SemEval-2020 Task 11: Experiments on a Novel Approach for the 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 XLNet. The described architecture achieved an F_1-score of 0.46072 on the SI task (ranking 8/45) and proved flawed for the TC task, with important adjustments having to be made before being able to achieve an F_1-score of 0.57572.

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.

Our entire project is publicly available on our GitHub repository[2]. You can also see the results provided by our architecture on the leaderboard of the SemEval-2020 shared task, our team name being *nlpboomers*.

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-

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

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

task. We decided to use the same PLMs as the other teams, with the addition of XLNet. 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.

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. Note that during training, we also update the weight parameters of the pre-trained models in order to fine-tune them.

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

XLNet (Yang et al., 2020) is a state-of-the-art extended Transformer using an autoregressive method to learn bidirectional contexts by maximizing the expected likelihood over all permutations of the input sequence factorization order. In our work, the $\text{XLNet}_{\text{LARGE}}$ 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. The framework is built on top of PyTorch (Paszke et al., 2019) and SpaCy (Honnibal and Montani, 2017), and was explicitly designed for developing state-of-the-art deep learning models on a wide variety of NLP tasks.

3 Dataset

3.1 Data description

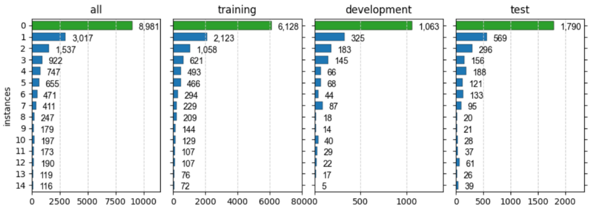
The dataset used for the task, PTC-SemEval20 corpus (Martino et al., 2020a), consists of a sample of news articles collected from mid-2017 to early 2019. The articles were retrieved from 13 propaganda and 36 non-propaganda news outlets, as labeled by Media Bias/Fact Check[3], and manually annotated by 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.

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).

One such piece of information provided by the organizers is the distribution of the different propaganda techniques in the datasets. Those results can be seen in Figure 1, as reported in (Martino et al., 2020a).



0	Overall	5	Doubt	10	Appeal to authority
1	Loaded language	6	Appeal to fear/prejudice	11	Black-and-white fallacy, dictatorship
2	Name calling or labeling	7	Flag-waving	12	Thought-terminating cliché
3	Repetition	8	Causal oversimplification	13	Whataboutism, straw man, red herring
4	Exaggeration or minimization	9	Slogans	14	Bandwagon, reductio ad hitlerum

Figure 1: Number of instances for each technique.

In addition to this data, a more fine-grained exploration of the training corpus was performed as one of the first steps in tackling the task. 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 2 and 3. Due to space constraints, other results (e.g. the distribution over token categories in gold spans and border tokens), were omitted but can be accessed in our GitHub repository.

[3] https://mediabiasfactcheck.com

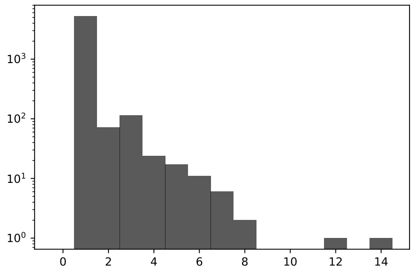


Figure 2: Number of sentences in training gold spans.

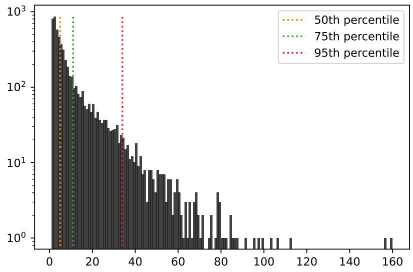


Figure 3: Number of tokens in training gold spans.

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, that can be seen in Figure 4, 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. Intuition was that this would let us regulate the number of false positives we forward onto 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 extensively prune the set of considered spans, and 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 2, 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.

A detailed overview of the SI model can be seen in Appendix A.

