BERT, RoBERTa and T5 transformers implementations for Detection of Propaganda Techniques in News Articles.

Intermediary Paper Report

Progress report notes

Since the project proposal was submitted on November 1st, our team focused on designing an architecture that could handle `RoBERTa` and `T5` technologies for propaganda detection. We propose below a novel approach to solve both the span identification (`SI`) and the technique classification (`TC`) sub-tasks.

Although the result section is still mostly empty in this paper, we already have functional running models for both `SI` and `TC` tasks. Our entire code work is visible on our public GitHub repository[1].

Reshaping the scope of the project

Recall now that the objective we defined in our project proposal was two-fold:

1. To implement different models able to automatically detect the use of propaganda techniques in text snippets, accomplishing both the `SI` and `TC` sub-tasks of the shared task.

2. To compare the implemented models and draw conclusions on their performance through an error analysis for each of them.

We proposed to fulfill Goal (1) with the following:

(a) A small self-trained language model to provide a baseline performance we can compare other models to.

(b) `RoBERTa` (Liu et al., 2019), a state-of-the-art pre-trained model based on the Transformer architecture.

(c) `T5` (Raffel et al., 2020), another state-of-the-art Transformer based architecture that uses a text-to-text approach.

So far, we have spent our time designing and implementing a modular architecture, which would allow us to smoothly integrate different models. Because we are confident in our ability to further enhance our architecture, we have decided to focus on pre-trained language models (PLMs), and not to develop our own self-trained language model from scratch (as said in (a)). Therefore, we will use `BERT` as our new baseline model, to compare the effects of different `PLMs`.

Remaining objectives

To conclude the project, we would like to:

1. Solve the problems we're currently facing with our architecture.

   (a) Fine tuning the weights for the `SI` model, introduced to combat class imbalance, our model now predicts too many spans as being propaganda (this behaviour can be observed interpreting the results in Table 1).

   (b) Memory management problems in PyTorch, that prevent us from successfully training our model using GPUs.

   (c) Find correct weights for our `TC` model to combat class imbalance.

2. Fine-tune our architecture using different state-of-the-art `PLMs`, changing hyperparameters and adding regularization methods.

3. Train multiple epochs of our models on the Euler ETH cluster GPUs to better fit them.

4. Implement a final predictions writer to create human readable output, allowing us to feed our model any kind of text to retrieve the fragments classified as propaganda from it, and the technique associated with each of them.

5. Possibly exploring the implementation of add-on features to our architecture such as conditional random field (`CRF`) or data augmentation techniques such as *back translation*, *random replacement* and *random insertion* in order to further enhance the results.

6. Compare the implemented final models and draw conclusions on their performance through a complete error analysis.

[1] https://github.com/andreakiro/nlpropaganda

BERT, RoBERTa and T5 transformers implementations for Detection of Propaganda Techniques in News Articles.

Natural Language Processing Project Paper, Fall 2021

Antoine Basseto, Giacomo Camposampiero and **Andrea Pinto**
ETH Zurich - Swiss Federal Institute of Technology
{abasseto, gcamposampie, pintoa}@ethz.ch

Abstract

This paper describes the design of our system contributing to the Task 11 of SemEval-2020 (Martino et al., 2020a) aiming to detect propaganda techniques in news articles. We investigate a novel approach allowing the technique classification task (TC) to work under relaxed assumptions and be more easily applicable to real-world scenarios, leading to changes in the span identification task (SI) as well. Both models are built on top of heterogeneous pre-trained language models (PLMs) such as BERT, RoBERTa and T5. The described architecture achieved an F_1-score of 0.29651 on the SI task (ranking $34/45$) and an F_1-score of X on the TC task (ranking $X/45$).

1 Introduction

The proliferation of online misinformation has led to a significant amount of research into the automatic detection of fake news (Shu et al., 2017). However, most of the efforts have been concentrated on whole-document classification (Rashkin et al., 2017) or analysis of the general patterns of online propaganda (Garimella et al., 2018; Chatfield et al., 2015), while little has been done so far in terms of fine-grained text analysis. This approach could complement existing techniques and allow the user to extract more informed and nuanced judgment on the piece being read. Moreover, it could also inform journalists on the pitfalls they might be falling into when writing articles.

