NoPropaganda at SemEval-2020 Task 11: A Borrowed Approach to Sequence Tagging and Text Classification

Ilya Dimov
ABBYY ARD NLP
Lomonosov Moscow State University
ilia.dimov@abbyy.com

Vladislav Korzun
ABBYY ARD NLP
MIPT ABBYY Lab
vladislav.korzun@abbyy.com

Ivan Smurov
ABBYY ARD NLP
MIPT ABBYY Lab
ivan.smurov@abbyy.com

Abstract

This paper describes our contribution to SemEval-2020 Task 11: *Detection Of Propaganda Techniques In News Articles.* We start with simple LSTM baselines and move to an autoregressive transformer decoder to predict long continuous propaganda spans for the first subtask. We also adopt an approach from relation extraction by enveloping spans mentioned above with special tokens for the second subtask of propaganda technique classification. Our models report an F-score of 44.6% and a micro-averaged F-score of 58.2% for those tasks accordingly.

1 Introduction

In recent years natural language processing has experienced rapid development. In particular, application of deep learning to NLP (Collobert et al., 2011), the introduction of pre-trained context-independent word embeddings such as word2vec (Mikolov et al., 2013a; Mikolov et al., 2013b), GloVe (Pennington et al., 2014) and fastText (Bojanowski et al., 2017), and utilization of RNN-based (Hochreiter and Schmidhuber, 1997) pipelines such as CharCNN-BLSTM-CRF (Lample et al., 2016; Ma and Hovy, 2016) allowed to improve SOTA for an overwhelming majority of NLP tasks. In 2018 a monumental breakthrough happened when context-dependent embeddings based on pre-trained language models emerged, such as ELMo (Peters et al., 2018), ULMFit (Howard and Ruder, 2018), and BERT (Devlin et al., 2018). The resulting boost in the models' performance cannot be underestimated, amounting to a 30% relative error reduction rate.

Researchers compare this latest development to the introduction of ImageNet to Computer Vision. Furthermore, as breakthroughs in CV allowed such technologies as DeepFake to arise, similar processes are happening to NLP: the authors of GPT2 (Radford et al., 2019) anticipated that their model could be used for malicious purposes such as fake news generation or even impersonation. Given that at present, the problem of media swaying public opinion to the benefit of certain parties is more acute than ever, we must keep our news fair and unbiased; thus, techniques for fake news and propaganda detection need to be developed. SemEval-2020 task 11 is an essential step in that direction.

Another noteworthy consequence of contextualized embeddings introduction is the generalization of models. While many models are developed for a particular task, they can be effectively utilized in order to solve other tasks. Our main contribution is the utilization of two such models for new tasks, namely LaserTagger (Malmi et al., 2019), initially developed for summarization, which is used for span identification, while R-BERT (Wu and He, 2019), originally developed for relation extraction, is used for span classification.

Section 2 contains the description of our systems[1], section 3 contains evaluation results and error analysis, while part 4 contains the concluding remark.

---

[1]The code is released at https://github.com/hawkeoni/Semeval2020_task11.

2 System description

All of the following models based on **BERT** (Devlin et al., 2018) use **BERT$_{BASE}$** pretrained model weights. Hyperparameter tuning and cross-validation was performed on 5 folds split by articles. The data used for training and evaluation consists of various news articles and is described in detail in (Da San Martino et al., 2019). The tasks will be briefly explained in the following sections; their full description can be found in the (Da San Martino et al., 2020).

2.1 Span Identification

The goal of this task is to find continuous spans of text which contain any propaganda technique. Consider the following example, which has *loaded language* marked up in bold and double squared brackets:

> It's essentially an admission of guilt, given that it is **[[absolutely ludicrous]]** to think that "national security" would be threatened by the release of the CIA's long-secret JFK-assassination-related records.

We treat this task as a binary sequence tagging where each token is assigned a label: 0 - for normal text, 1 - for propaganda text. Although there are spans which overlap multiple sentences, we decided to split such spans and solve the problem on a sentence level. This approach has a distinct disadvantage, as the connection between sentences is lost.

