NLFIIT at SemEval-2020 Task 11: Neural Network Architectures for Detection of Propaganda Techniques in News Articles

Matej Martinkovic, Samuel Pecar, Marian Simko
Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovicova 2, 842 16 Bratislava, Slovakia
{xmartinkovicm1, samuel.pecar, marian.simko}@stuba.sk

Abstract

Since propaganda became more common technique in news, it is very important to look for possibilities of its automatic detection. In this paper, we present neural model architecture submitted to the SemEval-2020 Task 11 competition: "Detection of Propaganda Techniques in News Articles". We participated in both subtasks, propaganda span identification and propaganda technique classification. Our model utilizes recurrent Bi-LSTM layers with pre-trained word representations and also takes advantage of self-attention mechanism. Our model managed to achieve score 0.405 F1 for subtask 1 and 0.553 F1 for subtask 2 on test set resulting in 17th and 16th place in subtask 1 and subtask 2, respectively.

1 Introduction

Nowadays, many publishers provide articles online to reach their audience faster. This shift from traditional journalism via printed press enabled more extensive misuse of objective news for own agenda using different propaganda techniques. Propagandistic articles try to convey their message using different techniques. Propaganda is the spread of purposefully shaped information in media to promote some sort of agenda (Da San Martino et al., 2019). This kind of information is produced by increasing number of biased news outlets in attempt to sway or manipulate public opinion. However, there is also a connected problem to propaganda, online spreading of fake news and misinformation in general (Simko et al., 2019).

However, nowadays the widespread use of internet and access to multiple sources of information has caused traditional techniques of propaganda, like spreading of disinformation, to become far less effective. Along with this trend, propaganda techniques are naturally getting more complex to maximize their impact. Use of logical fallacies and appealing to the emotions of the audience is getting much more prominent because these techniques are significantly harder to spot at first glance and require much deeper analysis to get revealed.

Research on detecting propaganda has focused primarily on news articles (Rashkin et al., 2017; Barrón-Cedeno et al., 2019). However, analysis of articles produced by propagandistic news outlets indicates that these sources can often publish objective articles in attempt to increase their credibility (Horne et al., 2018). Recently the attention has moved towards fragment and sentence level propaganda detection. A new problem has been introduced – to detect the use of specific propaganda techniques in articles along with large corpora to assess this problem (Da San Martino et al., 2019). To tackle this problem many different neural model architectures were proposed, such as fine-tuned BERT pre-trained model (Yoosuf and Yang, 2019), ensemble of BERT, Bi-LSTM, and XGBoost (Al-Omari et al., 2019), or ensemble of logistic regression, CNN and BERT (Gupta et al., 2019).

The main goal of this SemEval task was to develop tools for automatic detection of propaganda and consisted of two subtasks (Da San Martino et al., 2020):

- *Subtask 1 - Span Identification* was focused on searching for propaganda spans in a given plain-text document. Propaganda span was defined as offset of character, where the span begins, and offset of character, where it ends. Overlapping spans would be then merged together.

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 1771–1778
Barcelona, Spain (Online), December 12, 2020.

- *Subtask 2 - Technique Classification* was focused on identifying propaganda technique in the given propagandistic text fragment. Data has been annotated with 18 techniques, but some of the classes were merged together due to their low frequency, which resulted in 14 classification classes for this subtask.

In this paper, we present our proposed neural network architecture consisting of different types of neural layers. We experimented with different word representations, such as ELMo, GloVe, and different encoder layers, such as GRU, LSTM, and also with transformer model BERT. We report description and results for multiple model architectures for both subtasks in the following sections.

2 Model

We experimented with multiple model setups for each subtask. We tried out multiple pre-trained embeddings, which fed Bi-LSTM layers with word representations (Pecar et al., 2019). For subtask 2, we also experimented with a fairly big pre-trained transformer model. The general architecture portraiting a structure of our proposed model is shown in Figure 1.