In this context, Task 11 of SemEval-2020[1] (Martino et al., 2020a) aims to bridge this gap, facilitating the development of models capable of spotting text fragments where a defined set of propaganda techniques are being used. This shared task provides a well-annotated dataset of 536 news articles, which enables the participant to develop detection models that automatically spot a defined range of 14 propaganda techniques in written texts.

The focus of the task is broken down into two well-defined sub-tasks, namely (1) *Span identification* (SI) to detect the text fragments representative of a propaganda technique in the news articles and (2) *Technique classification* (TC) to detect the propaganda technique used in a given text span.

2 Related Work

2.1 Literature review

Literature regarding fine-grained propaganda detection and analysis has known a significant development only in the last few years, mostly thanks to the different shared tasks that covered this particular topic (Da San Martino et al., 2019a; Martino et al., 2020b).

One of the first contributions can be traced back to (Da San Martino et al., 2019b), which proposed a BERT-based model to detect propaganda spans and to classify their techniques. In the NLP4IF-2019 shared task, the participants used pre-trained language models (PLMs), LSTMs and ensembles to tackle the problem of fine-grained propaganda classification (Yoosuf and Yang, 2019; Vlad et al., 2019; Tayyar Madabushi et al., 2019). Also in SemEval-2020 most of the winning teams solutions relied on Transformers and ensembles (Chernyavskiy et al., 2020; Morio et al., 2020; Dimov et al., 2020; Jurkiewicz et al., 2020).

Our work is especially related to the cited studies of winning teams of the SemEval-2020 shared-task. We decided to use the same PLMs as the other teams, with the addition of T5, proposed by (Chernyavskiy et al., 2020) as an avenue of future work. However, we differ by tackling the TC sub-task in a way none of the previous teams had explored, leading to other subtleties in the SI sub-task as well.

[1] The official task webpage: https://propaganda.qcri.org/semeval2020-task11/

2.2 Pre-Trained Language Models (PLMs)

In this study, three different types of Transformer-based PLMs (Vaswani et al., 2017) were used to tackle the tasks (see Section 5.2).

BERT (Devlin et al., 2019) is the epoch-making Transformer-based masked language model. In our work, the $\text{BERT}_{\text{LARGE}}$ model was employed.

RoBERTa (Liu et al., 2019) is a fine-tuned BERT-based model where the authors investigated hyper-parameters and training data size. RoBERTa has achieved state-of-the-art results. In our work, the $\text{RoBERTa}_{\text{LARGE}}$ model was employed.

T5 (Raffel et al., 2020) is a state-of-the-art Transformer that uses a unified text-to-text approach. Encoder and decoder are similar in size and configuration to a BERT stack. In our work, the T5_{LARGE} model was employed.

2.3 Technology stack

We opted to implement our architecture in AllenNLP (Gardner et al., 2017), a recent NLP research library developed by the Allen Institute for Artificial Intelligence, built on top of PyTorch (Paszke et al., 2019) and SpaCy (Honnibal and Montani, 2017), for developing state-of-the-art deep learning models on a wide variety of NLP tasks.

3 Dataset

3.1 Data description

The dataset used for the task, PTC-SemEval20 corpus (Martino et al., 2020a), consists of a sample of news articles collected from mid-2017 to early 2019. The articles were retrieved from 13 propaganda and 36 non-propaganda news outlets, as labeled by Media Bias/Fact Check[2], and manually annotated by the organizers. The exact procedure of text labeling is discussed in depth in both (Da San Martino et al., 2019b) and (Martino et al., 2020a).

The training and validation part of the corpus are the same as those presented in (Da San Martino et al., 2019b). The test part of the corpus consists of 90 additional news article in respect to the original evaluation articles, retrieved and annotated using the same procedure as the original. In total, the collection consists of 536 news articles containing 8,981 propaganda spans, that belong to one of the fourteen possible techniques.