At first we started with several baseline bidirectional LSTM models with linear layer for token class prediction:

- **biLSTM$_{GLOVE + charLSTM}$** - a bidirectional LSTM over GloVe embeddings and a character-level LSTM. This is a simple sequence tagging model proposed originally in (Lample et al., 2016).

- **biLSTM$_{ELMO}$** - a bidirectional LSTM over ELMo embeddings from (Peters et al., 2018).

Preliminary results of those models achieved an F-score of 24-26% on the development set. Postprocessing model predictions by labeling all tokens between the first sentence-wise positive label and the last sentence-wise positive label as propaganda pushed F-score to 34.5%. A CRF instead of a linear layer provided no significant benefit.

Our next model BERT$_{LINEAR}$, which consists of a linear layer over **BERT$_{BASE}$**, achieved a stronger baseline with F-score of 40.8% without any postprocessing. However, unlike in previous models postprocessing only made things worse, degrading prediction quality.

Generating token tags with a linear layer has one big flaw: predictions for each token are made independently. CRF mitigates this problem and improves the metrics, but we decided to employ **LaserTagger** - an autoregressive transformer decoder from (Malmi et al., 2019). One of the critical ideas of **LaserTagger** is directly consuming corresponding encoder activation without learning encoder-decoder attention weights. **LaserTagger** achieved an even stronger baseline with an F-score of 42% on development set, which at the time was only several points away from top-performing teams. It was decided to keep this model and extensively tune it during further experiments.

We have employed several techniques to boost model performance:

- We fine-tune **BERT** as a masked language model on task corpus for 3 epochs without next sentence prediction loss. This provides no noticeable increase in F-score.

- Teacher forcing (scheduled sampling) (Bengio et al., 2015) is a method of improving performance for autoregressive models. As **LaserTagger** predictions explicitly depend on predictions from the previous step, errors accumulated earlier may lead to a completely wrong result. Teacher forcing aims to close the gap between training and inference by supplying a model with its own (possibly wrong) predictions at a specific rate during training. This allows the model to explore more of its state space during training, which in turn increases its robustness during inference. We linearly decay teacher forcing rate from 1 (only correct labels) to 0 (only model predictions) during training. The downside of this approach is a significant slowdown in training as we do not utilize the transformer's capability to process the whole sequence at once.
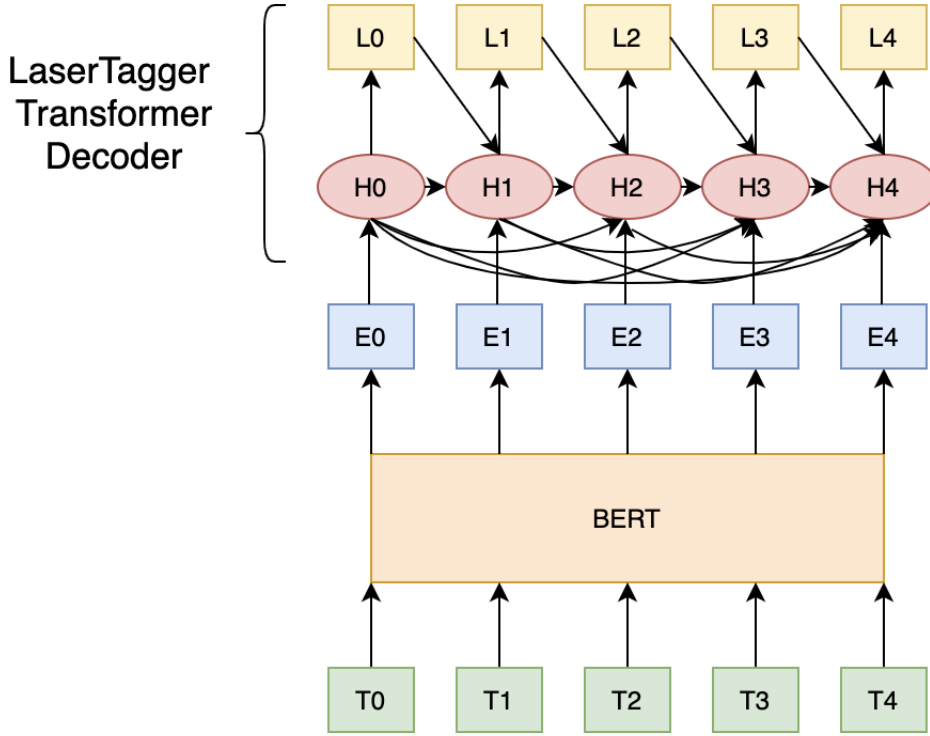
Figure 1: LaserTagger model architecture. T_i are token embeddings for **BERT**, E_i are **BERT** outputs, H_i represent LaserTagger hidden states and L_i are predicted labels.

