newsSweeper at SemEval-2020 Task 11: Context-Aware Rich Feature Representations For Propaganda Classification

Paramansh Singh* **Siraj Sandhu*** **Subham Kumar*** **Ashutosh Modi**
Indian Institute of Technology Kanpur (IITK)
{params,sssandhu,subhamkr}@iitk.ac.in
ashutoshm@cse.iitk.ac.in

Abstract

This paper describes our submissions to SemEval 2020 Task 11: *Detection of Propaganda Techniques in News Articles* for each of the two subtasks of *Span Identification* and *Technique Classification*. We make use of pre-trained BERT language model enhanced with tagging techniques developed for the task of Named Entity Recognition (NER), to develop a system for identifying propaganda spans in the text. For the second subtask, we incorporate contextual features in a pre-trained RoBERTa model for the classification of propaganda techniques. We were ranked 5^{th} in the propaganda technique classification subtask.

1 Introduction

Propaganda is commonly defined as information of a biased or misleading nature, possibly purposefully shaped, to promote an agenda or a cause. Commonly used propaganda techniques are psychological and rhetorical - ranging from the selective presentation of facts and logical fallacies to the use of loaded language to produce an emotional response. Historically, propaganda has been widely employed and is often associated with governments, activists, big business, religious organizations, and partisan media.

Having been first introduced in a previous iteration of the SemEval shared task (Martino et al., 2019), the current shared task (Da San Martino et al., 2020) consists of two subtasks, **(i) Span Identification(SI):** Given a plain-text document, identify those specific fragments which are propagandistic, and **(ii) Technique Classification(TC):** Given a text fragment identified as propaganda and its document context, identify the applied propaganda technique in the fragment. Both the subtasks focus on English texts only. An illustration is shown in Figure 1.

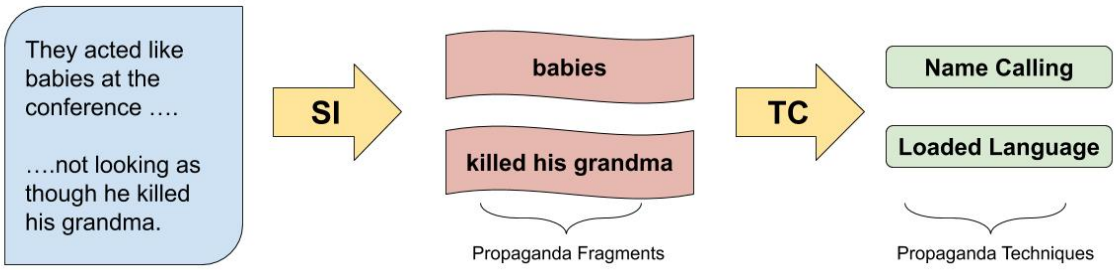


Figure 1: Example of subtasks *span identification*(SI) and *technique classification*(TC)

The previous iteration of this task required identifying the propaganda spans and also identifying the propaganda technique for that fragment. The highest scoring system by Yoosuf and Yang (2019) employed an out-of-the-box BERT (Devlin et al., 2018) language model fine-tuned on the token classification task - whether the token is part of a propaganda span or not. For the classification task, the same system employed an N+2 way classification, where N is the number of propaganda classes. For the technique

* Authors equally contributed to this work.

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 1764–1770
Barcelona, Spain (Online), December 12, 2020.

classification, we make use of document context. For context inclusion, we build upon an approach by Hou and Chen (2019), where they use title and previous sentence as context. They pass the context concatenated with the original text for fine-tuning the BERT model.

Our approach for span identification makes use of a state-of-the-art language model enhanced by tagging schemes inspired from Named Entity Recognition which aim to model spans better - logically and intuitively - by involving *Begin* and *End* tags (Ramshaw and Marcus, 1995) to better formulate spans. In this regard, we experimented with BERT, RoBERTa (Liu et al., 2019), SpanBERT (Joshi et al., 2019), GPT2 (Radford et al., 2019) language models and with BIO, BIOE, BIOES tagging schemes. For the final technique classification model, we use RoBERTa language model to get the contextual sequence representation for the propaganda fragment and perform classification.

Our best performing models ranked 13^{th} for SI subtask and 5^{th} for the TC subtask on an unknown test set. The implementation for our system is made available via Github[1].

