conditional random field (CRF) (Ye and Ling, 2018), with CRF being the mainstream method. With the development of deep learning, convolutional neural networks (CNN) (Kim, 2014) and recurrent neural networks (RNN) (Cho et al., 2014) have achieved great success in text classification and se-

Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), pages 478–484
Bangkok, Thailand (online), August 5–6, 2021. ©2021 Association for Computational Linguistics

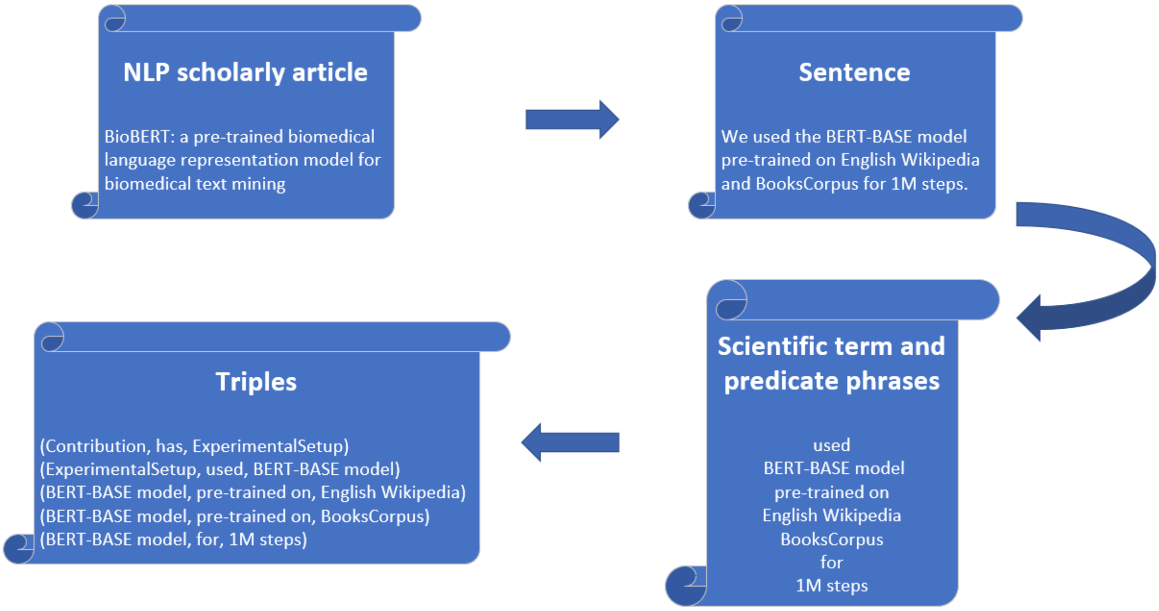

NLP scholarly article
BioBERT: a pre-trained biomedical language representation model for biomedical text mining
Sentence
We used the BERT-BASE model pre-trained on English Wikipedia and BooksCorpus for 1M steps.
Scientific term and predicate phrases
used
BERT-BASE model
pre-trained on
English Wikipedia
BooksCorpus
for
1M steps
Triples
(Contribution, has, ExperimentalSetup)
(ExperimentalSetup, used, BERT-BASE model)
(BERT-BASE model, pre-trained on, English Wikipedia)
(BERT-BASE model, pre-trained on, BooksCorpus)
(BERT-BASE model, for, 1M steps)

Figure 1: Example of contribution extraction of NLP scholarly articles

quence labeling. Since then, long short-term memory (LSTM) (Wang and Jiang, 2016), Bi-LSTM (Bi-directional long short-term memory), and other models (Yuan et al., 2020) have performed better than CNN and RNN in text classification and sequence labeling . However, since the introduction of bidirectional encoder representations from transformers (BERT) (Devlin et al., 2018), the accuracy and training efficiency in both text classification and sequence labeling have reached new heights.

The SemEval-2021 shared Task 11 (D'Souza et al., 2021) consists of three subtasks:

- Subtask 1: identifying contribution sentences;
- Subtask 2: identifying scientific terms and predicate phrases;
- Subtask 3: categorizing, identifying and organizing scientific terms and predicate phrases into subject-predicate-object triples.

