Synthetic Propaganda Embeddings to Train a Linear Projection

Adam Ek **Mehdi Ghanimifard**
Centre for Linguistic Theory and Studies in Probability
Department of Philosophy, Linguistics and Theory of Science
University of Gothenburg, Sweden
{adam.ek,mehdi.ghanimifard}@gu.se

Abstract

This paper presents a method of detecting fine-grained categories of propaganda in text. Given a sentence, our method aims to identify a span of words and predict the type of propaganda used. To detect propaganda, we explore a method for extracting features of propaganda from contextualized embeddings without fine-tuning the large parameters of the base model. We show that by generating synthetic embeddings we can train a linear function with ReLU activation to extract useful labeled embeddings from an embedding space generated by a general-purpose language model. We also introduce an inference technique to detect continuous spans in sequences of propaganda tokens in sentences. A result of the ensemble model is submitted to the first shared task in fine-grained propaganda detection at NLP4IF as Team Stalin. In this paper, we provide additional analysis regarding our method of detecting spans of propaganda with synthetically generated representations.

1 Introduction

Automatic propaganda identification is a task which requires a full set of natural language technologies, including language understanding, discourse analysis, common-sense reasoning, fact-checking and many more. By focusing on the genre to political news articles, it is possible to some extent identify content expressing propaganda based on its stylistic features, readability level, and keyword features (Barrón-Cedeno et al., 2019).

We propose a simple method for extracting and curating features of propaganda by utilizing contextualized token representations obtained from pre-trained language models. Contextualized token representations have been used successfully in several natural language understanding tasks, such as question answering, natural language inference and more (Devlin et al., 2019; Peters et al., 2018a; Wang et al., 2018). A contextualized token embeddning represent a token in-context, i.e. the same word in different contexts will have different contextualized embeddnings. The embeddnings in this paper is used for the task of identifying fine-grained propaganda. The task of fine-grained propaganda detection is defined as finding which spans of tokens in a text express some type of propaganda.

The standard procedure for using pre-trained models is to *train* a language model on unlabeled data, then *fine-tune* its learned feature representations as contextual embeddings on specific tasks. Often, the fine-tuning of pre-trained language models require a large annotated dataset to be able to extract invariant and discriminatory features for the task. While fine-grained propaganda detection potentially can benefit from the these model designs, the available annotated data for fine-grained propaganda techniques is relatively small. This pose a problem, as the distribution of propaganda classes is imbalanced, in addition to the dataset being small.

In this paper, we explore a data augmentation procedure aimed at balancing the dataset by generating synthetic contextualized embeddings of propaganda techniques based on expert annotations. This address the problem of fine-tuning the model for our task, as we both balance the class distributions and increase the size of the dataset.

The remainder of the paper is organized as follow: Section 2 gives a brief introduction to the task, Section 3 presents a detailed description of our system, and in Section 4 an evaluation of our system is performed and discussed.

* Authors sorted alphabetically.

Proceedings of the 2nd Workshop on NLP for Internet Freedom: Censorship, Disinformation, and Propaganda, pages 155–161
Hong Kong, China, November 4, 2019 ©2019 Association for Computational Linguistics

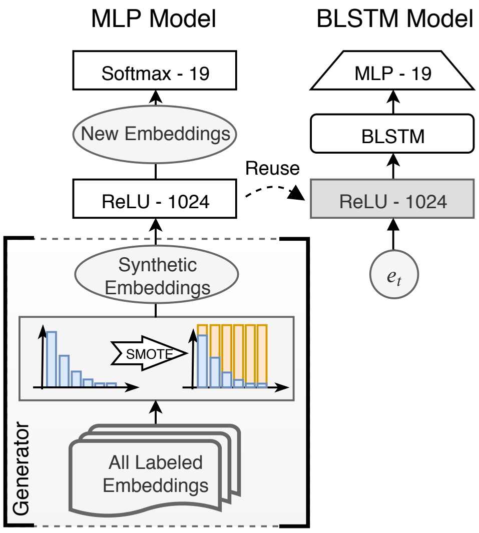

Figure 1: The STALin system architecture

2 Task overview

The task our system is trying to solve is the following: given a text, identify all spans of text (can be multiple tokens) that contain propaganda. Propaganda is categorized into 18 different classes, spanning single tokens in some cases and longer phrases in other. Thus, a successful system must identify both short and long spans of text that include propaganda. While some classes appear simple such as $name\_calling, labeling$ and $exaggeration/minimization$, other classes such as $straw\ man$ require both world and context knowledge to solve. The propaganda classes and the task is further described in Da San Martino et al. (2019).