[2]https://mediabiasfactcheck.com

3.2 Data exploration

Some statistics about the corpus (e.g. the number of instances and the average length in terms of tokens/characters for each propaganda technique, the average length of articles and others) were already given by the organizers as part of the shared task description paper (Martino et al., 2020a).

In addition to this data, a more fine-grained exploration of the training corpus was performed as one of the first steps in the task tackling. The main reasons for this additional exploration were:

- To extract meaningful insights that could be used to infer robust and effective heuristics for span pruning in SI preprocessing, as discussed in Section 4.1.1.
- To justify some of our model architecture choices, especially for the SI model and its specificities we discuss in Section 4.1.

Some of the results of this analysis have been reported in Figures 1 and 2. Due to space constraints, other results (e-g. the distribution over token categories in gold spans and border tokens[3]), were omitted but can be accessed on our GitHub repository.

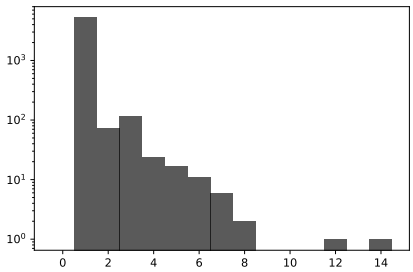

Figure 1: Number of sentences in training gold spans.

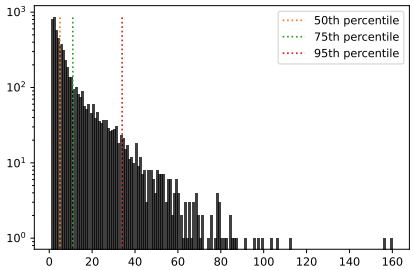

Figure 2: Number of tokens in training gold spans.

[3]Tokens near the beginning and end of a span.

4 System description

Our approach was motivated by considering a real-world use of the TC model. As described in the SemEval-2020 task, TC models are supposed to classify a span as one of fourteen possible propaganda techniques, but this assumes that TC models are always fed with spans that necessarily contain a propagandist argument. However, in a real-world scenario no such guarantees could be made, unless using a well-chosen list of manually selected spans.

Novel approach to architecture

This conclusion resulted in two major changes compared to the architecture proposed in the SemEval-2020 shared task, leading to an approach where the SI model is part of the preprocessing stage of TC:

1. TC model should train on the results provided by the SI model, and not on a given set of gold spans already known to be propaganda.

2. Because the SI model will make mistakes, the TC model should also be able to handle false positives and predict spans as "*Not Propaganda*", adding an extra 15th class.

To provide additional means of fine-tuning the final architecture, we also decided to consider the SI model as a span classification task rather than a sequence labeling task (see Section 4.1). This meant that for each possible span, the SI model assigns a probability of being a propagandist argument, and therefore lets the TC model only classify spans that have this propaganda likelihood exceeding a well-chosen threshold. This way, we can regulate the number of false positives we forward on to TC and make full use of the slackness offered by the added "*Not Propaganda*" class.

In this architecture, it could be argued that the addition of this new 15th label renders the SI model unnecessary, but its use has strong computational advantages in allowing us to counteract the very heavy class imbalance we would have if we were considering every possible spans in the TC task.

4.1 Span Identification (SI)

Span identification is often seen as a sequence labeling task, using Begin (B), In (I) and Out (O) labels to classify each token as being in, out, or the beginning of a span. Despite the fact that many teams have used this common technique to model the problem, we decided to go another route and see it as a *span classification* task. This means that we enumerate all possible spans in the article, filtering them with heuristics (see Section 4.1.1), and we classify each of those as being a propaganda span or not. Our reasons for approaching this problem that way are the following:

- To be able to use our SI model as intended in our general pipeline (see Section 4), we need a model that takes a span as input and classifies it as being propaganda or not, whereas a BIO-tagging scheme would take a text as input and output the predicted propaganda spans.