In this study, after analysis, we converted the above three subtasks into four downstream tasks in the field of NLP: binary classification for solving Subtask 1, sequence labeling for solving Subtask 2, multiclass classification and multi-label classification for solving Subtask 3. Then, we used a pre-trained language model, BERT, to generate word embeddings and integrated them into the corresponding models for the different tasks. After completion of the task, our results were satisfactory. Our submission ranked 7th in both Part 1 and Part 2 of Evaluation Phase 2. The implementation for our system is made available via Github[1].

The remainder of this paper is organized as follows. Section 2 describes the details of the BERT model used in our system. Section 3 presents the experimental results. Finally, the conclusions are presented in Section 4.

2 System Description

We used a pre-trained BERT model to accomplish the task, which was defined in terms of three dataset annotation elements, where the extraction of each data element relied on the extraction of the previous data element.

2.1 Subtask 1: Sentence Classification

The first part of this task was to extract sentences that reflected the research contribution in the given NLP scholarly articles. We termed this sentence classification (Dao et al., 2020), where we predicted whether a sentence in an article was a contribution sentence. To this end, our approach was to pass each sentence in an article through the pre-trained BERT model to generate 768-dimensional word embeddings for each word in the sentence. The next thing we were going to do was to take the word embeddings of the first token of each sentence

[1] https://github.com/maxinge8698/SemEval2021-Task11

(i.e. '[CLS]') to do sentence classification because it integrated the semantic information of the whole sentence. Then this word embeddings acquired from the previous step was connected with a fully connected layer that converted the 768-dimensional input into 2-dimensional numerical values. These values were then input into *softmax* to calculate the probability of a sentence being a contribution sentence. Finally, the probability outcomes were input into *argmax*, where, in our experimental setup, an output of 1 indicated a contribution sentence and 0 indicated the contrary. The overall architecture of the system is shown in Figure 2.

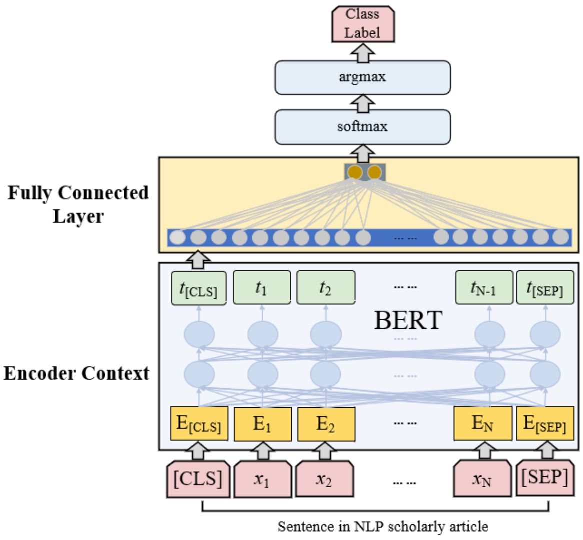

Class Label
argmax
softmax
Fully Connected Layer
Encoder Context
BERT
$t_{[CLS]}$ t_1 t_2 t_{N-1} $t_{[SEP]}$
$E_{[CLS]}$ E_1 E_2 E_N $E_{[SEP]}$
[CLS] x_1 x_2 x_N [SEP]
Sentence in NLP scholarly article

Figure 2: System of binary classification for sentence classification task

2.2 Subtask 2: Span Identification

Span identification (Singh et al., 2020) was a binary sequence tagging task where we classified each token in a contribution sentence to indicate whether it was part of a scientific term or predicate phrase fragment. We passed the contribution sentence identified from Subtask 1 into a pre-trained BERT model and obtained embeddings for each token in the sequence. Next, the word embeddings for each token were passed through a fully connected layer, and thereafter through *softmax* and *argmax*, where they were mapped to a class label respectively except the tokens of '[CLS]' and '[SEP]', indicating whether the token was part of a scientific term or predicate phrase fragment. The model architecture is illustrated in Figure 3. Note that the fully connected layer, *softmax*, and *argmax* were shared across all tokens.