2 Problem Statement

For the SI subtask, given a document **D** we have to identify all such fragments of text which are propagandistic. These character level input spans are converted to token level labels (see §4.2.1), which makes this problem a token classification problem. The approach has been detailed out in § 3.1. For the TC subtask, given a span identified as propagandistic, we have to classify it into one of the 14 techniques (Da San Martino et al., 2020). Note that we are also given the document **D** containing the span which we can use as context. The approach has been detailed out in § 3.2. In § 4.1 we describe the dataset.

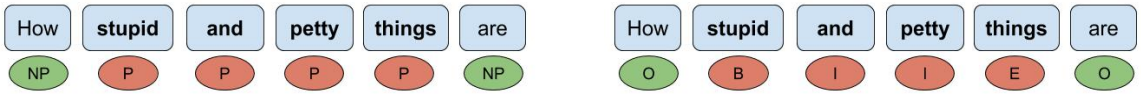
3 System Description

3.1 Span Identification

Span Identification is a binary sequence tagging task where we classify each token in a sequence as a propaganda/non-propaganda text (P/NP), that is whether it is part of a propaganda fragment. For this, we pass the input sequence to a pre-trained Transformer based model (for example, BERT, RoBERTa etc.) and get embeddings for each token in the sequence. These embeddings are then passed to the classification layer on top of the Transformer which classifies the token as P/NP. The overall architecture of the system is shown in Figure 3a. Note that the classifier weights shown are shared across all tokens.

Tagging

In the approach described above, each token in the sequence is tagged as P/NP. We take this approach forward and tag using BIOE scheme. In this scheme, **B** represents the *Beginning* of propaganda text, **I** means *Inside* propaganda text, **E** represents the *End* of propaganda text, and **O** means *outside* the propaganda text i.e., non-propaganda text. The comparison between the texts tagged using the two schemes is shown in Figure 2. For span identification, the rest of the model architecture is the same, but the tokens are now labelled as BIOE instead of just P/NP earlier. During prediction, the tokens labeled as **B**, **I**, or **E** are classified as propaganda token.



(a) P/NP tagging

(b) BIOE tagging scheme

Figure 2: Comparison of the two tagging schemes. The text shown in bold is a propaganda fragment.

Begin and End tags help to formulate the notion of spans better and model it as a *span identification problem* rather than a *token level classification problem* by introducing dependencies between the various tokens part of a propaganda span.

[1] https://github.com/paramansh/propaganda_detection

3.2 Technique Classification

Technique classification is a multiclass sequence classification task, where we identify the propaganda technique for a given propaganda fragment. In addition to the fragment, we also have the document containing the fragment. To solve this task, we first pass the propaganda sequence and its context to a pre-trained Transformer. We use the Transformer outputs to get the sequence representation $\mathbf{S}$ and context representation $\mathbf{C}$. Both of these are combined to get the contextualized vector representation $\mathbf{V}$ of the sequence. This vector representation $\mathbf{V}$ is then passed to the classifier layer on top, which performs the final classification (refer Figure 3c).

Sequence and Context Vectors

We use pre-trained RoBERTa to obtain the representation for the sequence via the CLS output vector. The CLS output vector gives the aggregate sequence representation as modeled in Devlin et al. (2018).

The context surrounding the span can be at different granularity: article level, paragraph level, or sentence level. Obtaining vector representation for a paragraph or an article context using BERT or other language models is difficult as articles can be very long. As a result, the article context vector is obtained using the CLS representation of the headline of the news article as suggested by Hou and Chen (2019). To capture the sentence context, we use the sentence corresponding to the propaganda fragment. If the fragment spans across multiple sentences, all such sentences are considered. However, we limit the length of sentence contexts to 130 words. This is because, for sufficiently long propaganda spans, the exact meaning can be inferred directly from the text without any surrounding context.

Contextual Sequence Representation

We followed two approaches to include context. In the first system, we pass the context text concatenated with the propaganda fragment text directly to the Transformer (refer Figure 3b). We take the resultant sequence embedding output from the Transformer as the contextual representation ($\mathbf{V}$) of the fragment. In this case, $\mathbf{S}$ and $\mathbf{C}$ are not explicitly generated. One problem with this approach is that in case of small propaganda fragments (2 or 3 words), the longer context text will influence the final representation, which may not be ideal.