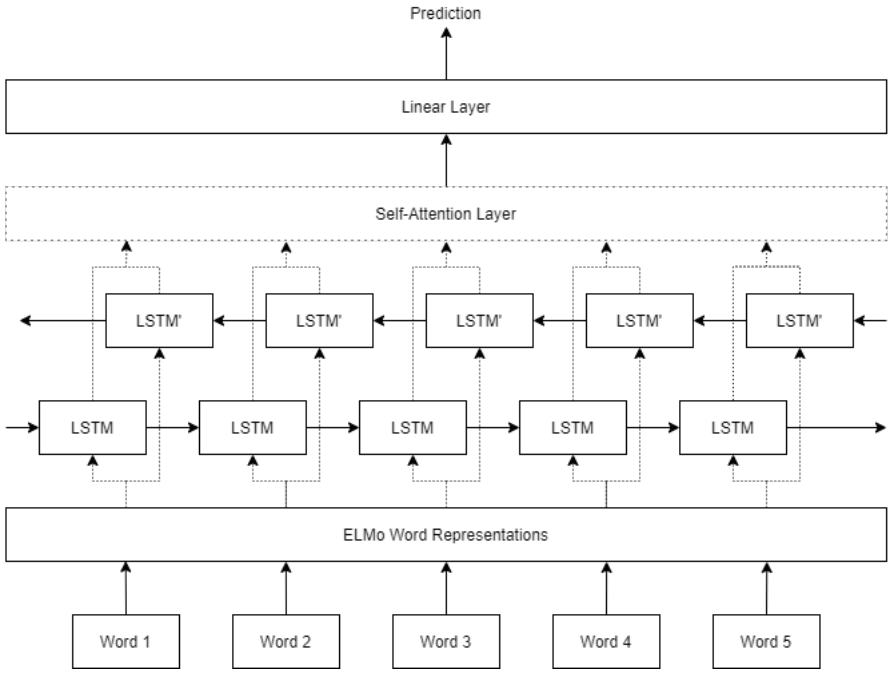

Figure 1: The proposed model architecture

Preprocessing Preprocessing can be considered as an important step in each natural language processing task. The dataset for this task consist of news articles thus the input text is not as much noisy as the texts from social media but still contains significant number of tweet footers, timestamps, web surveys, hyperlinks, and advertisements. In attempt to reduce number of input tokens that are very likely not propaganda, we decided to partly or completely remove these samples. In addition, we also removed all non-Latin characters, such as Hebrew, emoji characters, and also twitter mentions. We also substituted unicode quotation marks, apostrophes and hyphens with their ASCII equivalents (Pecar et al., 2018).

Word Representation To be able to feed input samples into models, we need to use appropriate word representations. In our experiments, we used different word representations to represent those samples. First, we used deep contextualized word representation also known as ELMo (Peters et al., 2018) along with its available pre-trained model. We also experimented with an unsupervised learning algorithm for obtaining vector representations for words - GloVe (Pennington et al., 2014). In subtask 2, we also experimented with transformer model known as BERT (Devlin et al., 2019) and its pre-trained model (Wolf et al., 2019).

Encoder Layer Word representation produced by pre-trained embeddings are fed to encoder layer. After initial experiments with both, unidirectional and bidirectional GRU (Cho et al., 2014) and LSTM layers (Hochreiter and Schmidhuber, 1997), we decided to continue with only bidirectional LSTM (Schuster and Paliwal, 1997) in our experiments. We also tried to tweak the size of a hidden state (sizes of 128, 256, 512 and 1024 were used), but it resulted in negligible performance change. For both subtasks, we also experimented with different number of stacked encoder layers (2 and 3 stacked layers were considered), but the most stable results was produced by single encoder layer. To enhance performance of LSTM layers for long input samples we also experimented with the self-attention mechanism on top of the encoder layer.

Decoder Layer We used standard linear layer to decode output representation of both setups of recurrent layers without and with self-attention mechanism for both tasks. We also experimented with conditional random fields (CRF) for subtask 1. The results especially in combination with GloVe were highly unstable regarding to the run and we decided not to continue in further experiments with CRF.