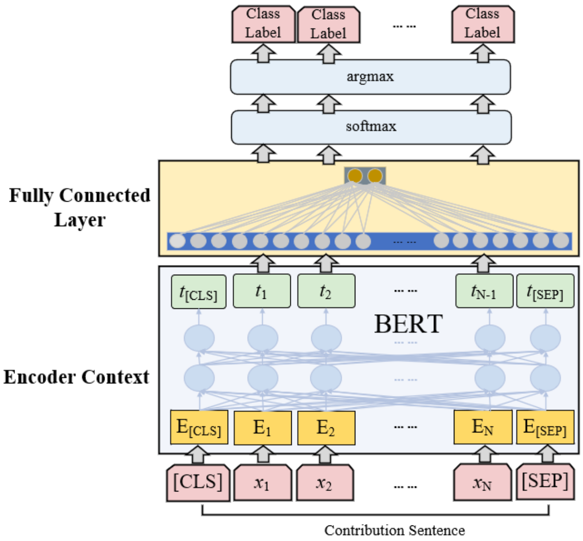
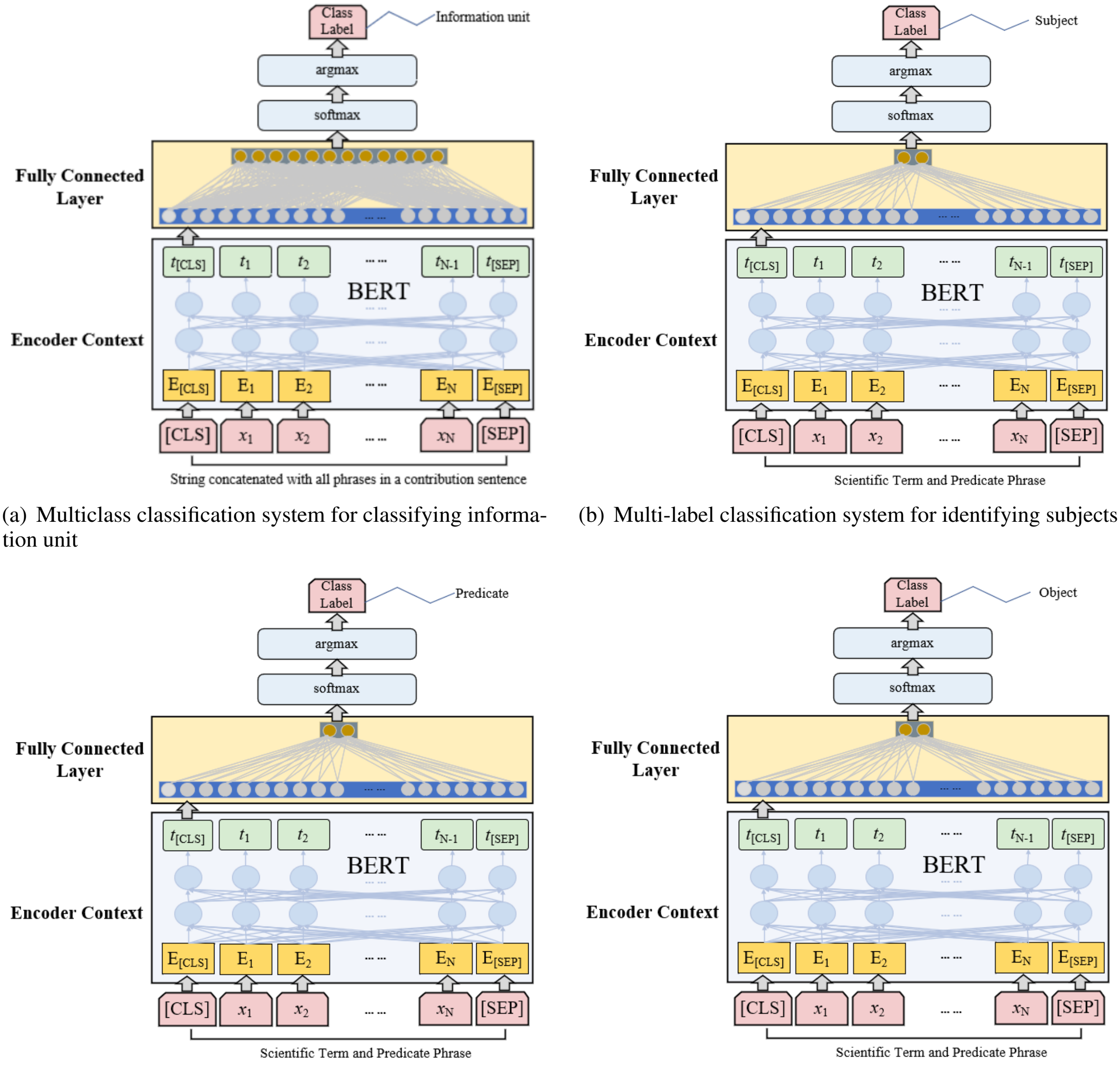

