syrapropa at SemEval-2020 Task 11: BERT-based Models Design For Propagandistic Technique and Span Detection

Jinfen Li
College of Arts and Sciences,
Syracuse University
jli284@syr.edu

Lu Xiao
School of Information Studies,
Syracuse University
lxiao04@syr.edu

Abstract

This paper describes the BERT-based models proposed for two subtasks in SemEval-2020 Task 11: *Detection of Propaganda Techniques in News Articles*. We first build the model for Span Identification (SI) based on SpanBERT, and facilitate the detection by a deeper model and a sentence-level representation. We then develop a hybrid model for the Technique Classification (TC). The hybrid model is composed of three submodels including two BERT models with different training methods, and a feature-based Logistic Regression model. We endeavor to deal with imbalanced dataset by adjusting cost function. We are in the seventh place in SI subtask (0.4711 of F1-measure), and in the third place in TC subtask (0.6783 of F1-measure) on the development set.

1 Introduction

Propaganda exists in social media content and can subvert and distort public deliberation (Farkas, 2018). Natural language processing (NLP) researchers develop computational techniques that automatically detect propaganda in the content.

In this paper, we develop two systems for two subtasks in the SemEval-2020 Task 11: (1) Span Identification (SI) (2) Technique Classification (TC) in News Articles respectively. The SI subtask focus on identifying fragments in a given plain-text document which contain at least one propaganda technique; while TC subtask aims to classify the applied propaganda technique given a propagandistic text fragment (Da San Martino et al., 2020).

For SI, we interpret the task as to detect a span in a context, which is the smallest detect unit. We base our system on SpanBERT (Joshi et al., 2020), and combines three jointly trained classifiers: sentence, start, and end classifiers. Specifically, the start and end classifiers detect the start and end position of the span, while the sentence classifiers provide sentence-level information for the start and end classifiers. For TC, we come up with a hybrid system combining the two BERT models with different training methods, a Logistic Regression model, and extra rules to classify a propagandistic fragment into one of the 14 techniques.

For SI, we find that the segment approach of a context affects the result. While for TC, emotion features extracted from NRC lexicon are not effective to distinguish 14 classes. However, features such as text length, TF-IDF, occur times in a document, superlative form, question words, hashtags and supplement are useful in distinguishing different propaganda techniques. Overall, we are in the seventh place in SI subtask (0.4711 of F1-measure), and in the third place in TC subtask (0.6783 of F1-measure) on the development set.

2 Related Work

It is believed that news media plays an active and major role in producing and distributing propaganda and there can be a tool box that helps detect propaganda in news articles (Zollmann, 2019). Da San Martino

This work is licensed under a Creative Commons Attribution 4.0 International License. License details: http://creativecommons.org/licenses/by/4.0/.

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 1808–1816
Barcelona, Spain (Online), December 12, 2020.

et al. (2019a) annotated 18 propaganda techniques in news articles. The researchers later merged the similar underrepresented techniques, resulting in 14 techniques (Da San Martino et al., 2020).

The model trained on the sentence level task (whether the input sentence is propagandistic) is found to be effective for the fragment level task (detect the propagandistic fragment). Gupta et al. (2019) designed a multi-granularity and multi-tasking neural architectures to jointly perform both the sentence and fragment level propaganda detection. Da San Martino et al. (2019b) designed a multi-granularity neural network that includes document-level, paragraph-level, sentence-level, word-level, subword-level and character-level task to detect fragments in news articles. Different from their studies, our SI model utilizes a sentence-level classifier which predicts whether a context contains a span or not, and embeds the hidden representations from it into the other two classifiers. We also explore different approach of concatenation of other higher representations.

Previous researches explore the effectiveness of hybrid model. For instance, Al-Omari et al. (2019) use several submodels including BiLSTM, XGBoost, BERT model to predict if a sentence is propagandistic. Our TC model is also an hybrid model of three submodels (BERT, cost_BERT, LR†‡) that combine the partial results based on their learning capacity of different categories. We compare the unweighted and weighted cost function in our approach and find out that the latter outperforms on the minority classes such as *Whataboutism/Straw Men/Red Herring, Thought-terminating Cliches and Bandwagon/Reductio ad hitlerum*.

