To Context or Not to Context? Analysis of Non-contextualized Word Embeddings in Propaganda Detection Task

Luka Barišić, Marko Pisačić, Filip Radović

University of Zagreb, Faculty of Electrical Engineering and Computing
Unska 3, 10000 Zagreb, Croatia
{luka.barisic, marko.pisacic, filip.radovic}@fer.hr

Abstract

In this paper, we explore the impact of non-contextualized word embeddings on the sequence labeling task of propaganda fragment identification. More specifically, we evaluate the addition of non-contextualized word embeddings to contextualized ones, testing the hypothesis that they will give additional, more general information to the model. The results do not show an improvement of statistical significance, however, we discuss potential reasons behind them, and propose further research to analyze this problem more deeply.

1. Introduction

Propaganda is a mode of communication used to manipulate or influence the opinion of groups to support a particular cause or belief. Though its use is not exclusively negative, propaganda very often involves a heavy emphasis on the benefits and virtues of one idea or group, while simultaneously distorting the truth or suppressing the counter-argument.

With the dawn of the internet, propaganda has become very influential and a part of our daily lives. Propaganda is usually written in a way that can't be detected by the reader. That can become dangerous because it can shift the reader's perception without the reader knowing it. Because of its huge impact and the danger it poses to our lives, much effort has been put into stopping and detecting propaganda.

One limitation of current approaches to propaganda detection is that they use limited knowledge about outside, more general information. That poses a problem because propaganda often subtly changes the meaning of the words in a sentence. By not knowing the meaning outside that sentence, we deprive ourselves of information.

Our idea is to construct general word representations that may bring additional information about the base meaning of the word as a contrast to the meaning of the word in a particular context. For example, consider the propagandist sentence: "Entering this war will make us have a better future in our country". We expect the usually negative connotation of the word 'war' to be alleviated by its context, more precisely by the phrase 'better future'. Our assumption is that this would be reflected in the word embeddings, so the contextualized and non-contextualized embeddings would be different. We expect that this difference will behave in a more specific way when dealing with embeddings of a propaganda fragment. Our model should then be able to capture those specifics and be better informed which would result in a better performance.

In section 2., we give an overview of the work done on propaganda detection so far. We evaluate our new embeddings on the task of fragment-level propaganda detection. Our model is explained in section 3. We also elaborate on how certain layers of BERT behave and the type of information they encode. Although our results don't show any improvements over using only standard embeddings, we argue why the concept should be further explored.

2. Propaganda Detection in News Articles

Recently, there has been a lot of interest centered around detecting propaganda. There are multiple types of propaganda detection tasks, but they can all be approached in similar ways. Barrón-Cedeño et al. (2019) showed that modeling writing style and text complexity are very effective in detecting propaganda on the article level. They argue that articles that contain propaganda have different writing styles compared to normal articles. With the release of powerful pre-trained models like BERT and ELMO, researchers have tried to tackle more fine-grained tasks of propaganda detection. Da San Martino et al. (2019) released a corpus that contains sentence- and fragment-level labels. They proposed a novel multi-granularity neural network which was trained on both sentence- and fragment-level data and achieved state of the art results. By solving two problems at the same time, the model was able to combine the knowledge that was learned from both tasks. Yoosuf and Yang (2019) used a pre-trained BERT model and fine-tuned it on the fragment-level classification task. They also observed that solving the class imbalance problem improved their model drastically. We will see that our model also takes the problem of imbalance into consideration. Hou and Chen (2019) used article titles as additional context for their BERT model. They argued that the titles could serve as a summary of the whole article thus provide additional knowledge to the model.

Similar to propaganda detection, there has also been a new wave of research centering around analyzing and interpreting state-of-the-art natural language processing models. Tenney et al. (2019) show that specific parts of BERT layers encode different information. They reported that basic, syntactic information appears in the lower layers, while more abstract, semantic information appears at higher layers. We investigated those differences in layers and based our model on the results which will be discussed in a future section.

