Transformers at SemEval-2020 Task 11: Propaganda Span Identification using Diversified BERT Architectures based Ensemble Learning

Ekansh Verma
IIT Madras, India
vermaekansh55@gmail.com

Vinodh Motupalli
IIT Madras, India
mvk793@gmail.com

Souradip Chakraborty
Indian Statistical Institute, India
souradip24@gmail.com

Abstract

In this paper, we present our approach for the 'Detection of Propaganda Techniques in News Articles' task as a part of the 2020 edition of International Workshop on Semantic Evaluation. The specific objective of this task is to identify and extract the text segments in which propaganda techniques are used. We propose a multi-system deep learning framework that can be used to identify the presence of propaganda spans in a news article and also deep dive into the diverse enhancements of BERT architecture which are part of the final solution. Our proposed final model gave an F1-score of 0.48 on the test dataset.

1 Introduction

Propaganda (prop-uh-gan-duh) is defined as "information, ideas, opinions, or images, often only giving one part of an argument, that are broadcast, published, or in some other way spread with the intention of influencing people's opinions". Developing artificial intelligence to identify the propaganda in an article will not only help the readers but also the authors to be careful and write with more rationality, devoid of unintentional propagandist elements. For this, the paper presents a comprehensive solution by team 'Transformers' to detect propaganda spans in news articles. We model this task as a word-level binary classification problem, which needs a lot of latent space and deep representations to encapsulate and detect the words with propaganda engraved in them. The central idea we utilized is fine-tuning a BERT (Devlin et al., 2018) model with multi-sample dropout and convolution layers on top of it to stabilize the training loss and enhance the model generalizability. We have also experimented with our model performance with various loss functions including the focus loss and weighted cross-entropy. Finally, we have developed an ensemble of BERT based models with combinations of convolutional neural network (CNN) and multi-sample dropout to efficiently detect the text spans having hidden propaganda. Our ensemble learning approach [1] has shown to perform exceedingly well and we were one of the top 10 teams in the propaganda span identification challenge .

2 Background

There were two challenge tracks as part of the SemEval2020 shared-task 11 (Da San Martino et al., 2020) namely, span identification (SI) and technique classification (TC). We have worked on the SI task and explained our methodology in this paper.

The input data provided by the organizers (Da San Martino et al., 2019) are news articles in plain text format. The article can be further divided into sentences using new line elements. The dataset has 371 train articles, 75 development articles, and 90 test articles. Each article is provided with indices of the start and end of a span of characters that represent a propaganda span. The details of the data provided can be found in Table 1.

(Li et al., 2019) propose a Logistic regression with Tf-IDF, BERT vector, sentence length, readability grade level, motion features to solve the propaganda detection task as a binary classification problem.

[1] Code to reproduce the framework can be found in https://github.com/vinodhmvk/propagnada-detection

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 1823–1828
Barcelona, Spain (Online), December 12, 2020.

	Articles	Sentences	Words	Spans
Train Set	371	16539	344095	5468
Development Set	75	3177	57783	-
Test Set	90	3185	67003	-

Table 1: Count of Dataset Items.

(Yoosuf and Yang, 2019) approached the fine-grained propaganda detection task by fine-tuning the BERT model and also investigated the attention heads for model interpretability. Also in (Yoosuf and Yang, 2019), authors methodology for propaganda sentence level classification problem involved an ensemble of BiLSTM, XGBoost, and BERT models.

3 System overview

3.1 Pre-processing

Since the data provided by task organizers had propaganda spans annotated at the character level, we converted these character level indices to word-level tokens and used them as input to our models. We also preserved the character level indices of words with respect to articles. This helped us in generating the character level indices for post-processing. We have omitted empty sentences from the model training. The words are then treated to remove any irrelevant characters using regular expressions. Since our experiments are performed on a sentence level passing, we have merged sentences that are part of a common unique propaganda span.

3.1.1 ELMo Pre-processing

For ELMo (Peters et al., 2018) based models we reconstructed the data as sequence input to sequence output. Here the input is a sequence of words belonging to one sentence with maximum length allowed to 90 words and used post padding. The output sequence of 0 and 1 with the same length as input.

3.1.2 BERT Pre-processing

For BERT based models we have the input sequence same as for the ELMo based models. The input sequence is passed through BERT tokenizer based on WordPiece (Wu et al., 2016) tokenization which divides every word in the input sequence into multiple sub-strings. For training the span identification model, we one-hot encoded the tokenized words for input sequence into a 3 class target vector which is defined as [x, y, z], where x is non-propaganda class, y is propaganda class and z indicates if the word is a sub-string created by BERT tokenizer.

3.2 Modeling

3.2.1 ELMo

ELMo provides contextualized word embeddings for various downstream tasks. ELMo extends a traditional word embedding model with features produced bidirectionally with character convolutions. We train our baseline model utilising the pre-trained ELMo embeddings. The input is the sentence wise list of tokens and the output is predicted entities: propaganda or not-propaganda. We stack two bidirectional LSTM layers on top of the ELMo embeddings. Now to this network, a residual connection is added between the first and second LSTM layers. The network is trained end to end to output token wise probability per sentence.