Class Label
Class Label
Class Label
argmax
softmax
Fully Connected Layer
Encoder Context
BERT
$t_{[CLS]}$ t_1 t_2 t_{N-1} $t_{[SEP]}$
$E_{[CLS]}$ E_1 E_2 E_N $E_{[SEP]}$
[CLS] x_1 x_2 x_N [SEP]
Contribution Sentence

Figure 3: System of sequence labeling for span identification task

2.3 Subtask 3: Triple Extraction

This subtask was the most cardinal and complex step in the entire task. This could be considered a relation extraction task (Lin et al., 2016), which was completed by dividing into parts information units classification and triple formation. First, it was necessary to classify all scientific term and predicate phrases in a contribution sentence extracted from Subtask 2 to determine which category of the 12 information units the extracted phrases belonged to. This was a multiclass-sequence classification problem, where we identified the unit information belonging to the scientific term and predicate phrase fragments in a given contribution sentence by concatenating all phrases into a single string to feed our model. The system architecture of this part was similar to the sentence classification system, except that there were 12 class labels and 12 output dimensions instead of 2 each (refer to Figure 4a). The next step was to identify the subject, predicate, and object in the scientific term and predicate phrases included in a contribution sentence obtained from Subtask 2 by using multi-label classification. More specifically, each scientific term and predicate phrase could be labeled with one or more of the three tags of subject, predicate and object, which could be solved by transforming the multi-label classification problem into three binary classification problems similar to the sentence classification system whereas using each phrase as input instead of each sentence. First, a binary classification system was used to


Class Label
Information unit
argmax
softmax
Fully Connected Layer
Encoder Context
BERT
$t_{[CLS]}$ t_1 t_2 …… t_{N-1} $t_{[SEP]}$
$E_{[CLS]}$ E_1 E_2 …… E_N $E_{[SEP]}$
[CLS] x_1 x_2 …… x_N [SEP]
String concatenated with all phrases in a contribution sentence
Subject
Scientific Term and Predicate Phrase
Predicate
Object

(a) Multiclass classification system for classifying information unit

(b) Multi-label classification system for identifying subjects

(c) Multi-label classification system for identifying predicates

(d) Multi-label classification system for identifying objects

Figure 4: System of multiclass classification and multi-label classification for triple extraction task

identify the subject in each scientific term and predicate phrase (refer to Figure 4b). Then, a second binary classification system was used to identify the predicate (refer to Figure 4c), and a third binary classification system was used to identify the object (refer to Figure 4d). A brief overview of this system is presented in Figure 4. At the end of the classification, for all phrases in a contribution sentence, that the corresponding label conformed to the subject-predicate-object order was found as triples iteratively from the beginning position.