Propaganda detection is a complicated problem that doesn't have one clear solution. Although previous work

Table 1: Vector similarities of the word killed compared to the meaining in the sentence ”Someone was killed in a horrifying manner”. Cosine refers to cosine similarity, and Euclid to Euclidean distance

Text	Cosine	Euclid
He brutally killed someone	0.7988	43.771
He accidentally killed someone	0.7915	44.649
He killed someone	0.7995	43.830

is fairly successful, all the methods mentioned above are too focused on only extracting information from the corpora they are dealing with. We argue that adding a more general representation of the text information can add better insights to the model.

3. Experimental Setup

3.1. Dataset

The data for our task is supplied by the Propaganda Analysis Project [1], as a part of SemEval 2020 shared task 11: ”Detection of propaganda techniques in news articles”. The task is split into two parts: identification of text fragments containing propaganda, and classification of these fragments into one of 18 specific propaganda techniques, such as whataboutism, red herring, and name-calling. We chose to focus only on the first part — propaganda fragment identification, as we found it more interesting to determine what differentiates propaganda from non-propaganda, rather than improving on the well-researched multi-class classification problem.

Since the mentioned task was active during the implementation of our model, we were unable to obtain the official test set, so we decided to combine train and validation sets into one dataset, and then did our own train-validation-test splits. The dataset consists of 445 news articles with manually annotated text fragments containing a propaganda technique. There are 19 830 sentences in total, which we split into tokens with a simple tokenizer. Out of a total number of 401,705 tokens, only 51,265 (12,8%) of them were part of fragments labeled as propaganda, and the remaining 350 440 (87,2%) as non-propaganda.

3.2. Embedding Creation

Input tokens (words, sub-words, and other parts of a sentence) are represented as numeric vectors called word embeddings. An embedding for a word can either be dependent on the context of the word (dynamically generated), or not (static). As discussed in Section 2., we would like to explore the impact of adding non-contextualized word embeddings to our baseline model, which is using contextualized embeddings only.

To obtain non-contextualized embeddings for all words in our corpus, we used the ”All the news” dataset provided by Andrew Thompson.[2] First, we found all occurrences of a certain word in the ”All the news” dataset. We then averaged the contextualized BERT embeddings of all those occurrences to obtain the non-contextual embedding. The idea is that by considering all possible contexts of the word, the contextual information fades away and all that remains is the non-contextual meaning of the word.

The obtained non-contextual embeddings are concatenated with contextualized BERT embeddings, and fed into our model as input. We expect the model to perform better than without non-contextualized embeddings. To provide further insights we also consider two simpler models which we expect to perform worse than both the baseline and our model. The first one uses non-contextualized embeddings only, and the second one uses another type of well-known static embeddings – GloVe.

3.3. Embedding Exploration

In this subsection, we will do an analysis of created non-contextual embeddings. To better understand how the embeddings compare to the contextual embeddings, we compared how similar are the contextual and non-contextual embeddings of the same word. The average cosine distance between those two embeddings in every BERT layer is shown in Figure 1. We can observe that the embeddings are most distant at higher layers of the BERT architecture, which guided our decision to use the average of only the four top-most layers in both our contextualized and non-contextualized embeddings. We further observed that the embedding similarities are different based on the contexts the words are in. In Table 1 we can see how this plays out with some real-world example sentences.

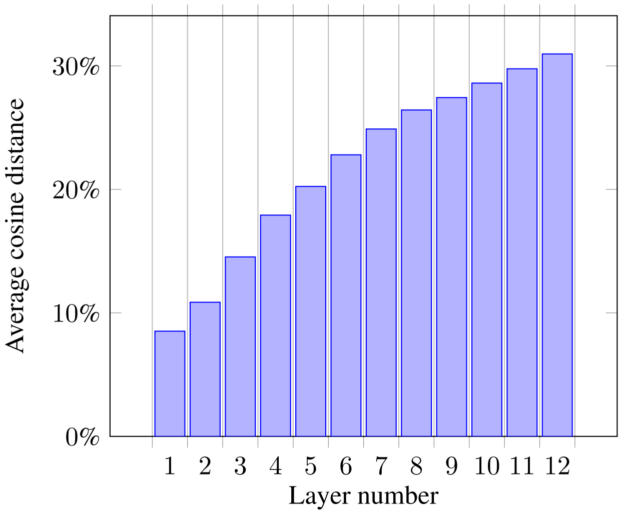

Figure 1: Average cosine distance between contextualized and non-contextualized embedding for each BERT layer

