Team DoNotDistribute at SemEval-2020 Task 11: Features, Finetuning, and Data Augmentation in Neural Models for Propaganda Detection in News Articles

Michael Kranzlein, Shabnam Behzad, Nazli Goharian
Department of Computer Science
Georgetown University
{mmk119, sb1796}@georgetown.edu
nazli@ir.cs.georgetown.edu

Abstract

This paper presents our systems for SemEval 2020 Shared Task 11: Detection of Propaganda Techniques in News Articles. We participate in both the span identification and technique classification subtasks and report on experiments using different BERT-based models along with handcrafted features. Our models perform well above the baselines for both tasks, and we contribute ablation studies and discussion of our results to dissect the effectiveness of different features and techniques with the goal of aiding future studies in propaganda detection.

1 Introduction

Propagandist news articles are misleading in nature and aim at biasing their audience towards a particular point of view by using psychological and rhetorical techniques, including loaded language, name calling, repetition, exaggeration, minimization, etc. With the rapid growth in the number of online sources of information and the speed with which information spreads online, manual flagging of propagandist news articles has become untenable, leading to an ongoing need for new research on methods for identifying these articles automatically to mitigate the negative influence they might have on users.

Until very recently, most of the work in this area has focused on article-level detection (Rashkin et al., 2017; Barrón-Cedeño et al., 2019a). However, in 2019, Da San Martino et al. (2019b) published a corpus of English news articles with individual spans of propaganda annotated that addresses the problem at a more granular level. This corpus is used in shared tasks at NLP4IF-2019 (Da San Martino et al., 2019a) and at SemEval-2020 (Da San Martino et al., 2020). The 2020 shared task is a modified version of the prior year's task and includes two subtasks[1]:

1. **Span Identification (SI):** Given a plain-text document, identify those specific fragments which contain at least one propaganda technique. This is a binary sequence tagging task.
2. **Technique Classification (TC):** Given a text fragment identified as propaganda and its document context, identify the applied propaganda technique in the fragment.

We present our models[2] for both tasks alongside discussions of our results and ablations.

2 Background

Automatic fact-checking and propaganda detection in news articles have attracted considerable attention in recent years (Potthast et al., 2018; Helmstetter and Paulheim, 2018; Baly et al., 2018). Many of these works focus on identifying propaganda at the article level. Rashkin et al. (2017) created a corpus of news articles with 4 different news types: Trusted, Satire, Hoax and Propaganda. They compared the language of these different types and reported classification results using LSTM, Max-Ent, and Naive Bayes models. Barrón-Cedeño et al. (2019b) experimented on this corpus with a maximum entropy classifier to discriminate propagandist from non-propagandist articles.

[1] https://propaganda.qcri.org/semeval2020-task11/

[2] Model hyperparameters, code, and instructions for reproducing our results are available at https://github.com/mkranzlein/propaganda

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 1502–1508
Barcelona, Spain (Online), December 12, 2020.

Da San Martino et al. (2019b) formulated the problem of detecting propaganda techniques in text and released a new corpus for the problem. They also presented a multi-granularity neural network for this task that outperformed several baselines. This dataset was used in the NLP4IF-2019 Shared Task (Da San Martino et al., 2019a) with 2 subtasks: Fragment-Level Classification (FLC) and Sentence-Level Classification (SLC). Multiple systems using novel approaches were submitted.

Gupta et al. (2019) designed multi-granularity and multi-tasking neural architectures which could jointly detect sentence- and fragment-level propaganda. They also report results on different ensemble schemes. Fadel et al. (2019) presented an ensemble of the transformer version of the Universal Sentence Encoder (Cer et al., 2018) and BERT. Many other teams also incorporated BERT-base or BERT-large with a linear layer or LSTM (Hou and Chen, 2019; Hua, 2019; Mapes et al., 2019; Yoosuf and Yang, 2019). Li et al. (2019) tested a model that makes indirect use of BERT, but ultimately attained slightly higher performance using a logistic regression model incorporating TF-IDF vectors and emotion features, among others.

