UMSIForeseer at SemEval-2020 Task 11: Propaganda Detection by Fine-Tuning BERT with Resampling and Ensemble Learning

Yunzhe Jiang[1], **Cristina Gârbacea**[2], **Qiaozhu Mei**[1,2]
[1]School of Information, University of Michigan, Ann Arbor, MI, USA
[2]Department of EECS, University of Michigan, Ann Arbor, MI, USA
{yunzhej, garbacea, qmei}@umich.edu

Abstract

We describe our participation at the SemEval 2020 "Detection of Propaganda Techniques in News Articles" - Techniques Classification (TC) task, designed to categorize textual fragments into one of the 14 given propaganda techniques. Our solution leverages pre-trained BERT models. We present our model implementations, evaluation results and analysis of these results. We also investigate the potential of combining language models with resampling and ensemble learning methods to deal with data imbalance and improve performance.

1 Introduction

Propaganda techniques are used to promote or publicize a particular political cause or point of view, especially of a biased or misleading nature (Orlov and Litvak, 2018). To achieve its desired outcome, psychological and rhetorical techniques are frequently used. While initially people tend to agree with propaganda messages due to their misuse of logic and/ or arousal of emotional response, they later change their opinion and realize arguments are not convincing. Indeed, for maximum effect propaganda techniques are intended to go unnoticed. It is therefore important to detect propaganda in initial stages and identify the specific propaganda techniques used. By successfully detecting and classifying propaganda, people can look at information more rationally and logically.

This paper describes the solutions towards the Techniques Classification (TC) task of the SemEval 2020 Task 11 "Detection of Propaganda Techniques in News Articles" competition. The task requires classifying textual fragments that relate to at least one of the 14 given propaganda techniques. Our solutions leverage the pre-trained BERT (Devlin et al., 2018) based classifier. Moreover, fine-tuning allows us to conveniently adjust its pre-trained bottom layer weights on the given propaganda detection corpus shared by the organizers. Among the 46 participating teams, our submission is ranked 17.

The rest of the paper is organized as follows. In Section 2 we give the problem definition. In Section we present the propaganda dataset, in Section 4 we describe our solution and elaborate on the architectural and implementation details of the model used, in Section 5 we present results achieved by our best submission and analyze of these results, and finally in Section 6 we conclude with directions for future work.

2 Problem Definition

Given a document and a propaganda-related text excerpt from the document, the task is to identify the specific propaganda technique present in the text excerpt from the 14 propaganda classes available (Da San Martino et al., 2020). Text excerpts can be overlapping and reflect multiple propaganda techniques simultaneously, therefore the propaganda identification task needs to be approached as a multi-label multi-class classification problem. In practice, for cases when multiple propaganda techniques are used in the same text fragment, the organizers created as many copies of the respective text fragment equal to the number of propaganda techniques present in the fragment. This allows us to approach the problem as a single-label multi-class machine learning classification problem.

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 1841–1846
Barcelona, Spain (Online), December 12, 2020.

3 Dataset

The data shared with us by the competition organizers contains 8,982 plain text articles retrieved from the Newspaper3k [1] library and divided into training/ validation/ testing sets. The training set contains propaganda labels, while for the validation and testing articles we need to predict the correct labels. In Table 1 we provide an overview of the propaganda training dataset we used in our experiments, along with the 14 propaganda techniques and their training set frequency in Table 2 (the validation and testing set propaganda labels are not shared with competition participants).

Training	Validation	Testing
6,129 examples	1,063 examples	1,790 examples

Table 1: Overview of SemEval Task II Propaganda Detection training dataset.

Propaganda Technique	Training Frequency
1. Appeal to Authority	144
2. Appeal to Fear Prejudice	294
3. Bandwagon, Reductio and Hitlerum	72
4. Black and White Fallacy	107
5. Causal Oversimplification	209
6. Doubt	493
7. Exaggeration, Minimisation	466
8. Flag Waving	229
9. Loaded Language	2,123
10. Name Calling, Labeling	1,058
11. Repetition	621
12. Slogans	129
13. Thought Terminating Cliches	76
14. Whataboutism, Straw Men, Red Herring	108
Total	**6,129**

Table 2: Overview of SemEval Task II Propaganda Detection training labels.

Each article in the dataset is made up of a title specified on the first line, followed by an empty second line, and the article content starting from third line onwards, one sentence per line. In Figure 1 we include an example of a training set article from the Propaganda Detection corpus for which the propaganda labels are given. For instance, the term "babies" on the first line denotes *Name Calling, Labeling* type of propaganda. On the fourth line, "stupid and petty" is *Loaded Language* propaganda and "not looking as though Trump killed his grandma" is an instance of *Exaggeration and Minimisation*. In Figure 2 we include an example of a test article from the Propaganda Detection corpus for which we need to identify the propaganda techniques present.