3 SI System Description

3.1 Context Segmentation

In the news article dataset provided by Da San Martino et al. (2020), the start character index and end character index of each span is annotated in a news article. A span may be part of a sentence or include up to five sentences. The segmentation of the context is essential in the SI task as our goal is to detect a span in one context. In order to detect as many spans as possible in one news article, we first split the article into sentences. We then merge the overlapped spans into one span, for the reason that a longer span improves the recall when we only predict one span in a context. We define two different contexts using the following strategies:

1. **Mini Context** For one sentence,
   a) if there are multiple merged spans: we extend the context from the span (including the merged one) to the left and right side until it meets the boundary of other spans. As shown in Figure 1, for sentence 1, span 3 combining span 1 and 2, is within context 1; while span 4 is within context 2.
   b) if there is no span or one span in a sentence: then the sentence itself is one context.

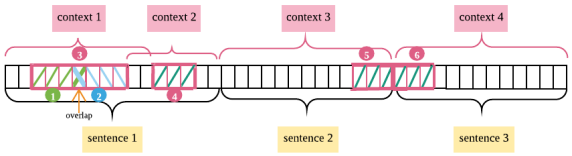

Figure 1: Context segmentation. The red color represents the merged spans and the green and blue colors represents the unmerged spans.

   In cases where a span covers across several sentences, it is split into multiple shorter spans at the sentence boundary. As shown in Figure 1, the long span across sentence 2 and sentence 3 is splitted into span 5 and 6. The advantage to segment in this way is that it includes all the spans in training dataset and it has least noise in a context since there is only one span in a context, and the context is not too long for training. However, some contexts sacrifices their semantics integrity as they are just part of one sentence. We refer to it as "mini context".

2. **Sentential Context** For the sentences containing multiple spans, we only keep the longest one and ignore the others. We refer to it as "sentential context".

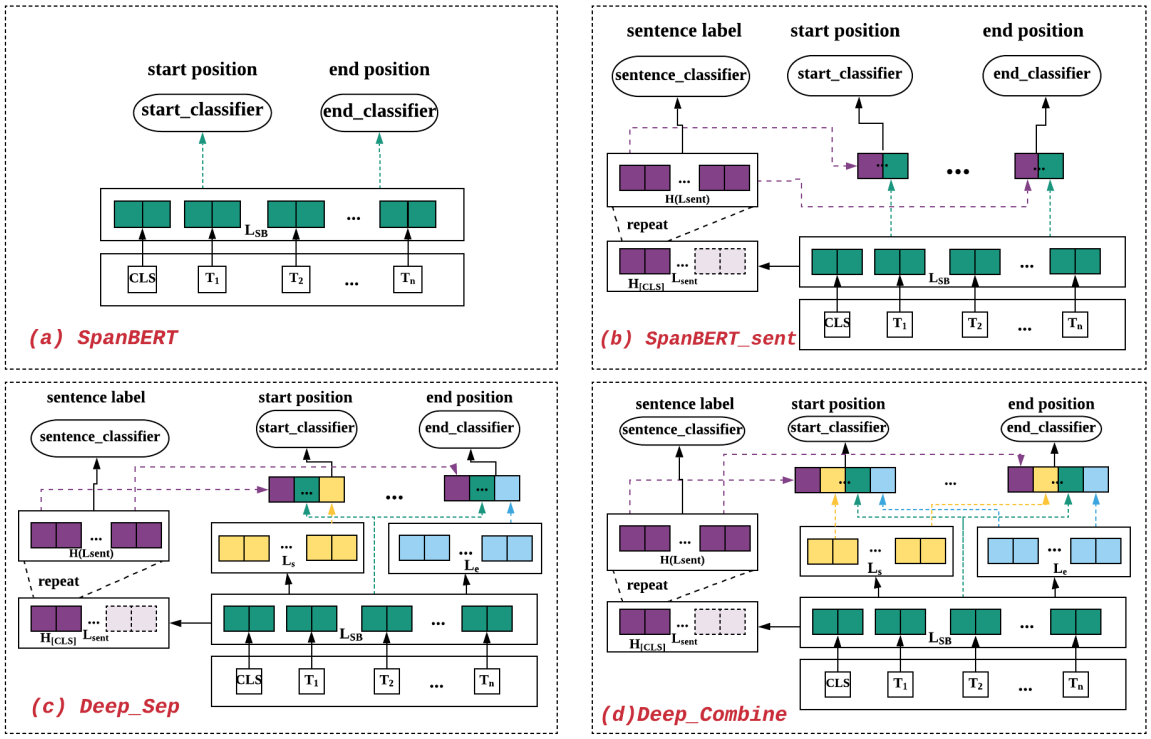

Figure 2: The architecture of the SpanBert models (a), and of our proposed modification networks (b-d).

