YNUtaoxin at SemEval-2020 Task 11: Identification Fragments of Propaganda Technique by Neural Sequence Labeling Models with Different Tagging Schemes and Pre-trained Language Model

Xin Tao, Xiaobing Zhou*
School of Information Science and Engineering
Yunnan University, Yunnan, P.R. China
*Corresponding author, zhouxb@ynu.edu.cn

Abstract

Information extraction is a hot topic in NLP, and detecting the use of propaganda techniques in news articles is part of this kind of task. This paper describes the solution of the Span Identification subtask in the Semeval 2020 Task 11: Detection of Propaganda Techniques in News Articles. The core idea of our method is equivalent to regard this task as a sequence tagging task and develop a neural sequence model to solve it. We use three different tagging schemes to tag sentences. Some pre-trained language models are used as the feature encoder like BERT, RoBERTa, and XLNet. In the final evaluation, we achieve the F1-score of 0.43208 and rank 11th among all the submitted teams.

1 Introduction

Propaganda is a form of information promoting or publicizing a particular cause or point of view, especially one of a biased or misleading nature. Propaganda usually uses psychological and rhetorical techniques to meet its purpose. In today's information age, propaganda is widely used in politics, business, science and technology, and other fields. And there are increasing propaganda techniques that are becoming more diverse and difficult to identify (Glowacki et al., 2018). We can search for the use of specific propaganda techniques via analyzing the language of an article, which is useful to find out the aims of propagandist news articles.

In the mass of propagandist news articles, how to quickly and accurately identify the use of propaganda techniques is a challenging task. SemEval-2020 Task 11 - Detection of Propaganda Techniques in News Articles aims to develop automatic tools to detect propaganda techniques. And there are two subtasks:

1. Span Identification (SI): Given a plain-text document, identify those specific fragments which contain at least one propaganda technique.

2. Technique Classification (TC): Given a text fragment identified as propaganda and its document context, identify the applied propaganda technique in the fragment.

The shared task provides a news article corpus, as well as 18 predefined propaganda techniques that have been annotated correspondingly. More detailed information can be seen in (Da San Martino et al., 2020).

The rest of the paper is divided into the following sections: Section 2 describes the previous works and the relationship between coarse-grained and fine-grained tasks. Section 3 reports the final approach and those we tried . Section 4 details the results for the experimented models followed by error analysis of the best model.

2 Related Work

Some previous works have proposed various methods for propaganda detection. LSTM is used for propaganda detection (Rashkin et al., 2017). Barrón-Cedeño et al. (2019) used the Maximum Entropy classifier with different features to deal with the same work of (Rashkin et al., 2017). However, their works

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 1875–1880
Barcelona, Spain (Online), December 12, 2020.

are all at the document level, which labels the whole article as propaganda. More fine-grained propaganda analysis has been proposed by (Da San Martino et al., 2019), which avoids noisy gold labels affect the quality of model training and also makes it possible to illustrate to the user why an article was judged as propagandistic. This is also the case in other areas, where more fine-grained research that can increase performance and interpretability on coarse-grained tasks has quite a few advantages for further follow-up studies. Wang et al. (2016) presented a community-based weighted graph model to predict valence-arousal ratings of affective words. Future research can benefit from such useful lexical resources to extend current valence-arousal prediction work from the word-level to sentence-levels and document-levels.

We treat this SI subtask as a sequence tagging task, and previous jobs have been handled in this way. Da San Martino et al. (2019) proposed a Multi-Granularity neural network model based on BERT to deal with both sentence-level classification and fragment-level classification. Tayyar Madabushi et al. (2019) presented a method of cost-sensitivity BERT to address sentence-level classification in imbalanced data and provide a measure of corpus similarity to determine the difference between the training and the development or test sets.

3 Methodology

We only participate in the SI subtask, and our approach is a neural sequence labeling model along with different tag schemes. Our model used some large pre-trained language model as the encoder, and other network structures as the output of the encoder.

3.1 Dataset

The dataset of the SI task used in all our experiments is provided by the organizer and no external datasets are used. Detailed information about the dataset can be found in (Da San Martino et al., 2020). The dataset is divided into training, development, and test set, and the distribution of data is shown in Table 1.

Dataset	Article num	Sentence num
Train	371(69.2%)	16673(72.4%)
Dev	75(14.0%)	3177(13.8%)
Test	90(16.8%)	3185(13.8%)
Total	536	23035

Table 1: The distribution of datasets.