4 Method

We address propaganda detection as a multi-class text classification problem, and our solution relies on the pre-trained BERT (Devlin et al., 2018) model. We choose BERT since it is already pre-trained on large amounts of data and has demonstrated strong performance in a multitude of natural language processing tasks including text classification. We fine-tune BERT-base on the textual fragments delimited by start and end indices from articles in the training set for which we know the correct labels. We apply basic text pre-processing techniques, such as lowercasing and tokenization. We also convert the textual fragments into a format compatible with BERT by adding special tokens to mark the start ([CLS]) and end ([SEP])

[1] https://github.com/codelucas/newspaper

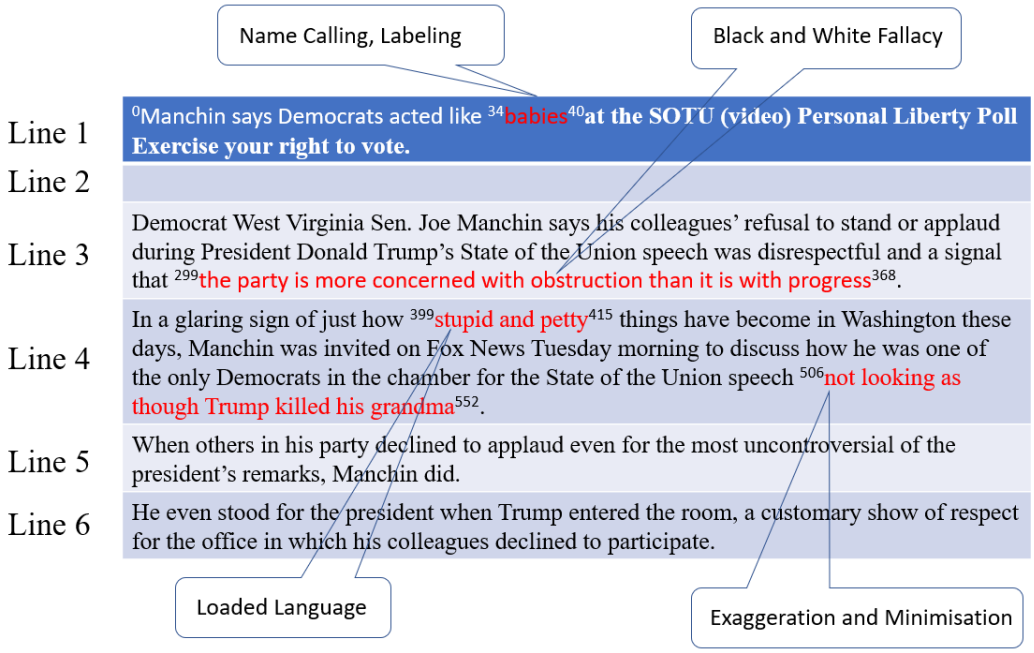

Figure 1: A training set article from the Propaganda Detection dataset. Start and end indices delimit the textual fragment (coloured in red) where a propaganda technique with known labels is present.

Line 1	Sadiq Khan [11]Slammed[18] for Pro-EU 'Message of Support' During Firework Display
Line 2	
Line 3	The spectacular fireworks that lit up the London sky on Monday night caused a stir on social media over the display's pro-EU message, at a time when the nation is divided over its [256]looming[263] withdrawal from the bloc.
Line 4	London Mayor Sadiq Khan faced [320]mounting criticism[338] after the capital's New Year's Eve fireworks display, which celebrated ties with the European Union, [440]left a bad taste in the mouths[470] of some Brits.
Line 5	The 135-metre-high London Eye was lit up in blue while its tubs turned yellow, with the giant Ferris wheel resembling the star-studded flag of the European Union.
Line 6	Sadiq Khan called his fireworks display a [691]"message of support"[711] to EU citizens living in London.

Figure 2: A testing set article from the Propaganda Detection dataset. Start and end indices delimit the textual fragment (coloured in red) where a propaganda technique with unknown labels is present.

of each sentence, pad sentences with special token [PAD] to ensure all have the same length of 64 tokens, and differentiate real tokens from padding tokens using a mask list which contains 0 for padding tokens and 1 for all other tokens. In Figure 3 we include an example of a propaganda textual fragment after pre-processing.