- Furthermore, as seen in Figure 1, a non-negligible number of gold spans span multiple sentences. In some implementations from other teams, such as (Dimov et al., 2020), using the BIO-tagging scheme meant they were training a model that worked on each sentence individually, and they had to split gold spans spanning multiple sentences, negatively impacting their model's performance.

4.1.1 SI Preprocessing

To deal with the exponential number of spans in an article, we used heuristics to filter-out as many of them as possible. First of all, we only consider spans of 20 tokens or less. According to Figure 2, that means we can cover about 80% of the gold spans in our training dataset. Second of all, we discard spans that consist exclusively of a combination of determinants, punctuation, space or EOL tokens, as we can safely assume those will not be propaganda.

4.1.2 SI Embeddings

After being extracted, spans are embedded. This is done by contextualising the text using a PLM, and using that to have a contextualized representation of our span. The contextualized representation of our span is the result of concatenating the embedding of the first token of the span, the last token of the span, and an average of the token embeddings in the span weighted by a normalized self-attention score. That last part is the result of a self attentive span extractor module from AllenNLP. Our results using different PLMs can be seen in Section 5.2.

4.1.3 SI Loss function

We are using the binary cross-entropy (BCE) loss to train our model. The use of the BCE loss is standard in binary classification tasks, but especially

relevant in our case. Indeed, since the outputs of the SI model will be used to prune spans given to the TC model, we are not only interested in the classification but in the actual confidence our model has in it, because we can change the confidence threshold for which we discard spans or not in TC.

A specificity of our approach is also that it is affected by an important imbalance between the two classes. Only a small fraction of the spans that are retrieved by the preprocessing stage effectively contain a propagandist argument. To deal with this problem and prevent the model from classifying every span as not propaganda, we introduce a weight for the positive class in the loss function, defined as follows:

$$weight_+ = \frac{\#\ spans\ to\ classify}{\#\ propaganda\ spans}$$

4.2 Technique Classification (TC)

The TC model has to label each element of a set S of spans with one of the 14 existing propaganda techniques. Note that this relies on the important assumption that the model is only provided with a set S of spans which contain a propagandist argument. Recall also that our overall architecture is designed to consider the real-world scenario where this assumption cannot be made (see Section 4). Our TC model was intended to be built on top of the results of the SI model. Consequently, we never have access to the ideal set S but rather a relaxed set S' of spans with the easier-to-satisfy assumption that $S \subset S'$. In order to correctly classify spans, we therefore had to add an extra label "*Not Propaganda*" for spans containing no propagandist argument (i.e. belonging to $S' \backslash S$).

4.2.1 TC Preprocessing

The key insight is that we can now think of the SI model as applying *an additional pruning* procedure on the set of possible spans.

For each article, we first apply the same preprocessing as we did for the SI model. Namely, we enumerate all spans following the same heuristics described in Section 4.1.1. We then use a pretrained SI model to get for each of those spans the probability of it containing a propagandist argument, and prune again according to those and a chosen threshold.

4.2.2 TC Embedddings

After being extracted and pruned according to the results provided by the pre-trained SI model, spans are embedded using the same techniques we employed for the SI span embedding stage (see Section 4.1.2). Our results using different PLMs can be seen in Section 5.2.

4.2.3 TC Loss function and metric

We are using the standard cross-entropy (CE) loss to train our model. As in the SI analog, this loss may suffer because of the design of our overall architecture. Indeed, depending on the threshold we set as a hyperparameter to filter the spans according to the results of the SI model in the TC preprocessing, we still could have much more false positives than real propaganda spans. This could lead to an important class imbalance and skew our model's predictions. To deal with this problem and prevent the model from classifying each of the new spans with the 15th label "*Not Propaganda*", we introduce weights for each of the classes in the loss function, defined as follows ***(see point 1.(c) of Remaining objectives)***.

5 Experiments

5.1 Experimental setup

The metric used to evaluate our SI model is a custom F_1-measure that allows non-zero scores for partial matches between predicted and gold spans, as proposed in (Martino et al., 2020a).