3.4. Model Architecture

The architecture of the model can be seen in Figure 2. Since our features are comprised of concatenated word embeddings, on the entrance of our model there are two word encoders — BERT, and the non-contextualized custom encoder, described in a previous subsection. BERT (Bidirectional Encoder Representations from Transformers) is a

[1] https://propaganda.qcri.org/index.html

[2] https://www.kaggle.com/snapcrack/all-the-news

transformer model which uses attention mechanism to learn contextual relation between words (or sub-words) in a sequence. In that way, for each input token, BERT analyses its context and produces a dynamic embedding. We use that output as the first half of our feature vector. The second half is the output of the second, custom encoder, which in this case performs a simple lookup function for every token.

After the embeddings of the token are obtained, they are sent to the bi-directional LSTM (BLSTM). We chose this variant of a recurrent neural network because of its ability to capture time dynamics in sequences, and has proved effective in other sequence labeling tasks. Before the BLSTM, there is a fully-connected linear layer with an equal number of inputs and outputs. After the BLSTM, there is also a linear layer with two outputs decoding the probabilities of non-propaganda and propaganda classes for that token. Optimization of the network parameters is done using stochastic gradient descent, with a cross-entropy loss function.

Hyperparameters for this model can be found in Table 2. Because of the lack of resources to properly train all models, we optimized hyperparameters only on one of the baselines and used them for all models. The optimal hyperparameter values for that baseline were found using a simple grid-search optimizer.

3.5. Evaluation

The main evaluation metric for our problem is a custom, task-adjusted F1 score used in the SemEval on which we base our paper. The gist of the metric is that it deals with normalized overlaps of fragments s and t given by the function C:

$$C(s,t,h) = \frac{|s \cap t|}{h} \tag{1}$$

where h is the normalizing factor. We can think of C being the precision on an individual predicted propaganda fragment if the normalizing factor is $|s|$, or recall on an individual reference propaganda fragment if the normalizing factor is $|t|$. Overall precision is then calculated by averaging precisions on individual predicted fragments. Calculation is analogous for overall recall. For a set of predicted fragments S and a set of reference fragments T the formulae for overall precision and recall are:

$$P(S,T) = \frac{1}{|S|} \sum_{\substack{s \in S \\ t \in T}} C(s,t,|s|)$$

$$R(S,T) = \frac{1}{|T|} \sum_{\substack{s \in S \\ t \in T}} C(s,t,|t|)$$

The main evaluation is F1 metric, the usual harmonic mean of the overall precision and recall.

4. Results

The training and evaluation of models with each of the four combinations of word embeddings was done on 10 different train-validation-test splits. Each dataset split was fixed during the training of all four models so the results for the split are directly comparable. Our null hypothesis is that the results with the contextual embeddings alone are at least as good as the results with the added non-contextual embeddings. We would like to see our null hypothesis rejected with the significance level $\alpha = 0.05$. That way we could conclude that the addition of non-contextual embeddings improved the performance of our model with a statistical significance.

Table 2: Hyperparameters used during model training.

Hyperparameter	Value
BLSTM state size	200
Dropout	0.2
Initial learning rate	0.05
Annealing factor	0.2
Patience	1
Epochs	5
Batch size	10

Table 3: Mean values for precision, recall, and F1 score for each model. Combined model is our proposed model, which uses both non-contextualized and BERT embeddings. Sample size N = 10

Model	Precision	Recall	F1
GloVe	44.83	8.06	13.31
Non-contextualized	30.95	29.65	29.18
BERT	33.11	49.91	38.47
Combined model	36.36	43.13	38.76

The mean values for the contest specific precision, recall and F1 measures for each embedding combination are given in Table 3. Although we observe both a better mean and pair-wise performance of our model compared to the baseline, the t-test was performed which showed that, unfortunately, the improvement is not statistically significant ($p = 0.351 > \alpha$). As was expected, the contextualized-embedding baseline outperforms both non-contextualized and GloVe embeddings on their own ($p < 10^{-4}$).