Several other teams also experimented with handcrafted features such as part-of-speech tags, character n-grams, punctuation frequency, sentence length variance, average word length and sentiment features (Vlad et al., 2019; Ferreira Cruz et al., 2019; Al-Omari et al., 2019).

The datasets for both 2019 subtasks are imbalanced. Almost all teams made some attempt to address this issue. Methods included undersampling, data augmentation techniques such as oversampling and synonym insertion, dropping words (Yoosuf and Yang, 2019; Aggarwal and Sadana, 2019; Tayyar Madabushi et al., 2019; Ek and Ghanimifard, 2019) or adjusting the threshold for prediction probabilities (Alhindi et al., 2019).

Here, we describe our efforts for both subtasks of SemEval 2020 task 11. For SI, we experimented with 3 different model architectures and combinations of features like BERT embeddings, part of speech (POS) tags, and named entities. For TC, our best model is an LSTM with BERT embeddings plus similar features. We were able to enhance the performance of this model via language model finetuning and data augmentation.

3 Task 1: Span Identification

Span identification shares some structural similarities with question answering (QA) and named entity recognition (NER), though these tasks all differ in purpose. In QA, many datasets contain question-context pairs with *one* gold answer span highlighted within the context paragraph of each pair. In SI and NER, while we still need to highlight a portion of the text, there is no question/query, and there's no prespecified number of markables. In one sense, SI is more challenging than NER because propaganda spans show greater variance in length and content. Named entities are usually noun-like and fairly short, while the length of a propaganda span can range from one word—as is often the case for loaded language—to multiple sentences. On the other hand, SI is simpler in that the prediction is not multiclass. In this task's framing of the problem, predicting propaganda types is done separately in TC after the spans have already been marked.

3.1 Preprocessing

We built three types of models, which all rely on similar preprocessing steps. Each news article in the corpus is a sentence-segmented text file. Gold labels for propaganda spans are given as pairs of character start and stop indexes. This leads to an important design decision: do we look for propaganda at the token level or at the character level? While character-level predictions have been used to good effect in similar tasks like named entity recognition (Kuru et al., 2016), we choose to make predictions over tokens in order to make use of BERT's contextualized word embeddings. This does bound performance slightly, since some gold propaganda spans include partial tokens, and our models can only make decisions about entire tokens. Future models may benefit from combining token-level and character-level information.

Because gold labels are character-indexed but we want to predict token labels, our first preprocessing step is to tokenize and and generate a binary *propaganda* or *not propaganda* label for each token based on whether any of the token's characters fall inside of a character-indexed propaganda span. This requires knowing the character index of each token. We use spaCy, which provides this functionality in its tokenizer. However, this means we lose some performance in BERT since we're not using the wordpiece tokenizer, where converting character indexes to token indexes and vice-versa is much more challenging.

For featurization, we use spaCy to automatically extract part-of-speech and named entity vectors. Then, in the training data, we count the number of propaganda spans consisting of a single token. For each token in our model input, we include the number of times it appeared in the training data as a single-token propaganda span as an additional feature. This is the propaganda keyword frequencies (KW) feature referenced in table 1. We then get BERT embeddings (using the base version due to GPU memory constraints) for each token.

3.2 Models

For each of our three SI models, a training instance is based on the tokens of one sentence. Each model makes a binary prediction about all the tokens in the sentence, and we convert these token predictions back to the original character-indexed format. For each article, we combine the predictions for each of the article's sentences and merge overlapping spans before evaluating.

BERT Our first model is the most straightforward. We simply train Huggingface's BertForTokenClassification off-the-shelf model on our preprocessed tokens and labels. This model is just the original transformer-based BERT architecture adapted for tasks like NER and SI by adding a linear layer that makes predictions over each token.