3 STALin Procedure

STALin is our proposed procedure to generate **S**ynthetic propaganda embeddings to **T**rain **A Lin**ear projection for contextual embeddings.[1] The neural network model we use is designed to be minimal and simple. The architecture is displayed as a schema in Figure 1. The general idea is that we use pre-trained contextual embeddings as feature representation of each token, then sample synthetic embeddnings from the representations. Then a neural classifier is trained for token level fine-grained propaganda prediction in two steps, first we use a MLP layer followed by a bidirectional LSTM layer.

Since the annotated data is small (350 articles) and the number of token instances for each of the 19 classes are not balanced, we propose a simple method to project contextual embeddings into a more balanced embedding space with synthetic samples. To create a balanced embedding space, we use synthetic minority over-sampling (SMOTE) (Chawla et al., 2002) to generated token embeddnings for the minority classes in the dataset. With the balanced training data we train the classifier described previously to predict labels for tokens representations on the propaganda identification task. After training using the balanced embedding space, we use the learned representation in an additional bidirectional LSTM. The contextual embeddings represent each token in its context, in other words, these representations not just encode the knowledge about each token they also encodes features about the current context.

Contextual embeddings In this report, we compare the performance of 3 different models of pre-trained contextual embeddings. We use an implementation with 1024 dimensions:

- ELMo (Peters et al., 2018a) is a weighted sum of multiple layers of BLSTM trained on a large sequence of text corpora as a word predicting language model.[2]
- BERT (Devlin et al., 2019) is a bidirectional transformer encoder trained on large corpora of documents for two tasks of language modeling (1) token predictions (2) next sentence prediction.[3]
- GROVER (Zellers et al., 2019) is a generative language model a transformer-based encoder similar to GPT-2 language model (Radford et al., 2019), which specifically is trained to generate news articles conditioned with metadata about title, authors, source and date of publication. We use the hidden state of the model as embeddings for propaganda identification task.[4]

The tokenization scheme in ELMo is based on white-space as token boundaries. We used the

[1] Our implementation is available at: https://github.com/GU-CLASP/nlp4if19-stalin

[2] We use version 2 implementation trained on 1 billion words at https://tfhub.dev/google/elmo/2

[3] We use *BERT-large cased* at https://storage.googleapis.com/bert_models/2018_10_18/cased_L-24_H-1024_A-16.zip

[4] We use the hidden states in *GROVER-large* trained on *realnews* corpus.

same tokenization for BERT according to the *bert-as-service* implementation. However, GROVER is using subwords vocabularies with *byte pair encoding* (Sennrich et al., 2016).

SMOTE oversampling As it was discussed earlier, contextual embeddings for each token represent token-in-context. Having tokens annotated by their propaganda techniques, we can over-sample on the contextualized embeddnings of the minority classes using SMOTE. The SMOTE algorithm, generates nearest neighbor vectors for all categories of token-context vectors then it balances the number of instances on each category by oversampling from minority class. The generated synthetic samples are not representative of any specific token-in-context but they are in proximal interpolations of the known token-in-context embeddings.

In our model, for a class k we generate new synthetic samples based on the 20 nearest neighbours within that class. We use a one vs all strategy during the sampling. For each class C_k we generate N synthetic samples where $N = |C| - |C_k|$, i.e. we pairwise generate new synthetic samples for the class based on the number of samples in the other propaganda classes. We use off-the-shelf implementation of the SMOTE algorithm in (Lemaître et al., 2017).