We address this in System 2, where we pass the fragment and the context to different Transformers to get $\mathbf{S}$ and $\mathbf{C}$ independently. There is an additional hidden layer on top of the context Transformer which reduces the dimension of context vector $\mathbf{C}$. This resultant vector is then concatenated with $\mathbf{S}$ to get $\mathbf{V}$. The additional hidden layer allows the classifier to give more attention to the propaganda sequence. This system is shown in Figure 3c. Apart from concatenation, we tried another approach where we set $\mathbf{V} = \alpha\mathbf{S} + (1 - \alpha)\mathbf{C}$, i.e., a weighted average of $\mathbf{S}$ and $\mathbf{C}$.

4 Experimental Setup

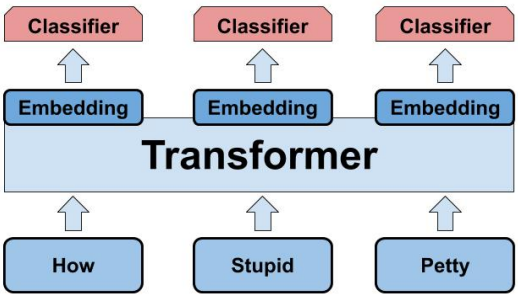
4.1 Data

The dataset is released by the Semeval task organizers (Da San Martino et al., 2019). The input for both tasks will be news articles in plain text format (unimodal). The gold labels for SI are in the form of an article identifier, a begin and an end character offset for each propaganda instance. The labels for TC are in the same format, but with an additional field for the class annotation pertaining to each instance. Note that if a text fragment falls into belongs to multiple classes, the same occurs as many times in the corpus with a single class label in each instance.

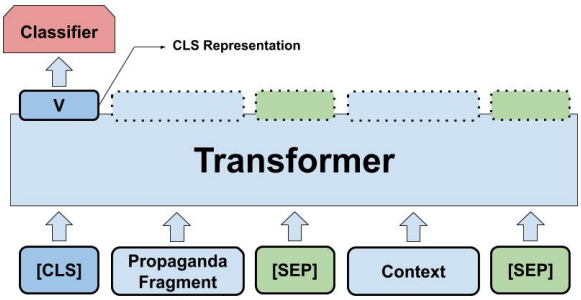
4.2 Pre-Processing

4.2.1 Span Identification

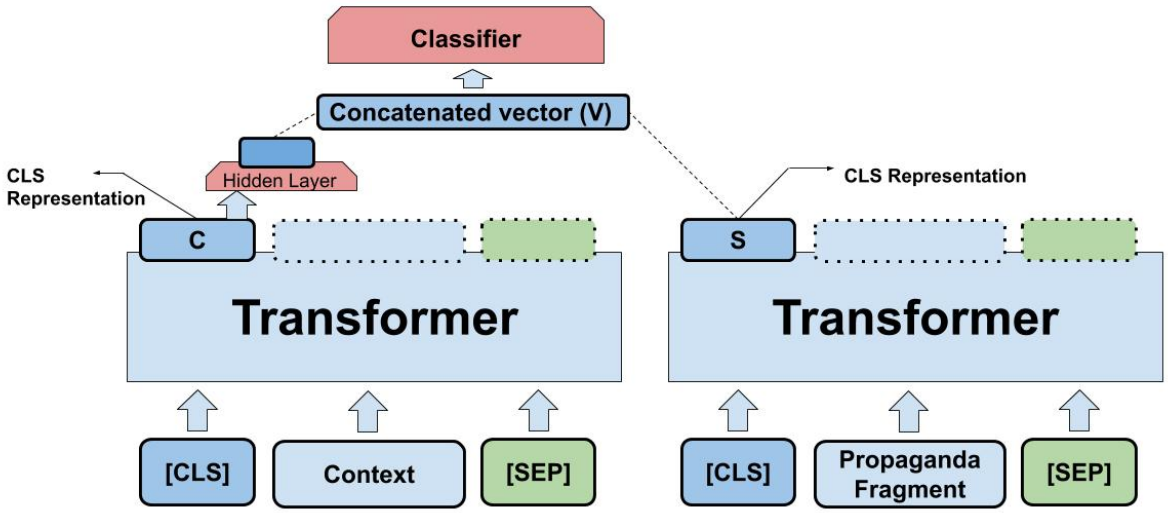
The articles are too long to directly feed standard language models and are thus split by sentences. This resulted in 17,855 sequences with a maximum length of 179 words. We convert the character level span labels to token level labels where a token labeled as positive implies that the token is part of a propaganda span. As a result, our training objective becomes classifying each token as a propaganda word or not. In



(a) Span Identification



(b) Technique Classification - System 1



(c) Technique Classification - System 2

Figure 3: Systems for SI and TC tasks. Note that Transformer can be any pre-trained language model like BERT, RoBERTa, SpanBERT etc.

the training data, approximately 13 percent of tokens were part of a propaganda text. During prediction, the consecutive tokens with the same label are merged to get the final character level span outputs.