3 Experimental Results

Datasets. The *NLPContributionGraph* shared task comprises a dataset of NLP scholarly articles with annotated contributions. The annotations were provided in terms of three data elements: (1) contribution sentences, (2) scientific term and predicate phrases from the sentences, and (3) (subject, predicate, object) triple statements. All the triples together formed the contribution-centered knowledge graph of the articles. The dataset released by the organizers contained 237 annotated articles as training data and 155 annotated articles as testing data for the final evaluation phases. For the training data, the annotations of each scholarly article were provided in a directory. The directory contained the full article in plain text, which was pre-processed for tokenization and sentence splitting. The annotations were provided in the following three files: (1) **sentence.txt**, specifying the annotated contribution sentence numbers from the plain text file; (2) **entities.txt**, specifying the sentence number, tab-separated from the start and end token numbers of the annotated phrase in the sentence; and (3) a directory **triples/** containing files with triples of scientific term and phrase pairs and a relation cue phrase, and the files were named to indicate the information unit that the triple data represented. For the article under the directory “training-data/natural_language_inference/0” as illustrated in Table 1, the sentence.txt file gave the line index of the articles contribution sentences (starting at 1). As illustrated in Table 2, the entities.txt file presented the line index which was identical to sentence.txt file, beginning position (starting at 0), end position, and corresponding text content of the scientific term and predicate phrases for each of the contribution sentences of the article. As illustrated in Table 3, the triples folder contained files named as one of the 12 information units covered in the article, and each information unit file provided (subject, predicate, object) triples that were comprised of the scientific term and predicate phrases.

Evaluation Metrics. An *NLPContributionGraph* submission would be considered complete with predictions made for all three tasks (sentences, phrases, triples). The evaluation metrics that were applied are

- Sentences: *precision*, *recall* and F_1-*score*;
- Phrases: *precision*, *recall* and F_1-*score*;
- Triples: *precision*, *recall* and F_1-*score* overall and for each information unit.

The calculation of these three evaluation metrics is as follows:

$$precision = \frac{true\ positives}{true\ positives + false\ positives} \tag{1}$$

$$recall = \frac{true\ positives}{true\ positives + false\ negatives} \tag{2}$$

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{3}$$

For the final evaluation stage of the task, the evaluation metrics is as follows:

$$\begin{aligned} F_1 = avg(&F_1(Sentences), F_1(Phrases),\\ &F_1(InformationUnits), F_1(Triples)) \end{aligned} \tag{4}$$

Implementation Details. The articles were split into sentences to feed into the language model, which resulted in 55,201 sentences in the training data (44,160 sentences served as the training set and 11,041 as the development set) with a maximum length of 398 words, and 33,800 sentences in testing data that originated from the evaluation phase, with a maximum length of 385 words. We used the Tensorflow framework provided by the Huggingface[2] library for the pre-trained BERT models and `bert-base-uncased` for binary classification, sequence labeling, multiclass classification, and multi-label classification included in this task. In addition, we fine-tuned the model using the Adam optimizer (Loshchilov and Hutter, 2018), by using a loss function of categorical cross-entropy with a learning rate of 2×10^{-5} and a batch size of 8 for three epochs. The activation function used by the fully connected layer was *softmax*.

[2] https://huggingface.co

article_directory	sentences
training-data/natural_language_inference/0	2
	11
	13

Table 1: Part of sentences.txt corresponded to an article

article_directory	sentences	begin_offset	end_offset	text
training-data/natural_language_inference/0	2	30	48	Text Comprehension
	11	37	75	https://github.com/bdhingra/ga-reader
	13	43	58	machine reading

Table 2: Part of entities.txt corresponded to an article

article_directory	research-problem.txt	code.txt
training-data/natural_language_inference/0	(Contribution||has research problem||Text Comprehension) (Contribution||has research problem||machine reading)	(Contribution||Code||https://github.com/bdhingra/ga-reader)

Table 3: Part of triples corresponded to an article

Result and Discussion. To allow a thorough evaluation of the systems, *NLPContributionGraph* was be organized into three evaluation phases:

- **Evaluation Phase 1: End-to-end pipeline testing phase.** The participant systems were expected to output contribution sentences, their corresponding scientific terms, and predicate phrases as well as triples.

- **Evaluation Phase 2, Part 1: Phrases extraction testing.** The participant systems were given gold-annotated contribution sentences and were expected to provide purely scientific terms and predicate phrases as well as triples as extraction output.

- **Evaluation Phase 2, Part 2: Triples extraction testing.** The participant systems were given gold phrases and were expected to provide triples as the only output.