While for the development and test dataset, we detect the spans in each sentence.

3.2 Our Model

We based our model on the pre-trained *SpanBERT_base* model and modify the top layers. The overall architectures of SpanBERT is shown in Figure 2-a, and our proposed models are shown in 2-(b-d).

3.2.1 Start and End Classifier

We add two separate linear layers L_s^o, L_e^o on top of SpanBERT and fine-tune it. The L_s^o layer outputs the probability of each token being the start boundary of the span and the L_e^o layer outputs that of each token being the end boundary of the span.

3.2.2 Sentence Classifier

The sentence classifier aims to classify whether a context contains a span, and the context is propagandistic if so. We come up with a layer L_{sent}^k to capture sentence-level feature. As shown in Figure 2-b, after feeding the hidden representations from the last layer of SpanBERT to L_{sent}^k, we only keep the feature of the first token "[CLS]", $H_{[CLS]}$ as it represents the whole context. We repeat $H_{[CLS]}$ to the number of tokens in the context and concatenate them to the hidden representation of each token. In addition, we feed $H(L_{sent}^k)$ into the output layer L_{sent}^o for binary classification.

3.2.3 Concatenation Layer

As shown in Figure 2-c (***Deep_Sep***), the output layer of the above three classifiers directly accepts the SpanBERT hidden representation, we deepen our model by adding the layers L_s^{k+1},L_e^{k+1} to L_s^k,L_e^k, respectively, which extract a higher-level representation. In addition, we keep the residual connection from SpanBERT hidden layer to the deepened layers, i.e., L_s^o and L_e^o. Therefore, L_s^o and L_e^o accepts concatenated representations from SpanBERT, L_{sent}^k and L^{k+1}. The two separate concatenated representations is shown in Equation 1 and the overview of architecture is shown in 2-c.

$$\begin{aligned} H_s &= [H_{SpanBERT}; H(L_s^{k+1}); H(L_{sent}^k)], \\ H_e &= [H_{SpanBERT}; H(L_e^{k+1}); H(L_{sent}^k)] \end{aligned} \tag{1}$$

As shown in Figure 2-d (***Deep_Combine***), the above concatenation layer generates two separate representations to feed into start and end classifiers respectively. We find that in a news article, it is more likely that one start boundary maps only one end boundary and vice versa. In order for both start classifier and end classifiers to adopt information from each other, we concatenate the output of L_s^{k+1} and L_e^{k+1}, together with the hidden representations from SpanBERT and L_{sent}^k, i.e., $H_{combine}$ in Equation 2. $H_{combine}$ is then fed into start and end classifiers. The architecture is shown in Figure 2-d.

$$H_{combine} = [H_{SpanBERT}; H(L_s^{k+1}); H(L_e^{k+1}); H(L_{sent}^k)] \tag{2}$$

3.2.4 Our Loss Function

We assign weights to different class (***cls_weighted***): we jointly train sentence, start and end classifiers in our model. The objective of the sentence classifier is the binary cross entropy loss.

$$L_{sent}(y, \hat{y}) = -\sum_{i}^{N} y_i \log(\hat{y}_i) \tag{3}$$

As for our start and end classifiers, we adopt the multi-class cross entropy loss function. Because the proportion of the context with span (minority class) and without span (majority class) is imbalanced, we assign a weight to the minority class. The equation is give in Equation 4, where w denotes the weight. Considering different convergence speed of the loss of three classifiers, we design the total loss function as Equation 5. We combine the results of the best start index and the best end index, i.e., $span(I_s, I_e|I_s)$ and $span(I_s|I_e, I_e)$ in the prediction process, where I is the boundary index.

$$L_{start/end}(y, \hat{y}) = \begin{cases} -\sum_{i}^{N} y_i \log(\hat{y}_i), y_{sent} = 0 \\ -\sum_{i}^{N} y_i \log(\mathbf{w} \cdot \hat{y}_i), y_{sent} = 1 \end{cases} \tag{4}$$

$$L_{total} = \alpha_{sent} L_{sent} + \alpha_{start} L_{start} + \alpha_{end} L_{end} \tag{5}$$

4 TC System Description

Our system includes three individual sub-models combined together with extra rules, which outperforms any of the sub-models itself.