LSTM Our second model is LSTM-based and uses fixed BERT embeddings plus a feature vector of part-of-speech tags, named entities, and propaganda keyword frequencies.

Bert Predictions & Features Our third (and best) model integrates the benefits of the previous two models. This model does not use any embeddings directly. Instead, it uses the same feature vector as the LSTM model plus one additional feature for each token: the predicted probability of that token being propaganda according to the BERT model.

3.3 Results

Here, we report results and ablations on the dev set[3]. The LSTM model input consists of BERT embeddings and features, which include part-of-speech tags (POS), named entities (NER), and propaganda keyword frequencies (KW). Included ablations show how performance is affected by removing each feature individually (e.g. LSTM - POS) and by omitting all features (i.e. input of fixed BERT embeddings only).

	P	**R**	**F**
# Gold Spans		941	
LSTM (Embeddings only)	25.3	46.5	32.8
LSTM (POS + NER + KW)	27.4	47.0	34.6
LSTM - POS	28.3	41.1	33.5
LSTM - NER	26.2	42.6	32.4
LSTM - KW	27.4	48.0	34.9
BERT	31.0	**50.5**	38.4
BERT Predictions & Features	**32.2**	50.2	**39.2**
Team syrapropa	39.9	80.8	53.4

Table 1: SI results on the dev set. Among our models, the best (by F1) is shown in bold. For comparison, we include the dev results for Team syrapropa, who had the best performance on the dev set leaderboard.

For the test set, we include results for the model that performed the best on the dev set, our BERT Predictions & Features model. Every model with added features outperforms our original LSTM model, which learns only from fixed BERT embeddings. However, while the part-of-speech and named entity features appear to help, the keyword frequency feature looks like it may actually hurt the model slightly. Removing it lead to roughly a 1% absolute increase in F1. Our vanilla BERT model outperformed all variants of our LSTM models, as expected, and integrating our BERT model predictions with our features gave us a modest further improvement.

[3] Our dev set results and ablations reflect bug fixes and optimizations after and are slightly higher than our leaderboard results. On the test set, however, in order to remain consistent with the final rankings, we only report our leaderboard results.

	P	R	F
# Gold Spans		1791	
Our Model	42.4	34.2	37.9
Team Hitachi	56.5	47.4	51.6

Table 2: SI results on the test set. Our model ranks 22 out of 36.

4 Task 2: Technique Classification

For TC, we are given propaganda spans annotated according to the set of 14 labels shown in table 3. Our approach is similar to our models for SI. We experimented with using an off-the-shelf BERT solution, but failed to find hyperparameters that lead to model convergence. One possible reason for this is the limited size of the training data for the task. Even though the datasets for both tasks are the same, the number of training instances is not, since the two tasks have very different objectives. In SI, every token gets a prediction, whereas in TC, we're only making predictions about previously identified propaganda spans. The training set for TC contains only 6,129 such spans.

Data Augmentation Another challenge of this task is the stark imbalance of class labels. In the previous version of this task, several teams addressed the class imbalance by undersampling, and a few took the opposite approach of oversampling. Both tactics lead to better results compared to training directly on the unbalanced dataset. In our approach to the current task, we used the Snorkel library [4] to add samples to classes with less data (12 of the classes). Snorkel facilitates the creation of "silver" data with techniques such as randomly replacing verbs/nouns/adjectives with synonyms and substituting different proper nouns in place of those the original training instances. Using these techniques, we generated about 3,000 new training samples. As shown in table 4, this lead to a relative increase in performance of 4.6%.

4.1 Preprocessing

All of our models for TC only consider the context of the provided span. In retrospect, incorporating contextual information at the article or sentence level probably should have been a priority, given that some spans can be very short and don't betray much information about themselves on their own. Preprocessing for TC is much simpler than preprocessing for SI, since we don't have to generate token labels from character indexes. Instead, all we have to do is tokenize the spans and then featurize. We use the same features as described in §3.2: POS tags, named entities, and KW frequencies.