3.2.2 BERT Baseline

BERT stands for bidirectional encoder representations from Transformers, is designed to learn deep bidirectional representations by jointly conditioning on both left and right context in all layers. The pre-trained BERT can be fine-tuned to create competitive models for a wide range of downstream tasks, such as named entity recognition, relation extraction, and question answering. Since its inception BERT model has become quite popular both as a Natural Language Processing (NLP) research baseline and as a final task architecture.

BERT has undergone many changes to become RoBERTa (Liu et al., 2019) from Facebook. RoBERTa builds on BERT's language masking strategy, wherein the system learns to predict intentionally hidden sections of text within otherwise un-annotated language examples. In our exploration we evaluate baselines based on the two variants of pre-trained BERT language model: BERT base, BERT large and pre-trained RoBERTa base for our task of detecting propaganda spans. Input for the model are the sentences which are token wise encoded using the corresponding pre-trained model tokenizers and the token-wise labels for the input sentences. Inputs are derived from the pre-processing step described above. For our baseline models, we use the dropout layer with the dropout probability of 0.1 and add the linear classification layer with softmax activation on top of the pre-trained transformer based architectures to output final probabilities per token. This output is post-processed from token level to get word level and subsequently the character level predictions. Experiments indicated that fine tuning of BERT base model outperformed the others based on the validation F1 score and chosen as baseline for comparison in our further experiments.

3.2.3 BERT-CNN

We modify BERT token level classifier by replacing the linear layer with a one-dimensional convolutional neural network (CNN) on top of BERT followed by the softmax layer. From the experiments, we show that CNN network helps to identify local patterns in the BERT's output features and learns suitable discriminatory representation to detect propaganda spans in the text.

3.2.4 Multi-Sample Dropout

In his work (Inoue, 2019) has shown that Multi-Sample dropout, an enhanced dropout technique leads to faster training and improved generalization over the original dropout. We take cues from the work mentioned and use the same technique for our BERT based fine tuning experiments. In our implementation we use 5 dropout layers with dropout probability set to 0.5 for all of them. Different masks are used for each dropout sample in the dropout layer so that a different subset of neurons is used for each dropout sample. However, the parameters are shared between the layers preceding the multi-sample dropout. We compute loss for each dropout sample using the same loss function and the final loss value is obtained by averaging the loss values for all dropout samples. This final loss value is used as the objective function for optimization during training.

We train both BERT-CNN, BERT with linear classification head accompanied by multi-sample dropout and use them in our ensemble to output final predictions.

3.2.5 Loss Functions

The train dataset released under this challenge contains 16539 sentences in total. Out of these 3988 sentences contain at least one propaganda phrase. Also we have 5468 propaganda spans in 344095 total words present. This indicates imbalance between propaganda vs non-propaganda instances on sentence and word level as well. Since the non-propaganda data dominates the training set, it is desirable to weight the propaganda instances higher in the training. We achieve this by weighting the loss function for a word higher if it belongs to a propaganda class in a labelled dataset.

Let X denote a labelled sentence after preprocessing of length (number of tokens) N and each token x_i $\in X$ has a gold label y_i = $[y_{i0}, y_{i1}, y_{i2}]$ and p_i = $[p_{i0}, p_{i1}, p_{i2}]$ are the predicted probabilities for the three classes respectively. As mentioned in Section 3.1.2, y_{i0}, y_{i1},y_{i2} are Non-Propaganda class, Propaganda class and BERT created sub-string class respectively. The cross entropy loss for X is given by:

$$\text{CrossEntropy} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{3} y_{ij}\log p_{ij} \tag{1}$$

We use a class-wise weighting scheme provided by w where w = $[w_0, w_1, w_2]$.

$$\text{WeightedCrossEntropy} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{3} w_j y_{ij}\log p_{ij} \tag{2}$$

We tune the value of class-weights based on the model's performance on the validation set. Our Experiments showed that the weight value of 10 for propaganda class and 1 for the remaining two yields the best performance for this task. Inspired by the success of focal loss (Lin et al., 2017) in computer vision, we experimented with it to fine tune the BERT model. Focal loss is modified from cross-entropy loss by adding a modulating factor $(1 - p_t^\gamma)$ to the cross-entropy loss. Formally, the focal loss is expressed as follows:

$$\mathrm{FL} = -\alpha_t(1 - p_t^\gamma)\log p_t \tag{3}$$

However in our fine-tuning experiments, it did not offer considerable improvements over weighted cross entropy loss. Hence, it was not utilised to train final models. We suspect we could not find suitable hyper parameters to make focal loss work in our limited experiments.

4 Experimental setup

Our implementation is based on pytorch (Paszke et al., 2017) framework, Transformers (Wolf et al., 2019) for all our experiments involving fine-tuning the BERT architecture.

While training our models, we split the labelled data into two parts: the training set, consisting of 80% of the data is used to train models, whereas the remaining non-overlapping 20% of the data referred to as internal validation set, is used to test the effectiveness of the models.