4.1 Polymorphic BERT

Bidirectional Encoder Representation Transformer (BERT) is a model based on the bidirectional transformer to embed more context information from left to right and from right to left. In order to incorporate features of emotion, we come up with the emotion representations and concatenate them with the default representation in BERT. To deal with the imbalanced dataset, we explore the performance of assigning different weights to different classes in the cost function.

4.1.1 Embedding Emotion Feature

Emotional appeal is an important strategy used in propaganda techniques (Da San Martino et al., 2020). (Li et al., 2019) found that features of emotions can be good indicators of a propagandistic and non-propagandistic fragment in news articles. We explore whether emotion features can help in the identification of the type of propaganda technique in the fragment.

We use emotion lexicon NRC Affect Intensity Lexicon provided by (Mohammad and Bravo-Marquez, 2017), which contains the affect intensity of words in categories such as anger, disgust, joy, negative, positive, etc. In order to utilize the pretrained uncased BERT-Base model, of which the hidden size is 768 (Devlin et al., 2018), we come up with an emotion embedding table E_e with the same hidden size, of which each row is randomly initialized except that the first ten values are the affect intensity score in the lexicon. In other words, we add the emotion representation to BERT (i.e., **emo_BERT**) that typically includes three types of embedding: word embedding, position embedding, and segment embedding.

4.1.2 Solving Class Imbalance

Because the training corpus is highly imbalanced (please check Appendix), we adjust the cost function (i.e., **cost_BERT**). We first use the cross-entropy loss where x is the softmax output and y is the onehot encoding of the label and the t^{th} element is the target class y_t (Equation is shown in 6). We then multiple the cross-entropy loss with the weight of the target class w_{y_t}, where w_{y_t} is the reciprocal of the frequency of y_t in training dataset (Equation is shown in 7). With the modification of the cost function, the model will punish more on the mislabeled minority class (intuitively a more "important" class), such as *Bandwagon,Reductio ad hitlerum, Thought-terminating,Cliches* and *Whataboutism,Straw Men,Red Herring*.

$$loss(x, y) = -\log(\frac{\exp(x_{y_t})}{\sum_j \exp(x_{y_j})}) \tag{6}$$

$$weight\_loss(x, y) = w_{y_t} \cdot loss(x, y), \quad \text{where} \quad w_{y_t} = \frac{\sum_j c_{y_j}}{c_{y_t}} \tag{7}$$

4.2 Sub-Model:LR†‡

Our hybrid model combines the partial results generated by the sub-models including the typical BERT (**BERT**), the BERT trained with the cost-weighted function (**cost_BERT**) and the Logistic Regression (**LR**†‡) introduced in this section. We extract two continuous features:**Length, TF-IDF**; and several Boolean features including **Repetition, Superlative, Whatabout, Doubt, Slogan** and **Supplement**.

- **Length** We found that the text length of most fragments in some categories tend to be shorter than those in the other categories. We use the text length as our baseline feature.
- **TF-IDF** We use TF-IDF values (Jones, 2004) to enrich the dimension of features.
- **Repetition** The Repetition feature is a Boolean feature and is True if the fragment occurs more than four times in an article.
- **Superlative** The technique of *Exaggeration,Minimisation* utilize words in superlative format (e.g., "largest", "best", "greatest") to exaggerate or minimize some facts. The Superlative feature is a Boolean feature and is True if the fragment contains words in superlative form.
- **Whatabout** Just as the name of *Whataboutism* tells, we detect whether the fragment starts with phrase "what about".
- **Doubt** Fragements that use *Doubt* technique are likely to start with auxiliary words (e.g., has, is, do) or modals (e.g., can) or question words (e.g., why, what). With this Boolean feature we consider the signal in the classification.
- **Slogan** Fragments in *Slogan* class contains words that start with hashtags (e.g., #NeverAgain, #StopTheSynod), or start with "we will" (e.g., "we will serve the Lord"). The TRUE value of this Boolean feature means that the input fragment starts with a hashtag or "we will".
- **Supplement** *Red Herring* technique introduces material, irrelevant to the focal issue, so as to divert people's attention away from the points made (Da San Martino et al., 2019a). Some fragments are encompassed by a pair of brackets like "(who Kennedy admired)", "(Faber was nominated by President George H.W.Bush.)", acting as a supplement. Some fragments use the "who clause" such as "who is ...". We view these linguistic expressions as a supplement to the sentence. This feature represents whether the the fragment is a supplement.