- Label smoothing was used to lower the number of false-positive predictions and mitigating minor data discrepancies like inconsistent inclusion of dots and quotations in propaganda spans, which were caused by limiting spans to sentences and other preprocessing imperfections.

Our final submission was a single **LaserTagger** model with **BERT$_{\text{BASE}}$** encoder, single-layered decoder with hidden dimension of 128 and 4 attention heads. The model was trained on an effective batch size of 32 achieved by gradient accumulation over every 2 steps; we used Adam optimizer with learning rate of 2e-5 and learning rate warmup for 10% of the steps. Teacher forcing was set to linearly decay from 1 to 0 for the duration of the whole training. This model achieved F-score of 46.1% on development set and 44.6% on test set.

2.2 Technique Classification

The second subtask was to classify spans from previous subtasks into 14 classes of propaganda techniques. As previously mentioned, spans can overlap several sentences, so for this subtask, we had to use all overlapping sentences for correct classification. As spans with different techniques may overlap, or a single span may be associated with several techniques, we treat this subtask as a multilabel classification problem.

Our main idea was taken from **R-BERT** (Wu and He, 2019), where special tokens were inserted around entities to highlight them in the task of relation extraction. As in **R-BERT**, we surround propaganda spans with special tokens, namely the ˆ token. **BERT** representations of propaganda span tokens are then averaged or taken as a weighted sum based on results of the linear layer applied to **BERT** token activations in the following way: let $e_i...e_{i+k} \in R^n$ be **BERT** representations of propaganda tokens in the current span and $w \in R^n, b \in R$ - learnable parameters. Then the weighted sum is calculated as $\sum_{j=i}^{i+k} \alpha_j e_j$, where α_j is the result of softmax function over scalar product of e_j and w:

$$\alpha_j = \frac{\exp\left((w, e_j) + b\right)}{\sum_{j=i}^{i+k} \exp\left((w, e_j) + b\right)} \tag{1}$$

The resulting vector is concatenated with **BERT** output of the **[CLS]** token for final multilabel prediction. We also combine these approaches with previously mentioned **BERT** intask finetuning (Sun et al., 2019), which proves to be fruitful for this task as it is not too dissimilar to text classification. We additionally train a simple **BERT** model, which predicts technique class using only **[CLS]** token representation. All models were trained with linear warmup for 10% of the training steps using the Adam optimizer with learning rate of 2e-5.