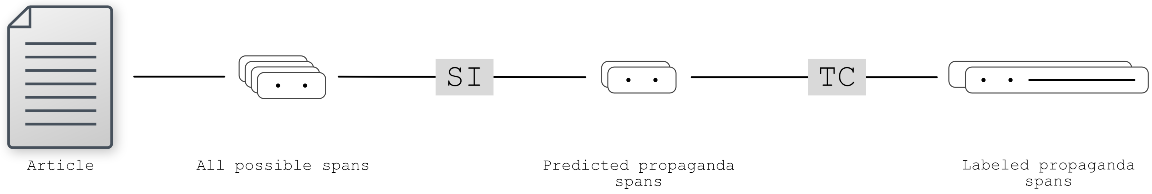


Figure 4: Overview of the architecture.

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 10 tokens or less. According to Figure 3, we can still cover 75% of the gold spans in our training dataset, while only dealing with a computationally manageable amount of spans when enumerating all possibilities (see discussion Section 7.1 for more information). 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 before being fed to the classifier. This embedding, also illustrated in Appendix A, has three components:

1. A weighted average of the word embeddings in the span. The weights used are from a general self-attention vector, masked and renormalized to only contain the span's words. Expectations were that this would encode a general representation of our span.

2. The contextualised representation of both of the span's endpoints, concatenated. These vectors are obtained by using a PLM to embed the text, and then using a Seq2Seq encoder to contextualise those embeddings, in our implementation a LSTM. Our intuition was that the first and last tokens in a span would be particularly important to detect propaganda, e.g. if the span begins and ends with a quotation mark, especially if those are contextualised in respect to the entire text.

3. Finally, the span width is also encoded.

Our results using different PLMs to embed words in our text can be seen in Section 5.1.

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 $S' \backslash S$). An overview of our TC implementation can be seen at Appendix B.

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 pre-trained 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.

Finally, before training the model, we also had to label the set of span S provided by our SI model. For each of the spans $s \in S$ we assigned its original label if the span s had a perfect match with a span in the original TC training dataset, or our new *"Not Propaganda"* label otherwise.

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).

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 introduced weights in the loss function. Those weights were assigned in inverse proportion to the distribution of original classes in the dataset (shown in Figure 1) and the ratio $r = 0.05$ of spans provided by SI model that exactly match a propaganda argument (pointed out in Table 6). The 15th class proportion is $1 - r = 0.95$ and the 14 original classes proportions are $\frac{w_c}{s}$ where $s = \frac{1}{r}\sum_c w_c$ and w_c is their original distribution. We finally reversed the proportions p_i by assigning to each of the classes $1 - p_i$ in order to have the final weights.

5 Experiments

5.1 SI results

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 rest of the experimental setup can be seen in Table 1.

Hyperparameter	Value
Epochs	10
Batch size	1
Max span width	10
Max sequence length	128
LSTM dimension	200
Learning rate (LR)	1e-3
Transformer LR	1e-5

Table 1: Experimental setup for SI.

As discussed in Section 7.1, our setup, and therefore our results, were heavily influenced by various limitations. Even though, SI achieved good results. RoBERTa obtains the highest F_1 score on the test set, as reported in Table 3, letting us rank 8th out of 45 teams.

Model	Custom F_1	Precision	Recall
BERT	0.40008	0.29371	0.62722
RoBERTa	0.42649	0.32754	0.61107
XLNet	0.37930	0.26213	0.68590

Table 2: Model results on SI task on validation data.

Model	Custom F_1	Precision	Recall
BERT	0.29651	0.17528	0.96147
RoBERTa	0.46072	0.40635	0.53189
XLNet	0.43133	0.50394	0.37701

Table 3: Model results on SI task on test data.

5.2 TC results

Even if our novel approached seemed reasonable on paper, the results of the experiments conducted with it clearly pointed to the opposite direction. The assumption that SI returns a set of spans containing all the gold spans was not respected, as discussed in Section 6.1. The problem being that a *perfect match* with a gold span is rarer than expected (see Table 6). Therefore, during training, when generating the labels for the spans provided by the SI model, we didn't often have a perfect match and thus all our training samples were labelled as *"Not propaganda"* (because of our strategy to generate gold labels, as discussed in Section 4.2). This was preventing our algorithm from learning to distinguish between our classes. Indeed, the model output the "*Not Propaganda*" class for every span.