Not all 14 classes are equally represented in the propaganda dataset, and to alleviate data imbalance issues we use oversampling and undersampling techniques. We oversample the number of examples for minority classes, and undersample the number of examples for majority classes; please see Figure 4. When oversampling, we resample each class containing less than 400 samples to 400 samples by bootstrap sampling, i.e random sampling with replacement. When undersampling, we resample every class for which the number of samples is greater than 400 to 400 samples also by bootstrapping. We choose 400 as the number of samples per class given there are 6,129 propaganda samples in total in the training data and the goal is to have all 14 propaganda classes equally represented.

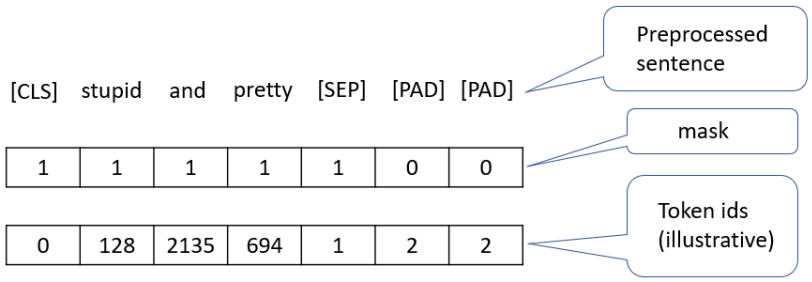

Figure 3: Textual fragment after pre-processing.

The data obtained after the resampling process is used in combination with bagging (Quinlan and others, 1996). A total of nine BERT based models are assembled on different subsets of the training data with randomly sampled examples. Each training subset contains 400 examples per propaganda class, therefore each BERT bagging model is trained on 5,600 training examples (14 classes x 400 examples/class = 5,600 total examples). In Figure 4 we present the architecture of BERT models used with bagging.

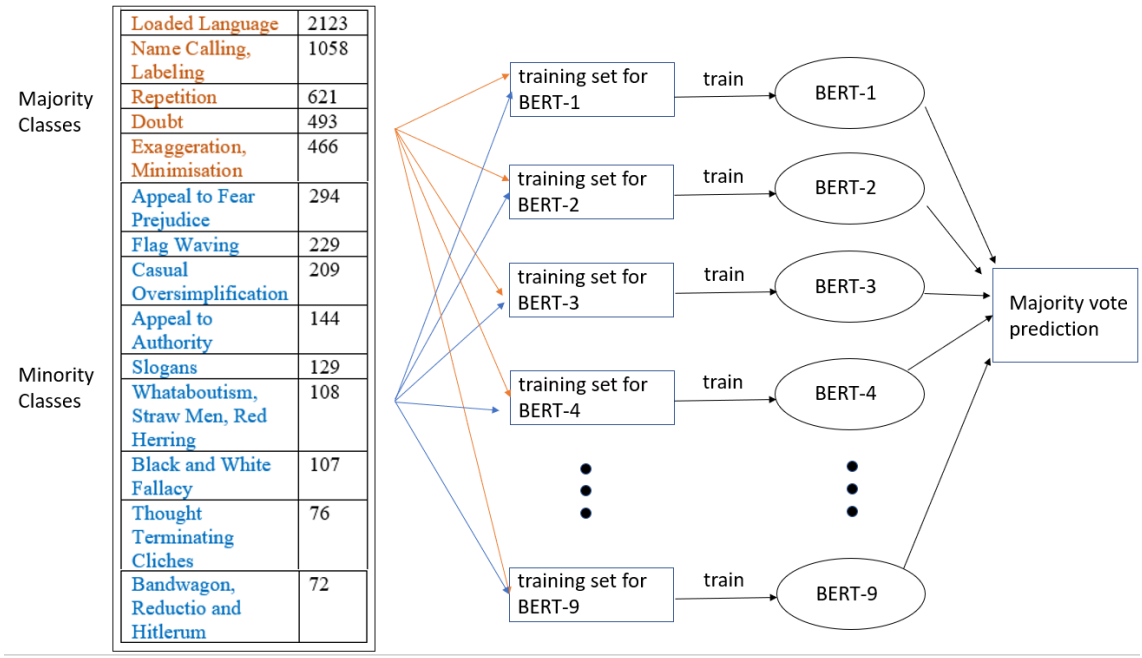

Figure 4: The architecture of BERT based models with resampling and bagging

We implement our solution in PyTorch and use HuggingFace (Wolf et al., 2019) library. We train the BERT-base-uncased model with 12 Transformer (Vaswani et al., 2017) layers and 110 million parameters. We chose model hyper-parameters by doing grid search and for our submission we use the following values: max sequence length set to 64, batch size equal to 32, a learning rate of 2e-5 and Adam (Kingma and Ba, 2014) as the optimization algorithm. We use 4 fine-tuning epochs with 10-fold cross validation on the training data set provided by competition organizers. We optimize for the F1 score in our evaluation.