3.2 Tagging Schemes

Since we formulate the task as a sequential labeling problem, and this task is similar to Named Entity Recognition (NER) that refers to identifying entities with specific meanings in the text, mainly including the names of people, places, institutions, proper nouns and so on. We use the IOB format (Inside, Outside, Beginning) to represent sentences where every token is labeled as *B-propaganda* if the token is the beginning of a propaganda technique, *I-propaganda* if it is inside a propaganda technique but not the first token, or O otherwise. Further, we also use the IOBES tagging scheme (Dai et al., 2015), a variant of IOB format, which encodes the propaganda technique for a single token as *S-propaganda* and marks the last token as *E-propaganda*. With this scheme, a token is marked *I-propaganda* and the selection of subsequent token is narrowed down to *I-propaganda* or *E-propaganda* with high confidence. However, the IOB scheme only determines that the subsequent token cannot be the interior of another label. In this task, we just need to identify the fragments of the propaganda technique, rather than the category of it, so that a simpler binary tagging scheme can be adopted. In other words, this is a binary sequence tagging task. We use *Propaganda* to tag the span of propaganda technique and *No-propaganda* for the others.

3.3 Pre-trained Language Model

We use some pre-trained language models as encoders for input sentences. The pre-trained models we tried to use included BERT, RoBERTa, and XLNet.

- BERT (Devlin et al., 2018) is a pre-trained model that is designed to train deep bidirectional representations based on Mask Language Model and Next Sentence Prediction. The pre-trained BERT model can be fine-tuned with one additional output layer to achieve good performance for a few NLP tasks.

- RoBERTa (Liu et al., 2019) is an improvement by BERT, which uses more training data and expands the number of model parameters. And the improvements of the RoBERTa in training approaches include using a dynamic Masking Language Model, removing Next Sentence Prediction, using Byte-pair Encoding, and utilizing large mini-batches.

- XLNet (Yang et al., 2019) used Permutation Language Model that not only learns the dependency between tokens but also allows models to capture bidirectional contexts. And the XLNet also integrates ideas from Transformer-XL including the relative positional encoding scheme and the segment recurrence mechanism into pre-training.

4 Experiment and Result

Our experimental implementation is based on the PyTorch framework and uses the transformers package (Wolf et al., 2019) to get the pre-trained language model. In the data preprocessing stage, the Spacy (Honnibal and Montani, 2017) is used for tokenization and tagging. And our experimental codes can be found on GitHub.[1]

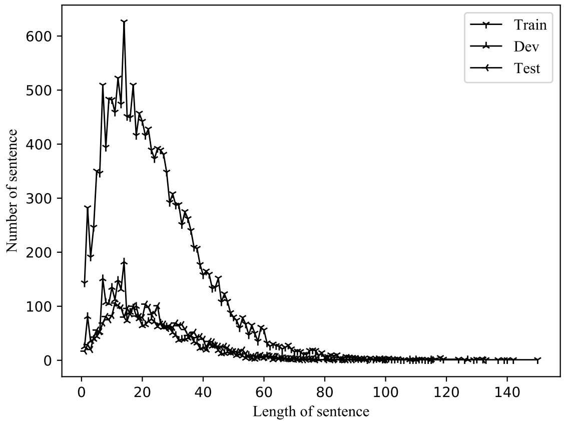

Figure 1: The sentence length distribution of training set, development set, and test set

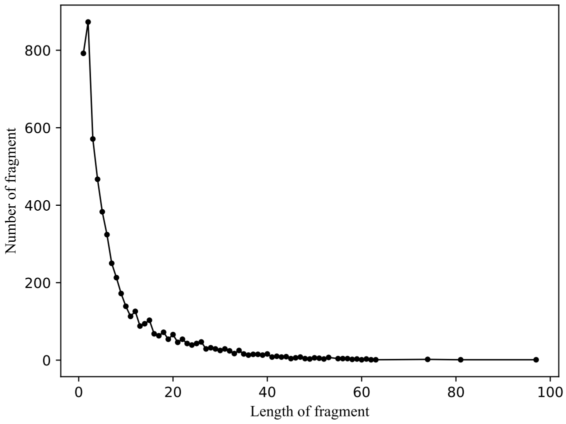

Figure 2: The distribution of the span of propaganda technique length of the training set

In the data preprocessing stage, we clean the text and remove the special symbols. The number of words in all sentences is analyzed, as shown in Figure 1, and the maximum length is 150. From the figure, we can find that sentences less than 80 tokens in length are 99.15%, and less than 100 tokens are 99.72% of the total. In the experiment, we set the sentence length as 80,100 and 150 respectively, and find that the sentence length as 100 is the best. The SI is a token-level task, which is supposed to preserve all tokens in the sentence. But the too big sentences length setting will result in a lot of irrelevant padding in short sentences, which will create a lot of data noise and affect the performance of the model. The Bert model is used to evaluate the sentence length parameters. The experimental results are displayed in Table 2.