4.2 Models

On the heels of the success we had with featurization in SI, our first attempt at TC is an LSTM over fixed BERT embeddings and all features. Then, we also experiment with training on an expanded training set (the union of the original training set and our previously described augmented data.)

4.3 Results

Many patterns in our results are expected, but some stand out. As we might have anticipated, our model did best on the two most frequent classes, which together make up more than 50% of the training data. Incorporating the augmented data into our training set to address class imbalance helped our model do better on rare classes, as compared to some of the systems with better overall performance.

Curiously, our model is not as good at identifying *Repetition* compared to other classes, despite the fact that this is the third most frequent class in the training data. On the surface, this is surprising, since repetition is usually superficially simple—it's often just multiple uses of the same token. Upon further inspection though, this seems to be an artifact of the annotation schema and the fact that we are not making use of contextual information around the span. In the training data, most spans of repetition are single tokens (likely where the second instance of the token is the one marked as propaganda and the first is unmarked). This creates two challenges: first, since the model only looks at the span as input, it is unaware that an identical token is nearby, and second, the model may learn to falsely associate specific tokens with being propaganda, leading to many false positives when that token is encountered at test time.

[4] `https://github.com/snorkel-team/snorkel`

Incorporating sentence context into the model input could lead to a significant increase in the model's ability to recognize instances of repetition, and potentially other classes that suffer from similar problems.

On flag-waving, our model does remarkably well given that flag-waving makes up only 4% of the training dataset. Flag-waving is likely an easier class in general, drawing from a smaller vocabulary of nationalistic terms that are easy for the model to pick out, compared to other classes.

And finally, we observe that while the mean span length varies considerably between classes, there doesn't seem to be a clear high-level correlation one way or the other between mean span length and F1.

Propaganda Class	**Proportion**	**Mean Span Length**	**F**
Loaded language	35%	24	68.4
Name calling, labeling	17%	26	60.6
Repetition	10%	18	19.4
Doubt	8%	125	46.3
Exaggeration, minimisation	8%	44	27.2
Appeal to fear/prejudice	5%	99	29.8
Flag-waving	4%	62	53.8
Causal oversimplification	3%	124	14.9
Appeal to authority	2%	139	22.6
Slogans	2%	25	28.1
Whataboutism, straw man, red herring	≤ 2%	97	9.7
Black-and-white fallacy	≤ 2%	105	24.5
Though-terminating clichés	≤ 2%	30	12.2
Bandwagon, reductio ad hitlerum	≤ 2%	96	4.5

Table 3: Test set micro-average F1 scores sorted by class proportion in the unaugmented training set. Mean length of propaganda spans measured in characters.

	F
# Gold Spans	1,064
LSTM	51.7
LSTM + Data Augmentation	54.1
LSTM + Finetuning	**54.8**
LSTM + Finetuning + Data Augmentation	54.5
Team ApplicaAI	70.4

(a)

	F
# Gold Spans	1,791
Our Model	49.7
Team ApplicaAI	**62.1**

(b)

Table 4: TC results on the dev set (a) and test set (b). P and R are not shown because this is a multiclass task. F1 is microaveraged. Our model ranks 24 out of 32 on the test set.

5 Conclusion

We investigated several models and combinations of features to identify propaganda spans in text, and classify the techniques used within the span. For the span identification task, we found that our LSTM-based model combining BERT predictions with our original features gives the highest F1 score of 39.2% on the dev set. Among the features we used, named entities and POS tags were found to be the most useful for this task. For the technique classification task, when using pre-trained BERT, we found data augmentation helps the LSTM model, however, when we finetuned BERT, we got better results without using the augmented data for training. Our best model for this task got an F1 score of 54.8% on the dev set.

The top teams in the SemEval 2020 shared task achieved an F1 score of 51.6% and 62.1% on the test set for the SI and TC task respectively. These scores represent significant improvements over the baselines. Nonetheless, we believe that there is still room for improvement on this challenging task, and we hope that our results, ablations, discussions, and the release of our models will help the research community in further studies.