4.2.2 Technique Classification

For this task, we get the propaganda fragment text using the given character labels. For getting the surrounding sentence context, we limit the maximum number of words to 130. We select words from both sides in our context until the end of the sentence is reached or the maximum word limit is reached. The training data consists of (sentence, context) pair and the propaganda technique is the label. We get 6129 such pairs in our training dataset. The dataset is highly unbalanced w.r.t the labels, for example, the category `Loaded Language` has 2123 instances, and `Thought-terminating_Cliche` has only 76 instances.

4.3 Training Details

We use PyTorch framework provided by huggingface[2] library for the pre-trained BERT and RoBERTa models. We use `bert-base-uncased` for span identification task, and `roberta-uncased` for Technique Classification task. In addition to this, we experiment with GPT-2 and SpanBERT as well. For both tasks, we fine-tuned using AdamW optimizer (Loshchilov and Hutter, 2017) for 4 epochs, with a learning rate of 3×10^{-5} and batch size = 8.

[2] `https://huggingface.co`

Model	F1 Score	Precision	Recall
BERT - P/NP	0.414	0.401	0.427
BERT - BIO	0.434	0.403	0.471
BERT - BIOE	**0.445**	0.387	0.521
BERT - BIOES	0.411	0.431	0.392
BERT_large - BIOE	0.411	0.416	0.407
GPT2 - BIO	0.256	0.298	0.225
SpanBERT - BIOE	0.445	0.398	0.504
RoBERTa - BIOE	0.429	0.403	0.458
BERT - BIOE - POS tags	0.411	0.431	0.392
BERT - BIOE - 4 layer concat	0.424	0.367	0.502
BERT - BIOE - 12 layer concat	0.399	0.436	0.368

Table 1: Task SI Results on development set trained with 90 percent of training data.

5 Results

5.1 Span Identification

For this task we experimented with the following, (i) Tagging Scheme: P/NP, BIO, BIOE and BIOES, (ii) Language Models: BERT (base uncased, large uncased), RoBERTa, GPT2, and SpanBERT, (iii) Concatenating different hidden layers of BERT to get token representation, (iv) Inclusion of linguistic features using POS tags of tokens. The evaluation metric used to compare is F1 score, which is calculated based on the overlap between the predicted and actual spans. The scores for different models on official development set trained with 90 percent training data have been reported in Table 1.

As it can be seen, the score improved on changing the tagging scheme from P/NP to BIO and from BIO to BIOE. As suggested earlier, this can be because, BIO, BIOE capture the span nature of the output. One way to realize this is by analyzing predictions on development set, where the average output length of spans using P/NP tagging is 33 chars, for BIOE it is 36 chars, compared to the actual average length which is 38 chars.

However, none of the other approaches resulted in any improvement. Due to less number of examples in the training data, all the models were overfitting and failed to generalize. We used the standard BERT-BIOE system for our final test set submission.

5.2 Technique Classification

In this we experiment with the following, (i) Language Models: BERT and RoBERTa, (ii) two approaches to include context as described in § 3.2, (iii) context to include: sentence level or document level (from article headline), and (iv) methods to combine the embeddings in System 2: default method, without hidden layer concatenation, and weighted average. The evaluation metric used to compare is micro-averaged F1 score. The results are reported for the official development set in Table 2. The Baseline model is the one not using contextual features.

We observed that for this task RoBERTa outperforms BERT and was thus used to generate input representations. Also, context inclusion did not result in any significant improvements in most cases. System 1 using sentence as context lead to a significant drop in F1 score, which can be attributed to the fact that the longer context sentence takes the attention away from the actual text to be classified. Other than that, most other methods performed similar to the baseline. However, System 2 with a hidden layer on top of the context vector, resulted in slightly better performance as it put more emphasis on the sequence to be classified compared to the surrounding context, which in some cases may lead to poor performance as mentioned above. Although we can expect the model to learn this after directly concatenating as well, it may not due to the lack of sufficient data. By slightly tuning hyperparameters and using all of the data for the RoBERTa baseline model, we were able to achieve an F1 score of 0.627 on the development set. We used this fine-tuned system for our test set submission.