A few discontinuous *I-propaganda* labels appear in the predictions when the IOB tagging scheme is used, which may be caused by the large length of the span of propaganda technique. We calculate the length of the fragment of the propaganda technique, as shown in Figure 2, and the mean length is 8.8203 tokens. The fragments are longer than the usually named entity. To better enable the model to recognize the boundaries of the fragments, the IOBES tagging scheme is used. According to the experimental

[1] https://github.com/taoxin778/Propaganda-Detection-2020

Experiment	F1-score	Precision	Recall
IOB + 80	0.4098	0.37725	0.44851
IOB + 100	**0.4198**	**0.38492**	**0.46163**
IOB + 150	0.41233	0.37708	0.45484
IOBES + 80	0.4075	0.38086	0.43814
IOBES + 100	**0.43018**	**0.38697**	**0.48425**
IOBES + 150	0.42181	0.36714	0.4956
Binary + 80	0.40975	0.36059	0.47441
Binary + 100	**0.44353**	**0.40737**	**0.48674**
Binary + 150	0.42983	0.38859	0.48086

Table 2: Experimental results of three labeling schemes and three maximum sentence lengths on BERT pre-trained model

results, shown in Table 2, compared with the IOB tagging scheme, the IOBES does get a performance improvement. However, due to the large length of the propaganda fragment, the labeling method of NER task is difficult to accurately identify its boundaries because it was found in the experiment that the tag set did not form an accurate boundary of the propaganda fragment. We believe that the propaganda fragment has longer semantic dependence, while its syntactic dependence is not clear. Therefore, it may be better to use the tagging scheme without emphasizing the boundary of propaganda fragment. And we use the simple binary tagging scheme. The experiments show that the simpler binary tagging scheme achieves the best results.

After using Bert as the feature encoder, we attempt to use RoBERTa and XLNet as the pre-trained language model, and the decoder also use the Softmax layer. We fine-tune the learning-rate, min-batch-size and num-epochs hyperparameters, and obtain the results with three different models on development data set, as shown in Table 3. The BERT pre-trained model works the best on the development set, and we submit the final results on the test set using the same model parameters. In the final evaluation, we achieve an F1-score of 0.43208 and rank 11th among all the submitted teams. Finally, the hyperparameters used in our final submitted model are shown in Table 4.

Pre-train model	F1-score	Precision	Recall
RoBERTa	0.41704	0.4085	0.42595
XLNet	0.42345	0.39875	0.45141
BERT	**0.44353**	**0.40737**	**0.48674**

Table 3: Experimental results of three pre-trained model

Hyperparameters	Value
Learn rate	5e-5
Epoch num	10
Batch size	24
Sentence length	100
Random seed	37
BERT model	bert-base-uncased

Table 4: Hyperparameters settings of the final model

After analyzing all our results, we found that the recall rate was greater than the precision rate. We analyzed this phenomenon according to the evaluation methods provided by the organizers. For the SI tasks, its evaluation formula (Da et al., 2020) is as follows:

$$C(s,t,h) = \frac{|(s \cap t)|}{h} \tag{1}$$

$$P(S,T) = \frac{1}{|S|} \sum_{s \in S, t \in T} C(s,t,|s|) \tag{2}$$

$$R(S,T) = \frac{1}{|T|} \sum_{s \in S, t \in T} C(s,t,|t|) \tag{3}$$

$$F_1(S,T) = 2P(S,T)R(S,T)\frac{P(S,T)R(S,T)}{P(S,T)+R(S,T)} \tag{4}$$

where S is the set of predicted fragments $s, s \in S$, T is the set of true fragments $t, t \in T$, and h is a normalizing factor in equation 1. The precision and recall are defined as equation 2 and equation 3 respectively, and equation 4 gives the F1-score that is the harmonic mean of precision and recall. By comparing the predicted file with the gold file, we find that the number of predicted fragments is larger than the number of gold fragments. This results in a larger penalty factor for calculating precision, thus reducing precision. We think that this is because the model has misidentified some fragments, and the model tends to classify the fragments that can not be judged as propaganda fragments. For the misidentification of the model, likely, the decoder does not capture the long-distance dependence well. Another possible reason is that the data distribution is unbalanced, which will make the model learn the classes with more instances, and the classes with fewer instances will be misjudged.