We can assume a couple of reasons for the lack of statistically significant improvement in performance. Since hyperparameters were optimized on a baseline which included only contextualized embeddings, it seems probable that the doubling of input feature size has caused our model performance to suffer. Double input size could increase the capacity our model needs to have, and hence demands additional tweaks to our model and hyperparameters. Also, our custom non-contextualized word embeddings aren't ideal. What we tried to do was to capture only the basic meaning of a word in a particular sentence. What was actually done was averaging across word occurrences in a big dataset. In that way, the most frequent word meaning dominates the embedding and isn't tailored in any way to the meaning in a specific sentence. We also weren't able to fine-tune BERT layers due to computational limitations.

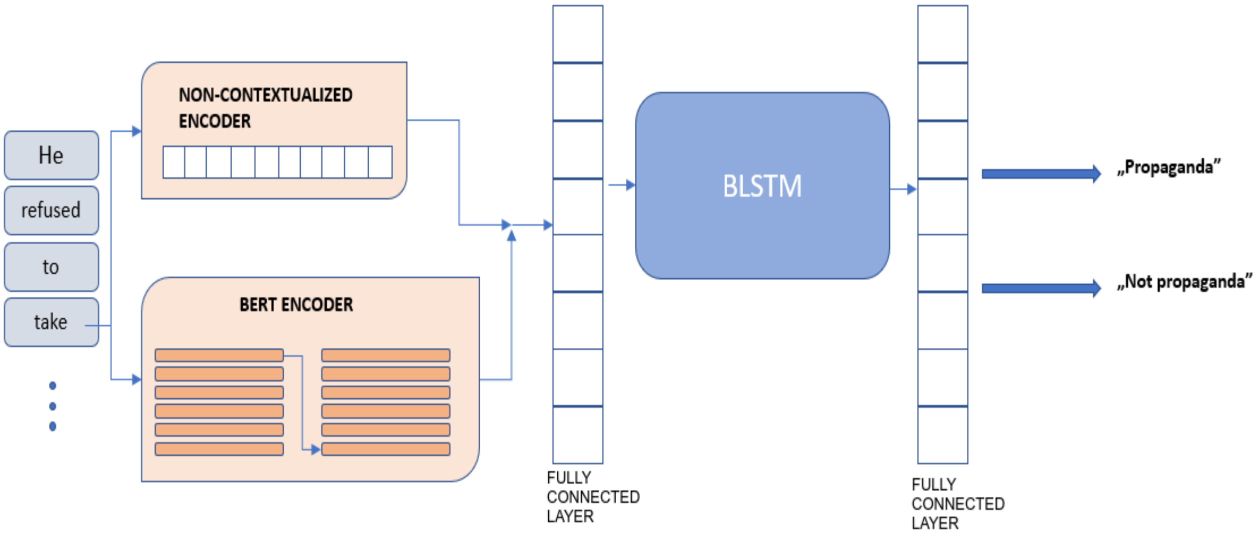

Figure 2: Model architecture

5. Conclusion

This paper investigated the task of propaganda detection, and described it as a sequence labeling task. We explored the role of non-contextualized embeddings in a propaganda fragment identification task. We proposed a way to construct custom, non-contextualized BERT embeddings and evaluated their impact on the performance of sequence labeling. Our hypothesis was that propaganda subtly changes the meaning of words and that by utilizing non-contextualized embeddings the model can detect those differences. Our embedding exploration confirms that changes in context can be detected using BERT embeddings. Although our idea initially looked promising, our results were inconclusive and we didn't manage to outperform our hard baseline that used only contextualized embeddings. We argue that the fault may lie in our custom non-contextualized embeddings which were too static and couldn't capture the base meaning of the word in a sentence well. We were also limited by computational resources, so the model training could certainly be improved in further work. Even though we didn't reach any conclusive support for our hypothesis, we believe that non-contextualized embeddings could have their role in propaganda detection. In our future work, we will test how other non-contextual embeddings work with BERT embeddings. We will also experiment with more model architectures that can better capture the underlying structure of our data.

References

Alberto Barrón-Cedeño, 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.

Giovanni Da San Martino, Seunghak Yu, Alberto Barrón-Cedeño, Rostislav Petrov, and Preslav Nakov. 2019. 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)*, pages 5636–5646, Hong Kong, China, November. Association for Computational Linguistics.

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.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. *CoRR*, abs/1905.05950.

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.