5 Result and Analysis

In what follows we present results for our final BERT-based submission which ranks 17 among 46 participating teams. Our final run achieves 0.5729 validation F1-score and 0.5514 testing F1-score. Comparing our submission with the performance of other teams in the competition, we observe that submissions ranked 4-17 achieve comparable F1-scores to ours (team 4 reports 0.589 F1 score), and that teams ranked 1-3 achieve higher F1 scores than us (0.617 F1 score and beyond).

Next we assess the performance of our run at finer granularity for each propaganda technique. In Table

Propaganda Technique	Unbalanced Data		Balanced Data and Bagging
	Validation F1	**Testing F1**	**Validation F1**
1. Appeal to Authority	0.0000	0.2941	0.1290
2. Appeal to Fear Prejudice	0.2970	0.3197	0.2736
3. Bandwagon, Reductio and Hitlerum	0.0000	0.0975	0.1600
4. Black and White Fallacy	0.0000	0.0000	0.2380
5. Causal Oversimplification	0.3999	0.0000	0.2285
6. Doubt	0.5060	0.5721	0.4297
7. Exaggeration, Minimisation	0.5306	0.2857	0.3896
8. *Flag Waving*	0.7100	0.5614	*0.6904*
9. Loaded Language	*0.7331*	*0.7302*	0.6482
10. Name Calling, Labeling	*0.7131*	*0.7079*	*0.6632*
11. Repetition	0.3083	0.2148	0.3609
12. Slogans	0.0444	0.3921	0.4175
13. Thought terminating Cliches	0.0000	0.1428	0.3030
14. Whataboutism, Straw Men, Red Herring	0.0000	0.0000	0.2424
Overall F1-score	**0.5729**	**0.5514**	**0.5093**

Table 3: F1-scores achieved by our final submission of each propaganda class on unbalanced data and F1-scores achieved by BERT with bagging when on balanced training data. On unbalanced data, Loaded language and Name Calling, Labeling are the propaganda techniques our model can identify best. On unbalanced data, Flag Waving and Loaded language are classes for which the model works best.

3 we present the F1 scores for each propaganda class in the development and testing sets. On the one hand, we observe that propaganda classes which are most represented in the training data achieve best F1-scores on the validation and testing sets. Propaganda techniques such as *Loaded Language* and *Name-Calling, Labeling* which contain the most samples in the training dataset also rank highest on validation and testing sets. On the other hand, propaganda techniques with less than 200 samples in the training data have validation F1-scores close or equal to 0. This is illustrative of the severe class imbalance present in the training dataset.

In Table 3 we also present the performance of the BERT based model when doing bagging in combination with oversampling and undersampling and selecting the majority vote of all nine classifiers as the final model prediction. We observe that when training on balanced data samples with bagging, we alleviate the lack of enough data problem for propaganda classes for which we previously obtained very low F1-scores. The overall Validation F1-score for BERT with bagging is 0.5093, which is lower value compared to BERT-based without bagging (0.5729 Validation F1-score). Given we did not perform extensive hyperparameter tuning for BERT with bagging due to time constraints, we believe it is possible to considerably improve its performance when hyperparameters are carefully selected. The results also show that ensemble methods are a promising approach to reduce variance between propaganda classes and provide better model stability.

6 Conclusion and future work

We have presented our solution for the SemEval 2020 Task 11 "Detection of Propaganda Techniques in News Articles" - Techniques Classification task. We find that fine-tuning the pre-trained BERT-base model on the propaganda detection corpus provides a solid approach given all external knowledge it incorporates. We aim to improve on that by using ensemble methods such as bagging and training multiple instances of BERT on different subsets of the training data assembled with oversampling and undersampling. We find that BERT with bagging improves F1-scores for classes underrepresented in the training data.

In future work we plan to investigate systematic ways to tune the model hyperparameters when bagging and combine the pre-trained BERT language model with other ensemble learning methods, such as boosting and stacking. In addition, we plan to explore the application of more advanced oversampling and undersampling methods such as SMOTE (Chawla et al., 2002).

References

Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357.

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*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Michael Orlov and Marina Litvak. 2018. Using behavior and text analysis to detect propagandists and misinformers on twitter. In *Annual International Symposium on Information Management and Big Data*, pages 67–74. Springer.

J Ross Quinlan et al. 1996. Bagging, boosting, and c4. 5. In *AAAI/IAAI, Vol. 1*, pages 725–730.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

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.