5 Conclusion and Future Work

We focus on the span identification task and develop an automated system with three tagging schemes and pre-trained language models in this paper. We analyze the sentence length and fragment length of the data set in detail and use experimental methods to verify the impact of the tagging scheme on fragment recognition in sentences, which can serve as a reference for future research. We select different pre-trained language models as feature encoders and compare their results. However, we do not make too much effort into the design of the decoder, which can be a bottleneck for the whole system. We believe that using a complex decoder may further improve system performance, which is one of the future research directions.

At present, the sequence annotation task has three main types of decoders, namely Softmax, CRF, and seq2seq. Softmax can decode tags in parallel, but cannot model dependencies between tags, which results in poor performance. CRF decodes the whole sentence tag by learning the transfer matrix between tags, which can model the dependency of adjacent tags. However, using Viterbi decoding, CRF can not be parallel and the decoding speed is slow. Seq2seq can model the long-distance dependence of tags, but it has a problem that the false prediction of the previous tag will lead to the error of the subsequent tags. For the task of propaganda techniques detection, because the detection fragment is long, the model needs to be able to model the long-distance dependence. But now there is not much research on decoding, which will be a worthy research goal.

Acknowledgments

This work was supported by the Natural Science Foundations of China under Grants 61463050, the NSF of Yunnan Province under Grant 2015FB113.

References

Alberto Barrón-Cedeño, Israa Jaradat, Giovanni Da San Martino, and Preslav Nakov. 2019. Proppy: Organizing the news based on their propagandistic content. *Information Processing and Management*, 56(5):1849–1864.

Giovanni Da, San Martino, and Alberto Barr. 2020. Evaluation of Propaganda Detection Tasks. (section 1):1–2.

Giovanni Da San Martino, Seunghak Yu, Alberto Barrón-Cedeño, Rostislav Petrov, and Preslav Nakov. 2019. Fine-grained analysis of propaganda in news articles. 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 2019*, EMNLP-IJCNLP 2019, Hong Kong, China, November.

Giovanni Da San Martino, Alberto Barrón-Cedeño, Henning Wachsmuth, Rostislav Petrov, and Preslav Nakov. 2020. SemEval-2020 task 11: Detection of propaganda techniques in news articles. In *Proceedings of the 14th International Workshop on Semantic Evaluation*, SemEval 2020, Barcelona, Spain, September.

Hong Jie Dai, Po Ting Lai, Yung Chun Chang, and Richard Tzong Han Tsai. 2015. Enhancing of chemical compound and drug name recognition using representative tag scheme and fine-grained tokenization. *Journal of Cheminformatics*, 7(Suppl 1):1–10.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Monika Glowacki, Vidya Narayanan, Sam Maynard, Gustavo Hirsch, Bence Kollanyi, Lisa-Maria Neudert, Phil Howard, Thomas Lederer, and Vlad Barash. 2018. News and political information consumption in mexico: Mapping the 2018 mexican presidential election on twitter and facebook. *The Computational Propaganda Project*.

Matthew Honnibal and Ines Montani. 2017. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear*, 7(1).

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Hannah Rashkin, Eunsol Choi, Jin Yea Jang, Svitlana Volkova, and Yejin Choi. 2017. Truth of varying shades: Analyzing language in fake news and political fact-checking. *EMNLP 2017 - Conference on Empirical Methods in Natural Language Processing, Proceedings*, pages 2931–2937.

Harish Tayyar Madabushi, Elena Kochkina, and Michael Castelle. 2019. Cost-Sensitive BERT for Generalisable Sentence Classification on Imbalanced Data. pages 125–134.

Jin Wang, Liang Chih Yu, K. Robert Lai, and Xuejie Zhang. 2016. Community-Based Weighted Graph Model for Valence-Arousal Prediction of Affective Words. *IEEE/ACM Transactions on Audio Speech and Language Processing*, 24(11):1957–1968.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5754–5764.