Our final submission for task TC consists of predictions averaged from several models, which are described in Section 3.2.
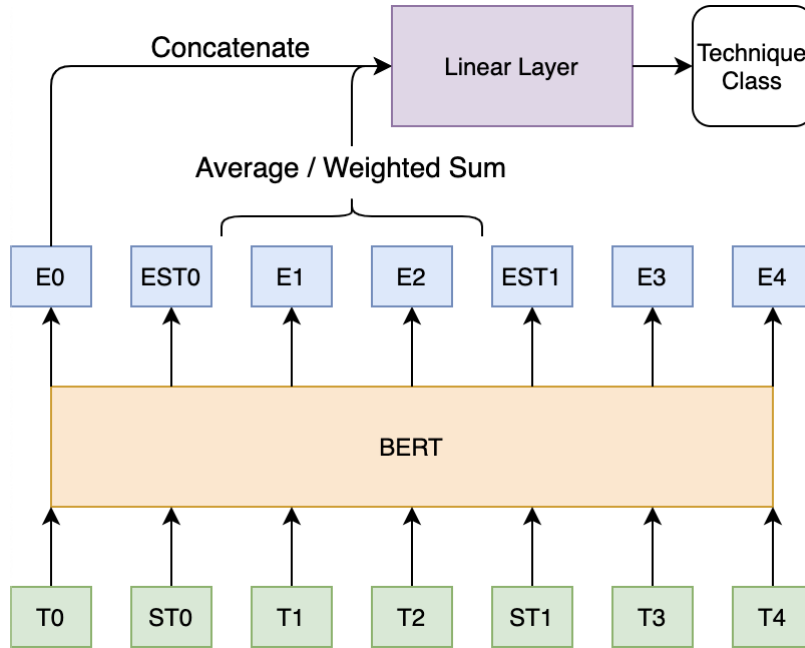

Figure 2: Task TC model architecture based on R-BERT. T_i are token embeddings for **BERT**; ST_0 and ST_1 represent special tokens used to highlight the propaganda span; E_i and EST_i are **BERT** outputs .

3 Experimental Results

3.1 Span Identification Results

Model	F1	Precision	Recall
biLSTM$_{\text{GLOVE + charLSTM}}$	0.249	0.302	0.212
biLSTM$_{\text{GLOVE + charLSTM}}$ + postprocessing	0.349	0.304	0.411
biLSTM$_{\text{ELMO}}$ + postprocessing	0.345	0.327	0.367
BERT$_{\text{LINEAR}}$	0.408	0.357	0.477
BERT + CRF	0.362	0.301	0.454
BERT + LSTM + CRF	0.414	0.363	0.482
LaserTagger	0.420	0.384	0.464
LaserTagger + TF	0.451	**0.423**	0.482
LaserTagger + TF + LS	**0.461**	0.406	**0.533**
LaserTagger + TF + LS (test)	0.446	0.556	0.373

Table 1: Task SI model results on development and test sets. TF and LS stand for teacher forcing and label smoothing accordingly.

As Table 1 shows **BERT**-based approaches significantly outperform suggested LSTM baselines. Main

Boundary errors
Gold: This is a **[[smear]]**, plain and simple.
Prediction: This is **[[a smear]]**, plain and simple.
Gold: Luke Harding and the Guardian Publish Still More **[[Blatant MI6 Lies]]**
Prediction: Luke Harding and the Guardian Publish Still More **[[Blatant]]** MI6 Lies
Quotation and punctuation errors
Gold: While trying to appear as a moderate, "Gillum is a Progressive" and "He is a part of the **[[crazy, crazy, crazies]]**."
Prediction: While trying to appear as a moderate, "Gillum is a Progressive" and "**[[He is a part of the crazy, crazy, crazies]]**."
Gold: "**[[I don't care who you are]]**, determine the scope and the parameters of an investigation.
Prediction: "**[[I don't care who you are, determine the scope and the parameters of an investigation]]**.
Gold: "**[[This is devastating]]** for our Democratic Latino community.
Prediction: "**[[This is devastating for our Democratic Latino community.]]**

Table 2: LaserTagger + TF + LS errors.

improvements over simple **BERT$_{\text{LINEAR}}$** model is **LaserTagger** decoder, which eliminates the flaw of independent tag prediction. It was further improved by linearly decreasing teacher forcing rate over the training and label smoothing. Our model scored 7th place out of 35 in the final table of the competition.

Table 2 illustrates the errors of our final model. Besides sometimes incorrectly adding or losing boundary propaganda span tokens, **LaserTagger + TF + LS** seems to have a problem with quotation and punctuation. When it encounters quoted text or a part of a sentence separated by commas, it either labels the whole highlighted area as propaganda or misses it completely. This may be partly attributed to the nature of quotes - speech often conveys more emotional words than written text. Another problem is that when the model finds comma, it often closes the propaganda span and does not reopen it.