MLP Model The MLP model consists of two dense layers trained with categorical cross-entropy loss and Adam optimization:

1. Dense layer of size 1536 projecting embeddings on to a 1024 space with ReLU activation and a dropout rate of 0.5

2. Dense layer with softmax activation to predict one of the 19 possible labels: the 18 classes of propaganda and a non-propaganda label.

After training the plain model, we use the first dense layer as a fixed projection function to transform any new contextualized embeddings into the new embedding space.

BLSTM Model We use the projected of embeddings from the first layer of the MLP model as the input for a one layer of bidirectional LSTM with 1024 units. The BLSTM layer use a dropout rate of 0.5. We then use a copy of the MLP model described above to predict which class a token belong to. For the BLSTM model we also use categorical cross-entropy loss and Adam optimization.

3.1 Training

We use contextualized embeddings as inputs to the model, and do not update the language model parameters. First, the MLP model is trained with a batch size of 1024 for 20 epochs. The input to this model is the synthetically generated token as described previously. Secondly, we freeze updates on the parameters in the first layer of MLP model, and we use it as inputs to the BLSTM model. The BLSTM model is trained for 10 epochs with same batch size. When training the BLSTM for GROVER, we used a batch of 256 due to the GPU memory limitations.

3.2 Inference

Despite using softmax activation to fit the model with one of the 19 classes during training, it is needed to infer concurrent classes. To select the most probable classes for each token, we apply a threshold to the softmax output. We experimented with several different techniques for generating a threshold but found that using the proportion of non-propaganda tokens to propaganda tokens in the training data gave the best results. Thus, all classes whose probabilities for a token is higher than the proportion of propaganda to non-propaganda in the training data is selected as a possible label for the current token.

After assigning possible propaganda labels for each token, we run two post-processing step on the predicted labels. First, we fill the gap between two labeled tokens: for each sub-sequence of three tokens, if the head and tail tokens have any propaganda labels, the intersection of their labels is going to be assigned to the middle token. Second, instead of reporting all token labels, we collapse continuous propaganda tokens into one label, representing one span. The final label for a multi-token span is determined by the label which has the highest estimated likelihood of all the labels assigned the span of tokens.

To summarize, we use one model to both detect relevant spans of text and to label them with the classes.

3.3 Ensemble model

For our final predictions on the test set we created a mapping from models to labels as we noted that some models performed better on certain classes of propaganda than other in our validation data. Thus, our ensemble model is a mapping between labels and models.

We selected the model-label mapping based on the F1-score of the models over our randomly selected sentences in the validation set split[5]. On our validation set, BERT did not perform well, thus it was not used in our final model. In our final submission, we used GROVER for: *Slogans, Doubt, Repetition, Name-calling,Labeling, Loaded_Language, Whataboutism* and *Obfuscation* and ELMo for the remaining classes.

4 Evaluation

4.1 Ablation study

Hypothesis Generating balanced data with SMOTE and using BLSTM to extract features of propaganda from language model embeddings improve the models ability to detect propaganda.

Method We perform an ablation study on ELMo, BERT and GROVER by including/excluding SMOTE and/or the BLSTM model. The results are obtained from the development set and are shown in Table 1.

	SMOTE	BLSTM	F_1-score	Precision	Recall
ELMo	-	-	0.041	0.022	0.278
	+	-	0.010	0.013	0.008
	-	+	0.069	0.039	**0.289**
	+	+	**0.141**	**0.137**	0.146
BERT	-	-	0.048	0.026	0.276
	+	-	0.098	0.089	0.111
	-	+	0.037	0.019	**0.279**
	+	+	**0.116**	**0.155**	0.093
GROVER	-	-	0.148	0.141	**0.156**
	+	-	0.076	0.067	0.089
	-	+	**0.153**	0.157	0.149
	+	+	0.125	**0.233**	0.085

Table 1: Effect of using SMOTE and BLSTM fine-tuning on the pre-trained language model using macro-averaged F1-score.