Loss Function As a loss function, we used standard cross-entropy loss. In addition, we also experimented with weight setup for classes in both subtasks, but unfortunately this yielded slightly worse results especially in subtask 1, due to massive class imbalance of propaganda tokens versus non-propaganda tokens.

Model Ensemble In order to further improve the performance, we decided to use model ensemble. Models were trained for up to 6 epochs and we summed predictions of multiple epochs (all epochs after 2 initial ones) and picked the class with maximum value. This method balanced precision and recall metrics in subtask 1, however it did not seem to have an impact on performance for subtask 2, thus we did not further investigate this option for subtask 2.

Regularization In attempt to prevent overfitting on the train dataset, we used dropout regularization which we applied to embedding and encoder layers. We also experimented with different values of learning rate which proved beneficial. Lowering learning rate value improved the performance of models using ELMo word representations (specific values are reported in Experimental Settings section).

3 Evaluation

In this section, we briefly provide information about the dataset, experimental settings used for our models and performance results of the proposed models.

3.1 Dataset

The dataset for this task consisted of around 550 news articles containing propaganda fragments that have been annotated with one of 18 propaganda techniques. As we mentioned, the dataset was noisy making the task more difficult. Table 1 shows basic information about the dataset, for more information see the task description paper (Da San Martino et al., 2020).

Dataset	Article count	Total sentences	Average # sentences in an article
Trainset	371	17484	47.13
Devset	75	3262	43.49
Testset	90	3377	37.52
Total	536	24123	45.01

Table 1: Basic Information about the Dataset

Original dataset annotation contained only identified span (offset of starting and ending character of fragment) for propaganda techniques. In order to be able to feed inputs to the network we can code each character whether it is part of propaganda or not. In table 2, we depict an example of the annotated string from the dataset. Characters are considered a propaganda and they are labeled 1 when their position

is inside one of propaganda spans. Characters not belonging in any span are labeled as 0. A character position is determined by its offset from the beginning of an article.

Character	h	o	w		s	t	u	p	i	d		a	n	d		p	e	t	t	y		t	h	i	n	g	s
Propaganda	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0

Table 2: An Example of Character Annotated String

However, in order to be able to use pre-trained word representations trained on large-scale datasets and to transfer knowledge about input words, we decided to transform the annotations from character to word level (see Table 3). For every word, we saved offset of its beginning character (included) and offset of its ending character (not included) and replaced the annotation on word level. A word would be then labeled as propaganda or not propaganda based on whether its span intersects one of the propaganda spans or not. After final prediction for words being part of propaganda, character offsets were reconstructed.

Word	how	stupid	and	petty	things
Propaganda	0	1	1	1	0
Span	0-3	4-10	11-14	15-20	21-27

Table 3: An Example of Word Annotated String

3.2 Experimental Settings

For our proposed models, we experimented with many hyperparameters which could be tweaked in process of development. For both subtasks, we trained model for 6 epochs, and took the best performing model measured by F1 score on the dev set. For encoder single bidirectional LSTM layer with hidden size of 128 and 512 was considered for subtask 1 and 2, respectively. We also experimented with sizes of 128, 256, 512 and 1024 along with multiple stacked layers (2 and 3) for both subtasks.

As optimizer, we used Adam (Kingma and Ba, 2015) for both tasks, the learning rate for the first subtask is set to 0.0005 and for subtask 2 as 10^{-3}, while experiments also with default values of 10^{-4} were considered. Batch size for subtask 1 was set to 64 and for subtask 2 to 32.

Dropout was used as a regularization technique for both tasks. We used different dropout values in different model architectures. After ELMo layer default value 0.5 was used, after GloVe layer only 0.2 dropout value was considered. Value 0.2 was also applied to the output of LSTM layers.

3.3 Results

For both subtasks, we experimented with several different architectures. Table 4 shows basic information about the architecture and performance comparison of models for subtask 1. Evaluation metrics for this subtask are computed by the function that gives credit to partial matching between two spans. For detailed description see task description paper (Da San Martino et al., 2020). We experimented with both GloVe and ELMo word representation, the latter proved to perform significantly better. We also tried to make use of self-attention mechanism on top of Bi-LSTM layers which performed significantly worse for substask 1. We also experimented with model using conditional random fields in combination with ELMo and Bi-LSTM.