3.2 Technique Classification Results

Model	micro-averaged F-measure
BERT$_{\text{CLS}}$	0.573
R-BERT	0.592
R-BERT$_{\text{FT}}$	0.584
R-BERT$_{\text{W}}$	0.590
R-BERT$_{\text{W + FT}}$	0.590
All models	**0.606**
All models (test)	0.582

Table 3: Task TI model results on development and test sets. W and FT stand for weighted sum instead of averaging **BERT** token representation and intask finetuning accordingly.

All models from the table use the ˆ special symbol to highlight propaganda spans because more than 20% of the sentences contain multiple propaganda spans. For example, if the model has 5 tokens $e_1...e_5$ and two propaganda spans e_1, e_2 and e_4, e_5 we would create two training samples with the use token ˆ, which we will write for readability as **st**: $st, e_1, e_2, st, e_3, e_4, e_5$ and $e_1, e_2, e_3, st, e_4, e_5, st$. This allows us to point our model to a specific propaganda span inside a sentence.

Models from Table 3 are the following:

- **BERT$_{\text{CLS}}$** - a model that predicts technique solely from **[CLS]** token embedding.

- **R-BERT** - a model that predicts technique from **[CLS]** token embedding and averaged propaganda span representation;

- **R-BERT$_{\text{W}}$** - a model that predicts technique from **[CLS]** token embedding and weighted propaganda span representation;

It is important to understand that due to class imbalance models with the same micro-averaged F-score correspond to different F-scores over specific techniques. While ensembling the models increases overall quality, two classes, namely **"Bandwagon, Reduction Ad Hitlerum"** and **"Whataboutism, Straw Men, Red Herring"**, are entirely missed by all our models. This is not surprising: those techniques have a very limited amount of positive samples, which is the reason they were merged into classes. Class wise F-score seems to respond to the quantity of corresponding class in training dataset: our model performs best on **Loaded Language** and **Name Calling, Labeling** followed by **Flag-Waving** and **Doubt**.

We did not implement anything to boost performance on underrepresented classes, while some of the other teams found a way to get an F-score of 20% and higher. Our model scored 6th place out of 31 in the final table of the competition.

3.3 Dataset Exploration

During the competition, most teams had a substantial shift between recall and precision: precision was 10-20% lower than recall. On the contrary, test dataset evaluation resulted in a different picture: top-performing teams' precision is 5-15% higher than recall. Additionally, on task TC all teams suffered a downgrade in micro-averaged F-score of approximately 5%.

This can be attributed to the fact that the datasets have imbalanced topic distributions. While the training dataset is not very large, it is well-rounded and covers a wide range of topics. The development set is focused on the US news, while the test set seems to have almost an even split between the UK and the US news.

4 Conclusion

In this work, we propose to use two models: autoregressive transformer decoder for sequence tagging and **R-BERT**-like approach for propaganda spans technique classification. The former is taken from such tasks as sentence fusion, sentence splitting, and abstractive summarization, while the latter was originally used for relation extraction.

Instead of exploring various pretrained transformers like **BERT$_{\text{LARGE}}$**, **RoBERTa**, **ALBERT** and **XLNet** or engineering domain-specific features and approaches, we focused on improving architectures and trying to push them as far as possible.

In the future the results of this task may be used for propaganda detection in various media as well as unbiasing texts and making them fair.

References

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, pages 1171–1179.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(Aug):2493–2537.

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. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

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

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf.

Eric Malmi, Sebastian Krause, Sascha Rothe, Daniil Mirylenka, and Aliaksei Severyn. 2019. Encode, tag, realize: High-precision text editing. *arXiv preprint arXiv:1909.01187*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

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

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune bert for text classification? In *China National Conference on Chinese Computational Linguistics*, pages 194–206. Springer.

Shanchan Wu and Yifan He. 2019. Enriching pre-trained language model with entity information for relation classification. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 2361–2364.