We used the Scikit-Learn[3] library to divide the training data into training and development sets in a 8:2 ratio. We trained our models on the training set and evaluated the prediction with the golden scores of the good performance of our approaches. For these three subtasks, the F_1-*score*, *Precision*, and *Recall* of our system on the development set are shown in Table 4.

Our system achieved an average F_1-*score* of 0.4562 in Evaluation Phase 2, Part 1 and ranked 7th among the participating systems, and an average F_1-*score* of 0.6541 in Evaluation Phase 2, Part 2 and also ranked 7th among all participants. The results showed that our proposed system was effective in extracting contributions from an NLP scholarly article. The main reason was that the BERT model is a multi-layer bidirectional transformer encoder, which can be integrated into various NLP downstream tasks and achieves the best results.

Subtask	F_1-*score*	*Precision*	*Recall*
Subtask1	0.6423	0.6554	0.6932
Subtask2	0.4768	0.5326	0.4356
Subtask3	0.4385	0.4109	0.6151

Table 4: Score of the pre-trained BERT model for the three subtasks on the development set

4 Conclusions

In this paper, we presented the system we submitted to the SemEval-2021 Task 11, which leveraged a pre-trained BERT model to extract contributions from an NLP scholarly article using binary classification, sequence labeling, multiclass classification, and multi-label classification. The experimental results showed that the proposed models achieved a good performance in the final evaluation phases.

[3] https://scikit-learn.org

Furthermore, in the three subtasks, there appeared to be significant room for improvement compared to the top-ranked participant systems. Therefore, in future research, we will attempt to generalize models with better capabilities to obtain better results.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (NSFC) under Grants Nos. 61702443, 61966038 and 61762091.

References

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*.

Jiaxu Dao, Jin Wang, and Xuejie Zhang. 2020. YNU-HPCC at SemEval-2020 task 11: LSTM network for detection of propaganda techniques in news articles.

Jacob Devlin, Ming Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv*, pages 4171–4186.

Jennifer D'Souza and Sören Auer. 2020. NLPContributions: An Annotation Scheme for Machine Reading of Scholarly Contributions in Natural Language Processing Literature. *arXiv*, pages 16–27.

Jennifer D'Souza, Sören Auer, and Ted Pedersen. 2021. SemEval-2021 Task 11: NLPContributionGraph - Structuring Scholarly NLP Contributions for a Research Knowledge Graph. In *Proceedings of the Fifteenth Workshop on Semantic Evaluation*, Bangkok (online). Association for Computational Linguistics.

Mohamad Yaser Jaradeh, Allard Oelen, Kheir Eddine Farfar, Manuel Prinz, Jennifer D'Souza, Gabor Kismihok, Markus Stocker, and Sören Auer. 2019. Open research knowledge graph: Next generation infrastructure for semantic scholarly knowledge. *arXiv*, pages 243–246.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. *15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017 - Proceedings of Conference*, 2:427–431.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, pages 1746–1751.

Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. Neural relation extraction with selective attention over instances. *54th Annual Meeting of the Association for Computational Linguistics, ACL 2016 - Long Papers*, 4:2124–2133.

Ilya Loshchilov and Frank Hutter. 2018. Fixing Weight Decay Regularization in Adam. Technical report.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. *54th Annual Meeting of the Association for Computational Linguistics, ACL 2016 - Long Papers*, 2:1064–1074.

Paramansh Singh, Siraj Sandhu, Subham Kumar, and Ashutosh Modi. 2020. newsSweeper at SemEval-2020 task 11: Context-aware rich feature representations for propaganda classification. *arXiv*.

Shuohang Wang and Jing Jiang. 2016. Learning natural language inference with LSTM. *2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2016 - Proceedings of the Conference*, pages 1442–1451.

Zhi Xiu Ye and Zhen Hua Ling. 2018. Hybrid semi-markov CRF for neural sequence labeling. *arXiv*, pages 235–240.

Li Yuan, Jin Wang, Liang-Chih Yu, and Xuejie Zhang. 2020. Graph Attention Network with Memory Fusion for Aspect-level Sentiment Analysis. *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 27–36.

GuoDong Zhou and Jian Su. 2001. Named entity recognition using an HMM-based chunk tagger. (July):473.