4.3 Rule-based Correction and Reinforcement

After the prediction by the hybrid model, we apply simple syntactic rules to correct the mislabeled instances. Specifically, we compile rules based on part of speech tag (aka., POS tag) as follows.

For a fragment that is predictd to contain *Repetition* but its occurrences is less than three times in the article, if its POS tag sequence contains ('NN','NN') or ('NN','NNS') or ('NNS'), it is corrected as *Name Calling,Labeling*; if it is ('JJ') or ('NN'), it is corrected as *Loaded Language*. Our experiment shows that this approach outperforms the alternative – to include this POS sequence as a feature in the Logistic Regression model. This is likely because fragments under the other categories may contain such POS tag sequences as well, adding noise to the classification.

5 Experimental Setup

We use the training, development and test datasets provided by (Da San Martino et al., 2020), which contains news articles from around 50 news outlets. For SI, the evaluation function gives credit to partial matching between two spans (Da San Martino et al., 2020). We base our model on the pretrained cased SpanBERT-Base model and fine-tuned them on development dataset with the following configuration: sequence length of 128, learning rate of $1e-5$, batch size of 4. We choose 64 for the hidden size of L_s^{k+1} and L_e^{k+1}. In addition, $\alpha_{sent}, \alpha_{start}, \alpha_{end}$ equal to 0.25, 0.5 and 0.5. For TC, we evaluate our model on by micro-averaged F1-measure. We base our BERT model on the pretrained uncased BERT-Base model and fine-tuned with the following configuration: sequence length of 128, learning rate of $1e-5$, batch size of 4. We utilize the solver of LBFGS, penalty of l2, C of 1.0 and "balanced" mode in Logistic Regression.

6 Results

6.1 Results of SI

We outline the performance of a set of models in Table 1. For the models trained on sentential context, the adoption of an extra sentence classifier (*SpanBERT_sent*) outperforms the base SpanBERT (*SpanBERT*). Our start and end classifiers, adopting the separate concatenated representation (*Deep_Sep*) and the combine concatenated representation (*Deep_Combine*), perform better comparing using the shallow representations. The decrease in F1-score of *Deep_Combine* implies that the start and end classifiers are conceptually equal and the boundary is not dependent to each other. To deal with the imbalanced dataset, we come up with the strategy to assign weights to different classes. *cls_weighted*† improves around 0.02 comparing to the unweighted model.

In contrast to these models trained on sentential context, the models trained on mini context mostly outperform them including *SpanBERT, SpanBERT_sent, Deep_Sep* and *Deep_Joint*. *cls_weighted*† and *loss_weighted*†‡ achieve similar F1-score with those trained on sentential context. This implies that remaining as many annotated minority classes as possible is essential when training on a small size of data. Our current model is not strict on semantics integrity. Also, while our strategy of identifying the sentential context ratains semantics integrity for the contexts, it loses some gold-labeled spans in the training dataset. Lastly, we combine *cls_weighted*†‡ trained both on sentential and mini context, which achieves 0.47108 of F1-score (SpanPro in Table 1).