Model	F1	Model	F1
BERT Baseline	0.583	RoBERTa Baseline	0.602
RoBERTa - Title - System 1	0.578	RoBERTa - Sentence - System 1	0.593
RoBERTa - Title - System 2 w/o Hidden	0.601	RoBERTa - Sentence - System 2 w/o Hidden	0.599
RoBERTa - Sentence - System 2 - Add	0.598	RoBERTa - Sentence - System 2	**0.611**
RoBERTa - Baseline - Length Feature	0.589	RoBERTa - Sentence - System 2 - Weighted Avg.	0.593

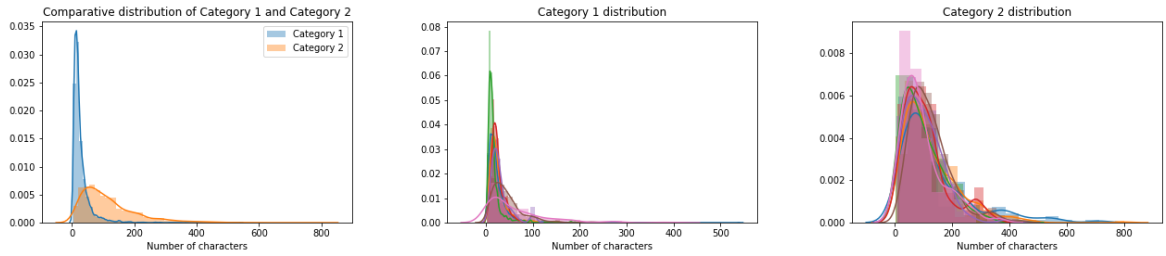
Table 2: Task TC Results on development set trained with 90 percent of training data. Baseline refers to model without contextual features. System 1 (Figure 3b) - including context by concatenating text. System 2 (Figure 3c) - including context by concatenating embeddings, with and without hidden layer above context vector.

Data Imbalance

All the models displayed a significant difference between the class-wise maximum and minimum F1 scores. Techniques such as `Loaded_Language` and `Name_Calling,Labeling` with many training examples had F1 scores close to 0.7. On the other hand, minority classes, especially the ones which are a combination of multiple techniques (`Whataboutism,Straw_Men,Red_Herrin` and `Bandwagon,Reductio_ad_hitlerum`), had F1 scores less than 0.1.

Span Length Distribution

Figure 4 details the distribution of lengths of spans for various categories. Category 1 [3] consists of 7 classes, while the remaining 7 are part of category 2. As the figure suggests, Category 1 techniques follow similar, more "peaky" distribution, while Category 2 techniques follow a similar "flat" distribution. We tried to model this by passing length as a feature after at the final classification layer and trying separate models for both, but none could beat our baseline model.



(a) Category 1 vs Category 2 (b) Category 1 techniques (c) Category 2 techniques

Figure 4: Distribution of span length for various categories

6 Conclusion

We proposed models for detecting propaganda fragments in articles and classifying the propaganda technique used in a given propaganda fragment. We portray how state-of-the-art language models can be useful for both subtasks. We show how BIOE tagging scheme can help detect spans better. For classification, we model a way to include context surrounding a propaganda fragment, although its inclusion does not improve the predictions significantly. Finally, we also show how the span lengths are distributed for different categories. Modeling this fact by doing hierarchical classification is something that can be explored in the future.

[3]Category 1 - Loaded_Language; Name_Calling,Labeling; Repetition; Slogans; Thought-terminating_Cliches; Exaggeration,Minimisation; Flag-Waving

References

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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

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*, pages 83–86, Hong Kong, China, November. Association for Computational Linguistics.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2019. Spanbert: Improving pre-training by representing and predicting spans. *CoRR*, abs/1907.10529.

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. *CoRR*, abs/1907.11692.

Ilya Loshchilov and Frank Hutter. 2017. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101.

Giovanni Da San Martino, Alberto Barrón-Cedeño, and Preslav Nakov. 2019. Findings of the nlp4if-2019 shared task on fine-grained propaganda detection.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Lance Ramshaw and Mitch Marcus. 1995. Text chunking using transformation-based learning. In *Third Workshop on Very Large Corpora*.

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*, pages 87–91, Hong Kong, China, November. Association for Computational Linguistics.