Model	F1	Precision	Recall
ELMo-BiLSTM	**0.4107**	0.3528	**0.4914**
ELMo-BiLSTM-Att	0.2231	0.1679	0.3325
ELMo-BiLSTM-CRF	0.3988	**0.3791**	0.4207
GloVe-BiLSTM	0.3739	0.3658	0.3824
GloVe-BiLSTM-CRF	0.3458	0.3359	0.3563

Table 4: Performance of Tested Model Architectures on Dev set of Subtask 1

Table 5 shows that ELMo performed better than GloVe also in subtask 2. Due to class imbalance, the evaluation measure for this subtask is micro-averaged F1 measure. However, self-attention mechanism on top of Bi-LSTM, in contrary to subtask 1, improved performance significantly in subtask 2. We also experimented with BERT pre-trained model, but due to high memory requirements we could not fully utilize the biggest pre-trained model that could potentially outperform our proposed model for subtask 2.

Technique	ELMo-BiLSTM	GloVe-BiLSTM	ELMo-BiLSTM-Att	Glove-BiLSTM-Att	BERT transformer
Appeal to Authority	0.0000	0.0000	0.0000	0.0000	**0.1621**
Appeal to fear/prejudice	0.0000	0.0000	0.2292	0.1290	**0.2899**
Bandwagon, Reductio ad hitlerum	0.0000	0.0000	**0.5714**	0.0000	0.3077
Black and White Fallacy	0.0000	0.0000	0.0000	0.0000	**0.1482**
Causal Oversimplification	0.0000	0.0667	0.2400	0.0000	**0.2778**
Doubt	0.1875	0.1481	0.4948	0.4211	**0.5395**
Exaggeration, Minimisation	0.0286	0.0286	**0.4445**	0.3333	0.4143
Flag Waving	0.0000	0.0000	0.6474	0.6592	**0.6861**
Loaded Language	0.4778	0.4778	**0.7518**	0.7079	0.7434
Name Calling, Labeling	0.0000	0.0000	**0.7117**	0.6595	0.6945
Repetition	0.0000	0.0000	0.3755	0.3695	**0.3857**
Slogans	0.0000	0.0000	0.4242	0.3000	**0.4638**
Thought-terminating Clichés	0.0000	0.0000	0.0000	0.0000	**0.0833**
Whataboutism, Straw Men, Red Herring	0.0000	0.0000	0.0645	0.0000	**0.1304**
Micro-F1	0.3133	0.3114	**0.5767**	0.5353	0.5739

Table 5: Performance of Tested Model Architectures on Devset of Subtask 2

3.4 Submitted Models

In Table 6, we provide information about our submitted models and their performance on the test set for both subtasks. For both subtasks, we chose ELMo word representations as embedding layer followed by single Bi-LSTM encoder layer. As we mentioned, LSTM hidden size seemed to not have an impact on performance and we settled with 128 units for subtask 1 and 512 units for subtask 2. On top of Bi-LSTM, we also used self-attention layer for subtask 2. For both subtasks we chose standard linear layer as decoder.

Subtask	Model	F1	Precision	Recall
Subtask 1	ELMo-BiLSTM	0.4057	0.5091	0.3373
Subtask 2	ELMo-BiLSTM-Att	0.5525	-	-

Table 6: Performance of Submitted Models on Test set for Both Subtasks

3.5 Further Results

In addition, we provide results on selected models for several model modifications, such as change class weight for loss function, and insight on model ensemble performance. In Tables 7 and 8, we provide performance of the best model for each task with different weight classes in loss function. For subtask 1, ELMo-BiLSTM model was considered while for subtask 2 ELMo-BiLSTM-Att model was used. As we can observe, higher weight of minority class has slightly improved the performance on recall measure but

several points worsen on precision which resulted in lower performance in F1 score. For subtask 2, using distribution of the train set for class weights showed the best performance .