Table 1: Performance of each model. †represents the inclusion of Deep_Sep into the model.‡ represents the inclusion of cls_weighted.

	sentential context			mini context		
Model	F1	Precision	Recall	F1	Precision	Recall
SpanBERT	0.44576	0.39232	0.51604	0.44761	0.41362	0.48768
SpanBERT_sent	0.45136	0.43301	0.47133	0.45587	0.42499	0.4916
Deep_Sep	0.45482	0.4357	0.4757	0.46510	0.44477	0.48738
Deep_Combine	0.45031	0.45561	0.44513	0.45041	0.43312	0.46914
cls_weighted†‡	0.46902	0.41035	0.54725	0.47013	0.43702	0.50867
SpanPro	**0.47108**	**0.37411**	**0.63591**	-	-	-

6.2 Results of TC

As shown in Table 2, the baseline, which only uses text length as features in Logistic Regression achieves 0.2653 of F1-measure. As for the polymorphic BERT, the emo_BERT with extra emotion feature embedded does not obtain a better result comparing to the typical BERT. One possible explanation is that we only extract ten types of emotion feature from the lexicon, which is not sufficient for this 14 classes

Model	F1
LR†	0.2653
emo_BERT	0.4781
cost_BERT	0.5550
BERT	0.5795
LR†‡	0.5475
Hybrid	0.6369
HybridPro	**0.6783**

Table 2: Performance of TC, † represents the inclusion of text length into the model. ‡ represents the inclusion of all features in Section 4.2 into the model.

SI in mini context			
Model	F1	Precision	Recall
SpanBERT	0.362	0.4953	0.28523
TC			
Model	F1		
Hybrid	0.54246		

Table 3: Performance of SI and TC in test dataset.

labelling task; in addition, many techniques use emotion appealing, so emotion is not a strong signal to distinguish one technique from another.

As discussed before, the training dataset is unbalanced and we refer to classes with occurrence less than 110 in training dataset as minority classes, including *Whataboutism/Straw Men/Red Herring, Thought-terminating Cliches and Bandwagon/Reductio ad hitlerum*; while the rest are majority classes. In Section 4.1.2, we introduce the cost-weighted learning approach to solve the problem of imbalanced training dataset. Table 4 shows that the cost-weighted learning approach outperforms on the minority classes by more than 0.20. We use the cost-weighted learning approach in our hybrid model.

Our experiment also shows that the Logistic Regression with the features outperforms the baseline which only integrates the text length feature, by 0.28. We compare three models: BERT, cost BERT and LR†‡with each other. For the majority classes, BERT outperforms the other two models except *Repetition*, upon which LR†‡obtains improvement by 0.42. For the minority classes, cost_BERT performs best among the three models.

We take advantage of each model's capacity of learning different features for different technique and integrate them as a hybrid model. Each of the submodels trains on the same whole training dataset and predicts one of the 14 classes. However, we only choose the predictions of *Repetition* from LR†‡, those of other majority classes from BERT and those of minority classes from cost_BERT. Our hybrid model outperform any its submodels by around 0.10 of F1-measure.

Our error analysis shows that although some instances occur only once or twice in the article, they are predicted into *Repetition*. We make further efforts to correct the mislabeled classes by the rules introduced in Section 4.3. Our model **HybridPro** achieves 0.68 of F1-measure indicating our rules for *Repetition, Slogans, Whataboutism/Straw Men/Red Herring* are effective in TC task. The details are shown in Table 4.

6.3 Remarks

It is worth noting that we present in Table 3 the performance of SpanBERT model in SI and Hybrid model in TC on the test dataset. At the test stage of the shared task, we observe that our models have the overfitting problem. We speculate that the label distribution between the development and the test datasets, hence, when the development dataset was made available again on the web site we modified the models to address the overfitting problem. The performance of our models is therefore based on the test dataset as shown in Table 3. We also report here that our F1-score of 0.5340 on the development set, shown on the leaderboard for SI task, is caused by the mistake in our text pre-processing code. Specifically, we miscalculated the end index in the processing and this results in the unreasonably high recall score in the leaderboard. We fix this problem in the code and abandon that score in this paper. All the right scores are reported in this paper.