References

Kartik Aggarwal and Anubhav Sadana. 2019. NSIT@NLP4IF-2019: Propaganda detection from news articles using transfer learning. In *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*. Association for Computational Linguistics.

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*. Association for Computational Linguistics.

Tariq Alhindi, Jonas Pfeiffer, and Smaranda Muresan. 2019. Fine-tuned neural models for propaganda detection at the sentence and fragment levels. In *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*. Association for Computational Linguistics.

Ramy Baly, Georgi Karadzhov, Dimitar Alexandrov, James Glass, and Preslav Nakov. 2018. Predicting factuality of reporting and bias of news media sources. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Alberto Barrón-Cedeño, Giovanni Da San Martino, Israa Jaradat, and Preslav Nakov. 2019a. Proppy: A system to unmask propaganda in online news. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

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

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.

Giovanni Da San Martino, Alberto Barrón-Cedeño, and Preslav Nakov. 2019a. Findings of the NLP4IF-2019 shared task on fine-grained propaganda detection. In *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*. Association for Computational Linguistics.

Giovanni Da San Martino, Seunghak Yu, Alberto Barrón-Cedeño, 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)*. Association for Computational Linguistics.

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.

Adam Ek and Mehdi Ghanimifard. 2019. Synthetic propaganda embeddings to train a linear projection. In *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*. Association for Computational Linguistics.

Ali Fadel, Ibraheem Tuffaha, and Mahmoud Al-Ayyoub. 2019. Pretrained ensemble learning for fine-grained propaganda detection. In *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*. Association for Computational Linguistics.

André Ferreira Cruz, Gil Rocha, and Henrique Lopes Cardoso. 2019. On sentence representations for propaganda detection: From handcrafted features to word embeddings. In *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*. Association for Computational Linguistics.

Pankaj Gupta, Khushbu Saxena, Usama Yaseen, Thomas Runkler, and Hinrich Schütze. 2019. Neural architectures for fine-grained propaganda detection in news. In *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*. Association for Computational Linguistics.

Stefan Helmstetter and Heiko Paulheim. 2018. Weakly supervised learning for fake news detection on twitter. In *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. IEEE.

Wenjun Hou and Ying Chen. 2019. CAUnLP at NLP4IF 2019 shared task: Context-dependent BERT for sentence-level propaganda detection. In *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*. Association for Computational Linguistics.

Yiqing Hua. 2019. Understanding BERT performance in propaganda analysis. In *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*. Association for Computational Linguistics.

Onur Kuru, Ozan Arkan Can, and Deniz Yuret. 2016. CharNER: Character-level named entity recognition. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 911–921, Osaka, Japan, December. The COLING 2016 Organizing Committee.

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*. Association for Computational Linguistics.

Norman Mapes, Anna White, Radhika Medury, and Sumeet Dua. 2019. Divisive language and propaganda detection using multi-head attention transformers with deep learning BERT-based language models for binary classification. In *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*. Association for Computational Linguistics.

Martin Potthast, Johannes Kiesel, Kevin Reinartz, Janek Bevendorff, and Benno Stein. 2018. A stylometric inquiry into hyperpartisan and fake news. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

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. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Harish Tayyar Madabushi, Elena Kochkina, and Michael Castelle. 2019. Cost-sensitive BERT for generalisable sentence classification on imbalanced data. In *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*. Association for Computational Linguistics.

George-Alexandru Vlad, Mircea-Adrian Tanase, Cristian Onose, and Dumitru-Clementin Cercel. 2019. Sentence-level propaganda detection in news articles with transfer learning and BERT-BiLSTM-capsule model. In *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*. Association for Computational Linguistics.

Shehel Yoosuf and Yin Yang. 2019. Fine-grained propaganda detection with fine-tuned BERT. In *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*. Association for Computational Linguistics.