Results and discussions The results of our ablation study show mixed results for both SMOTE and BLSTM. Using SMOTE appear to lower the recall on all models, while also lowering the precision in ELMo and GROVER. However, for BERT the precision is increased when using SMOTE. This seems to indicate that synthetic sampling works better for BERT than for ELMo and GROVER.

One of the key differences between BERT and ELMo/GROVER is that BERT is trained by using masking, where words in a sentence are removed and then predicted by the model. SMOTE may work better for BERT since it generates a synthetic sample by sampling from contextual embeddings, i.e. words in context, which can be regarded as a specific word in a specific context, which is what the training of BERT capture. Using only the BLSTM and not SMOTE increase the precision in ELMo and GROVER while lowering it for BERT.

Most interesting is that even with these fluctuations the best results are obtained by combining SMOTE and BLSTM. However, this is not the case for GROVER, where only using BLSTM provide the best results. This is perhaps not so surprising when we consider what type of data the models were trained on. Both ELMo and BERT are trained on varied types of text, while Grover is specifically trained on news articles and their metadata. Moreover, GROVER embeddings must have discriminatory meta-features encoded in the data such as author, source and date. The absence of this meta-information in the SMOTE embeddings may be the cause of the lowered performance. Including meta-features could potentially enrich the context for the tokens generated. This implies that if GROVER already has high-level encoded features to identify some classes of propaganda, using SMOTE with only local features simply introduce noise into the embedding space and discriminatory features are lost. One argument in favor of SMOTE in GROVER despite its poor performance is that GROVER achieves its highest precision of all models when SMOTE and BLSTM are combined, and high precision is a useful property for creating ensemble models.

4.2 Fine grained span predictions

Hypothesis The inference method for detecting continuous propaganda sequence can distinguish spans of different propaganda categories.

Method We report results per class for the FLC task on the development data in Table 2. The

[5] We used 1024 (one batch) of randomly selected sentences in the validation set.

	Total	Appeal to Authority	Appeal to fear-prejudice	Bandwagon	Black-and-White Fallacy	Causal Oversimplification	Doubt	Exaggeration, Minimisation	Flag-Waving	Loaded Language	Name Calling, Labeling	Obfuscation, Vague,...	Red Herring	Reductio ad hitlerum	Repetition	Slogans	Straw man	Thought-terminating Cliches	Whataboutism
ELMo	**0.14**	0.00	**0.28**	0.00	0.00	**0.04**	0.10	0.16	0.12	**0.30**	0.18	0.00	0.00	0.06	**0.08**	0.00	0.00	**0.10**	0.00
BERT	0.12	0.00	0.13	0.00	0.00	0.00	0.00	0.21	**0.36**	0.23	0.12	0.00	0.00	0.00	0.03	0.05	0.00	0.00	0.00
GROVER	0.13	**0.06**	0.16	0.00	0.00	0.00	**0.13**	**0.25**	0.27	0.28	**0.20**	0.00	0.00	**0.12**	0.01	**0.11**	0.00	0.00	0.00
Test-performance	0.14	0.00	0.14	0.00	0.00	0.00	0.16	0.10	0.35	0.25	0.25	0.00	0.00	0.35	0.10	0.07	0.00	0.07	0.03
Train	-	160	106	123	107	127	125	45	62	24	27	120	78	97	17	25	80	32	115
ELMo	-	307	22	-	10	25	29	15	31	13	16	-	6	86	10	21	-	9	35
BERT	-	93	22	-	10	23	31	15	21	12	13	-	6	65	10	13	20	25	40
GROVER	-	74	22	-	69	26	40	15	20	11	13	-	5	28	10	11	20	18	44

Table 2: (1) F1-score for classes in the FLC task. (2) Mean character length for each class in the training data, and in the labels predicted by the models on the development set.

F1-score per class is calculated to include partial matching as described in (Da San Martino et al., 2019). Of our three models with SMOTE and BLSTM, ELMo showed the overall best performance. However, for individual classes, the best model varies. We consider span length prediction as a qualitative analysis for the model, as some of the classes span whole phrases while some only span over single tokens.