The metric used to evaluate the TC model is a standard micro-averaged F_1-measure.

5.2 Experimental results

Partial results obtained so far are included in Table 1.

5.3 Error analysis

A specific error analysis for both SI and TC will be included in the final paper.

- Analysis of portions of entirely missed, partially missed and entirely classified spans for each propaganda technique.
- Analysis of partially identified spans in SI, to identify possible areas of improvements with post-processing.
- Visualization of a confusion matrix for classes in the TC task, to have a simple representation of the which categories of propaganda are harder to classify for the model.

Model	Custom F_1	Precision	Recall
BERT			
RoBERTa			
T5			
BERT$_{test}$	0.29651	0.17528	0.96147

Table 1: Model results on SI task for validation and test data.

6 Conclusion

So far, results could be improved, but considering the initial stage of fine-tuning we are in, our approach seems to be of interest because of the advantages discussed throughout this paper.

6.1 Future Work

Because of the context of this project, and the time limit associated with it, we were not able to implement all of the ideas we had to improve our model. To build upon our work, we propose to look into the following:

- Our architecture could be slightly modified, to give as an extra argument for the TC model the confidence of the SI model for its predictions in the form of the probability for each span. We expect this to be an important feature in determining whether a span belongs to the "*Not Propaganda*" class or not for the TC model.

References

Akemi Takeoka Chatfield, Christopher G. Reddick, and Uuf Brajawidagda. 2015. Tweeting propaganda, radicalization and recruitment: Islamic state supporters multi-sided twitter networks. In *Proceedings of the 16th Annual International Conference on Digital Government Research*, dg.o '15, page 239–249, New York, NY, USA. Association for Computing Machinery.

Anton Chernyavskiy, Dmitry Ilvovsky, and Preslav Nakov. 2020. aschern at semeval-2020 task 11: It takes three to tango: Roberta, crf, and transfer learning. *CoRR*, abs/2008.02837.

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*, pages 162–170, Hong Kong, China. 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)*, pages 5636–5646, Hong Kong, China. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Ilya Dimov, Vladislav Korzun, and Ivan Smurov. 2020. Nopropaganda at semeval-2020 task 11: A borrowed approach to sequence tagging and text classification.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke S. Zettlemoyer. 2017. Allennlp: A deep semantic natural language processing platform.

Kiran Garimella, Gianmarco De Francisci Morales, Aristides Gionis, and Michael Mathioudakis. 2018. Quantifying controversy on social media. *Trans. Soc. Comput.*, 1(1).

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Dawid Jurkiewicz, Łukasz Borchmann, Izabela Kosmala, and Filip Graliński. 2020. Applicaai at semeval-2020 task 11: On roberta-crf, span cls and whether self-training helps them.

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.

Giovanni Da San Martino, Alberto Barrón-Cedeño, Henning Wachsmuth, Rostislav Petrov, and Preslav Nakov. 2020a. Semeval-2020 task 11: Detection of propaganda techniques in news articles. *CoRR*, abs/2009.02696.

Giovanni Da San Martino, Alberto Barrón-Cedeño, Henning Wachsmuth, Rostislav Petrov, and Preslav Nakov. 2020b. Semeval-2020 task 11: Detection

of propaganda techniques in news articles. *CoRR*, abs/2009.02696.

Gaku Morio, Terufumi Morishita, Hiroaki Ozaki, and Toshinori Miyoshi. 2020. Hitachi at SemEval-2020 task 11: An empirical study of pre-trained transformer family for propaganda detection. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1739–1748, Barcelona (online). International Committee for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

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. Association for Computational Linguistics.

Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. 2017. Fake news detection on social media: A data mining perspective. *SIGKDD Explor. Newsl.*, 19(1):22–36.

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*, pages 125–134, Hong Kong, China. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

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*, pages 148–154, Hong Kong, China. 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*, pages 87–91, Hong Kong, China. Association for Computational Linguistics.