All our BERT-based models are based on the uncased model of BERT. In our training procedures we utilize the AdamW optimizer with a learning rate value of 0.001. We also add some weight decay as regularization to the model's weight matrices. Hyper parameters shared across models are batch size: 32, maximum sequence length: 128, weight decay: 0.01.

We used F1 score on the validation set as a proxy for the generalization error to monitor our model's performance as training progresses. We stop training once the model's performance stops improving. When picking the model to use for our final ensemble, we pick the model with F1 score greater than a threshold on the validation set.

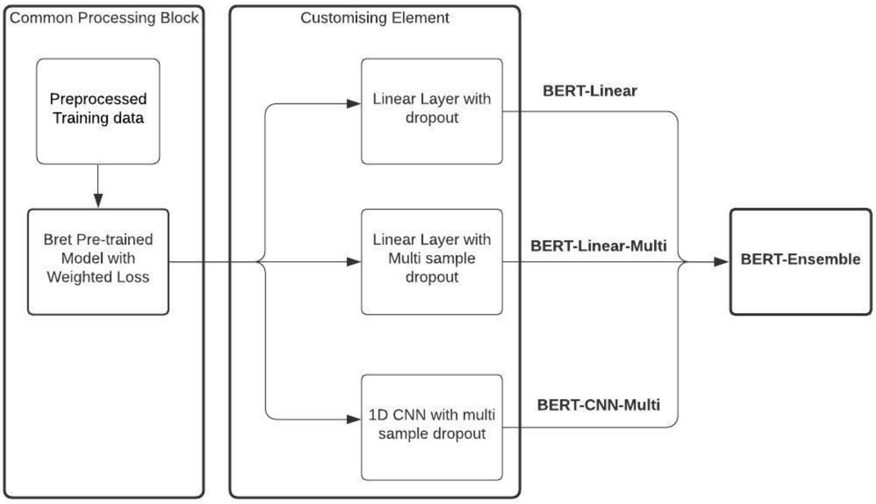
Common Processing Block
Preprocessed Training data
Bret Pre-trained Model with Weighted Loss
Customising Element
Linear Layer with dropout
Linear Layer with Multi sample dropout
1D CNN with multi sample dropout
BERT-Linear
BERT-Linear-Multi
BERT-CNN-Multi
BERT-Ensemble

Figure 1: Multi System Framework

5 Results and Conclusion

The final models for our ensemble are shown in Figure 1. We get the final probability by taking the average of raw scores of the propaganda class from all the three models, these world level scores range between 0, 1. In post-processing we convert the word level tags into character level indices for evaluation.

The performance of the final showcase models can be seen in Table 2. We used the performance of a model on a development dataset as a benchmark to choose better performing models. The best performing

Model	Metric	Dev	Test
ELMo	F1	0.38	0.41
	Precision	0.29	0.36
	Recall	0.56	0.46
BERT-Linear	F1	0.428	0.469
	Precision	0.338	0.427
	Recall	0.583	0.520
BERT-Linear-Multi	F1	0.426	0.469
	Precision	0.353	0.444
	Recall	0.536	0.496
BERT-CNN-Multi	F1	0.421	0.471
	Precision	0.324	0.421
	Recall	0.597	0.534
BERT-Ensemble	**F1**	**0.437**	**0.481**
	Precision	0.354	0.443
	Recall	0.571	0.527

Table 2: Performance of Models on Development and Test sets.

single model is BERT-Linear with a F1 score of 0.428. But the overall best model is BERT-Ensemble with an F1 score of 0.438. We achieved an F1 score of 0.481 on test datasets.

In this paper, we have investigated models and techniques to detect if a text span in an article is propaganda or not. Experimental results showed that the ensemble of fine tuning modified BERT based architectures has achieved the best results. Regarding future work, we plan to explore a semi-supervised paradigm to train the models with less labeled data. Also, we want to explore FLAIR embeddings (Akbik et al., 2018) and latest text-to-text transfer T5 (Raffel et al., 2019) architecture which have shown excellent performance in Entity extraction tasks.

References

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *COLING 2018, 27th International Conference on Computational Linguistics*, pages 1638–1649.

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, December.

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*.

Hiroshi Inoue. 2019. Multi-sample dropout for accelerated training and better generalization. *CoRR*, abs/1905.09788.

Jinfen Li, Zhihao Ye, and Lu Xiao. 2019. Detection of propaganda using logistic regression. In *Proceedings of the 2nd Workshop on NLP for Internet Freedom: Censorship, Disinformation, and Propaganda, pages 119–124 Hong Kong, China, November 4, 2019*, Hong Kong, China, November.

Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. *CoRR*, abs/1708.02002.

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.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *CoRR*, abs/1802.05365.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv e-prints*.

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

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

Shehel Yoosuf and Yin "David" Yang. 2019. Fine-grained propaganda detection with fine-tuned bert. In *Proceedings of the 2nd Workshop on NLP for Internet Freedom: Censorship, Disinformation, and Propaganda, pages 87–91 Hong Kong, China, November 4, 2019*, Hong Kong, China, November.