Discussion Each propaganda span in training data represents a meaningful continuous sequence often as linguistic units such as phrases or sentences. Depending on the propaganda method, the span might be short such as a single adjective as *Loaded Language* span or it might be a long sentence as the span of *Doubt*.

Earlier, we described our post-processing inference to predict continues spans. Observing the results from Table 2, not all models are predicting meaningful span length on each class comparing to the average length in training data (i.e. the mean number of characters for Red Herring is 6 in our models, while in the training data this class appears to span phrases). We calculate the correlation coefficient (r) between the average predicted length of propaganda techniques and the average length in training data. If a model has not predicted propaganda technique k, it was removed from the correlation calculation. Thus, this measurement only deals with predicted spans compared to gold spans and does not penalize the model if it does not predict spans for some classes.

Model	Correlation (r)	p
ELMo	0.567	0.027
BERT	0.638	0.007
GROVER	0.766	0.000

Table 3: Pearson correlation (r) and p-value for the predicted span lengths of the models.

Results The results are shown in Table 3. The result indicate that GROVER is the best model for identifying span lengths for all classes, while ELMo has the worst performance. It is rather surprising as ELMo is the model which performed best on the development data. This indicates that while GROVER is good at identifying spans, ELMo is generally better at labeling them with their correct class.

5 Summary and future works

In this paper, we presented STALin, a transfer learning method with linear tuning of contextualized token embeddings in the fine-grained propaganda detection task. We showed that balancing the data representation with synthetic token embeddings with SMOTE algorithm improved the representations of ELMo and BERT token embeddings. Our ablation study indicates that representations obtained by GROVER are fairly good for detecting propaganda out-of-the-box. GROVER performs better than BERT and ELMo without any fine-tuning, and our fine-tuning method on GROVER improved the precision but resulted in a lower overall recall (See Table 1). One possible reason for the lower performance of the fine-tuned GROVER is that some meta-data is missing, which GROVER relies on to update its representations. This project also raises questions in transfer learning about what features are learned in the fine-tuning phase, and what techniques for fine-tuning are appropriate for what tasks and datasets.

This study has the potential to be improved in several directions:

- Pre-trained models use surface information as input and learn deeper relations between words ”from scratch”. A way of introducing inductive bias into the embeddnings would be to annotate the words with syntax (Peters et al., 2018b). As the task of propaganda detection require a deeper understanding of the text than surface information this is a promising avenue to explore.

- Compare and combine other methods of fine-tuning in the procedure. As some of our results are inconsistent (Table 1) additional evaluation using conventional fine-tuning methods would aid us in understanding what is learned by fine-tuning.

- The fine-grained propaganda classes often overlap in context and concepts. As such, collapsing the fine-grained classes into more coarse-grained classes would yield a smaller and more balanced feature space from which samples can be drawn.

- Additional studies and evaluation using GROVER for high-precision propaganda detection. High precision models can be used as another source of generating training data instead of over-sampling balancing.

- Our model design is quite simple and sentences surrounding the current sentence are not used. This could be improved by expanding the models to include previous sentences as additional context to the current predictions. Also in the case of GROVER, including meta-information such as source and author would benefit the model.

Acknowledgements

The research reported in this paper was supported by a grant from the Swedish Research Council (VR project 2014-39) for the establishment of the Centre for Linguistic Theory and Studies in Probability (CLASP) at the University of Gothenburg.

References

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.

Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357.

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 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, EMNLP-IJCNLP 2019, Hong Kong, China.

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. Association for Computational Linguistics.

Guillaume Lemaître, Fernando Nogueira, and Christos K. Aridas. 2017. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, 18(17):1–5.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018a. 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. Association for Computational Linguistics.

Matthew E Peters, Mark Neumann, Luke Zettlemoyer, and Wen-tau Yih. 2018b. Dissecting contextual word embeddings: Architecture and representation. *arXiv preprint arXiv:1808.08949*.

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

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. *arXiv preprint arXiv:1905.12616*.