Techniques	cost_BERT	BERT	LR	Hybrid	HybridPro
Loaded_Language	0.72	0.75	0.67	0.77	0.77
Name_Calling,Labeling	0.69	0.69	0.55	0.73	0.77
Repetition	0.27	0.25	0.67	0.62	0.70
Doubt	0.50	0.48	0.45	0.53	0.57
Exaggeration,Minimisation	0.45	0.52	0.35	0.52	0.52
Appeal_to_fear-prejudice	0.32	0.34	0.19	0.32	0.41
Flag-Waving	0.70	0.77	0.69	0.75	0.79
Causal_Oversimplification	0.27	0.39	0.32	0.37	0.41
Appeal_to_Authority	0.06	0.19	0.00	0.21	0.20
Slogans	0.47	0.55	0.31	0.55	0.68
Black-and-White_Fallacy	0.13	0.07	0.12	0.07	0.14
Whataboutism,Straw_Men,Red_Herring	0.18	0.00	0.07	0.18	0.24
Thought-terminating_Cliches	0.20	0.00	0.11	0.24	0.26
Bandwagon,Reductio_ad_hitlerum	0.60	0.00	0.33	0.60	0.60

Table 4: F1-measure of each technique

7 Conclusion

This paper develops a SpanBERT-based model for span identification (SI) and a hybrid model for propaganda techniques classification (TC).

As for SI, our paper explores different segmentation of contexts from news articles. Based on SpanBERT, we facilitate the detection by a deeper model and a sentence-level representation. The start and end boundary are conceptually independent to each other, therefore, obtaining best indexes from both the start and end classifiers achieve the best performance comparing to that of any one of them. Our model is not restricted on semantics integrity, but remaining a high ratio of span-annotated data is essential especially for a small size of training data.

Our experiment of TC offers several insights. First, we find that emotion features extracted from NRC lexicon is not effective to distinguish 14 classes. Second, we find that the cost-weighted learning approach is effective in addressing the imbalance issue of the training dataset. Third, features such as text length, TF-IDF, occur times in a document, superlative form, question words, hashtags, and supplement are useful in distinguishing different propaganda techniques.

In the future, we will explore more on how to segment a context in the training dataset and how different context affect the results. In addition, our model lacks the ability to detect the multiple spans in one context. We will conduct a fine-grained analysis to examine whether a context contains a span and if so, how many spans are included and the exact start and end boundary of them. We also have two suggestions for future work of TC. First, the use of part of speech (POS) tags in correcting mislabeled data shows a good improvement of the performance. It would be interesting to further explore its use and representations in the model. Second, while the hybrid model achieves improvement from sub-models, it'd be interesting to investigate a single model that differentiates 14 classes at the same time in the future.

References

Hani Al-Omari, Malak Abdullah, Ola AlTiti, and Samira Shaikh. 2019. Justdeep at nlp4if 2019 task 1: Propaganda detection using ensemble deep learning models. In *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*, pages 113–118.

Giovanni Da San Martino, Seunghak Yu, Alberto Barrón-Cedeño, Rostislav Petrov, and Preslav Nakov. 2019a. 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, Seunghak Yu, Alberto Barrón-Cedeno, Rostislav Petrov, and Preslav Nakov. 2019b. Fine-grained analysis of propaganda in news article. 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 5640–5650.

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.

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*.

Johan Farkas. 2018. Disguised propaganda on social media: Addressing democratic dangers and solutions. *Brown J. World Aff.*, 25:1.

Pankaj Gupta, Khushbu Saxena, Usama Yaseen, Thomas Runkler, and Hinrich Schütze. 2019. Neural architectures for fine-grained propaganda detection in news. *arXiv preprint arXiv:1909.06162*.

Karen Spärck Jones. 2004. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Jinfen Li, Zhihao Ye, and Lu Xiao. 2019. Detection of propaganda using logistic regression. In *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*, pages 119–124.

Saif M Mohammad and Felipe Bravo-Marquez. 2017. Wassa-2017 shared task on emotion intensity. *arXiv preprint arXiv:1708.03700*.

Florian Zollmann. 2019. Bringing propaganda back into news media studies. *Critical Sociology*, 45(3):329–345.

8 Appendix

Technique	Count
Loaded Language	2,199
Name Calling,Labeling	1,105
Repetition	621
Doubt	496
Exaggeration,Minimisation	493
Appeal to fear-prejudice	321
Flag-Waving	250
Causal Oversimplification	212
Appeal to Authority	155
Slogans	138
Black-and-White Fallacy	112
Whataboutism,Straw Men,Red Herring	109
Thought-terminating Cliches	80
Bandwagon,Reductio ad hitlerum	77

Table 5: The Count of Propaganda Technique in Training Dataset