Minority class weight	Majority class weight	F1	Precision	Recall
1.00	1.0	**0.4107**	**0.3528**	0.4914
1.05	1.0	0.3955	0.3261	0.5024
1.10	1.0	0.3893	0.3150	**0.5097**

Table 7: Performance of ELMo-BiLSTM Model on Devset of Subtask 1 with Different Class Weights

Class weight	F1
equal	0.5767
distribution of training set	**0.5861**
distribution of dev set	0.5710

Table 8: Performance of ELMo-BiLSTM-Att Model on Devset of Subtask 2 with Different Class Weights

In Table 9, we show the performance of ELMo-BiLSTM for subtask 1, where model ensemble was used. The results showed that a single best epoch has a better performance than prediction ensemble of several epochs. We would need to evaluate model on the test set according to the results on the dev set to be able to fully adjust those results.

	F1	Precision	Recall
Ensemble of 4 epochs	0.4047	0.3683	0.4490
best single epoch	0.4107	0.3528	0.4914

Table 9: Comparison of Model Ensemble Performance

4 Conclusions

We proposed a neural model architecture for automatic detection of propaganda techniques in news articles. For both subtasks of the SemEval-2020 Task 11, we used ELMo embeddings, which fed word representations to Bi-LSTM encoder. For subtask 2, we also utilized the self-attention mechanism on top of the Bi-LSTM layer. We experimented with multiple stacked Bi-LSTM layers, however single layer proved to perform the best for both subtasks. Our experiments with embeddings showed significantly better performance of ELMo over GloVe for both subtasks. For subtask 1, we tried to employ CRF on top of Bi-LSTM with its perfomance coming closest to our proposed model. For subtask 2, we experimented with BERT pre-trained model that showed potential to outperform our proposed model, but due to high memory requirements it could not be fully utilized. We tried to improve performance of our proposed models by using model ensemble, but unfortunately our strategy seems not to have a significant impact on performance. Code for this submission is available via GitHub repository[1].

To obtain better results for subtask 1 we would probably try to fine-tune Bi-LSTM model with CRF and further experiment with ensemble of this model and BERT. As we mentioned, the biggest pre-trained BERT model utilized to its full potential could outperform our model for subtask 2. Another interesting possibility would be ensemble of BERT and Bi-LSTM with the attention.

Acknowledgments

This work was partially supported by the Slovak Research and Development Agency under the contracts No. APVV-17-0267 and No. APVV SK-IL-RD-18-0004 and the Scientific Grant Agency of the Slovak Republic, grant No. VG 1/0667/18 and grant No. VG 1/0725/19.

[1] https://github.com/NL-FIIT/NLFIIT-semeval20-task11

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, Hong Kong, China, November. Association for Computational Linguistics.

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

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, October. Association for Computational Linguistics.

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. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June. 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*, pages 92–97, Hong Kong, China, November. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Benjamin D. Horne, William Dron, Sara Khedr, and Sibel Adali. 2018. Sampling the news producers: A large news and feature data set for the study of the complex media landscape.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Samuel Pecar, Michal Farkas, Marian Simko, Peter Lacko, and Maria Bielikova. 2018. NL-FIIT at IEST-2018: Emotion recognition utilizing neural networks and multi-level preprocessing. In *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 217–223, Brussels, Belgium, October. Association for Computational Linguistics.

Samuel Pecar, Marian Simko, and Maria Bielikova. 2019. NL-FIIT at SemEval-2019 task 9: Neural model ensemble for suggestion mining. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 1218–1223, Minneapolis, Minnesota, USA, June. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October. Association for Computational Linguistics.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June. 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*, pages 2931–2937, Copenhagen, Denmark, September. Association for Computational Linguistics.

Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Jakub Simko, Martina Hanakova, Patrik Racsko, Matus Tomlein, Robert Moro, and Maria Bielikova. 2019. Fake news reading on social media: An eye-tracking study. In *Proceedings of the 30th ACM Conference on Hypertext and Social Media*, HT '19, page 221–230, New York, NY, USA. Association for Computing Machinery.

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.

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.