Partially overlapping TC

In order to counteract this limitation, we tried to further relax our model and enrich our gold labels set with *partially overlapping* spans (as described in Table 5). For instance, if SI predicts a span s spanning from token 12 to token 26 and there is a gold span going from token 13 to token 27 with label l, we assigned this propaganda label l to the span s provided by SI. We thus enriched our dataset with partially overlapping spans according to some threshold. However, this method was not successful either in practice, still leading to a model that predicted "*Not Propaganda*" for every span. Indeed, as shown in Table 6, even going as far as allowing for 50% of the span to be a false positive, we would only get 20% of labels being something other than "*Not Propaganda*" in our training set, which was not enough to overcome the issue.

5.2.1 Alternative TC results

To demonstrate the individual abilities of our TC model, we finally decided to implement an alternative version of TC, not taking as input the set of spans provided by our SI model, but taking the spans directly from the dataset as initially proposed by the organizers i.e. a perfectly pruned set of spans. We also removed our 15th class "*Not Propaganda*". This alternative TC demonstrated that our two modules can nevertheless work independently and are capable of providing decent results on both sub-tasks.

Hyperparameter	Value
Epochs	1
Batch size	1
Max span width	10
Max sequence length	128
LSTM dimension	200
Learning rate (LR)	1e-3
Transformer LR	1e-5

Table 4: Experimental setup for TC.

The metric used to evaluate the TC model is a standard micro-averaged F_1-measure, and the exact experimental setup is the same as for our other TC model, and can be seen in Table 4.

Training with RoBERTa, we achieved a validation F_1 metric of 0.57572. Notice that we were not able to generate results for the test data. Because the model took as input the gold spans to be predicted by the SI model, the organizers of the task decided to not share them publicly. After contacting them, we were not able to get access to the correct file in time. Although this is the case, because no hyperparameters were tuned, our validation data serves the same purpose as a test dataset, and should be representative of our model's performance on inputs not seen during training.

6 Error analysis

In order to draw meaningful conclusions about proposed architectures and their performances on both sub-tasks, a specific in-depth error analysis was conducted.

This analysis was performed on classification results of the top performing model for each task, namely RoBERTa-si and RoBERTa-tc. The data used as benchmark was validation data — since no hyperparameter tuning was performed on them (except for the choice of PLM) and therefore they could be used to obtain unbiased information.

6.1 Span Identification Task

As a first approach to the error analysis for SI task, we decided to further investigate the results by breaking them down by propaganda technique. Although, in this task, the model does not explicitly deal with technique classification, all propaganda spans still belong to a specific category, and analysis of how it influenced the prediction results was considered potentially insightful.

Moreover, since the custom F_1 metric used in SI allows non-zero scores for partial matches, the proportion between partially classified, totally identified and entirely missed propaganda spans in the validation articles were included in the analysis. The results of this investigation are reported in Table 5.

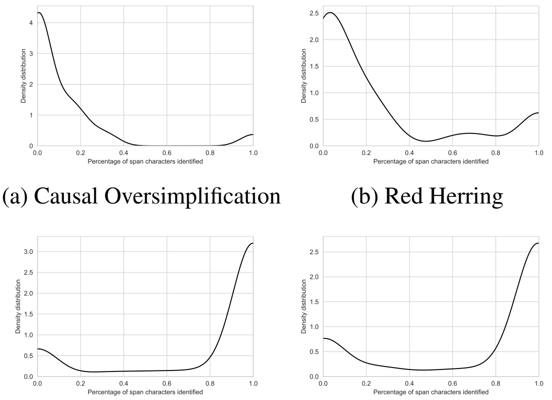
As the data highlights, our system was unable to identify almost one third of propaganda spans in the given articles. On the other hand, roughly 60% of the spans were *totally identified* (i.e. with more than 75% of the characters being correctly classified by SI as propaganda). However, the high disproportion between partial matches and

	Loaded Language	Name Calling	Repetition	Flag Waving	Exaggeration	Doubt	Prejudice	Slogans	Red Herring	Appeal to Authority	Reductio ad hitlerum	Oversimplification	Cliches	Authority	Total
Not identified	51	35	56	18	28	42	9	10	17	7	4	13	9	7	306
Partially identified	23	18	3	19	12	18	17	4	8	1	1	4	3	1	132
Totally identified	251	130	86	50	28	6	18	26	4	6	0	1	5	6	617
Total	325	183	145	87	68	66	44	40	29	14	5	18	17	14	1055

Table 5: `SI` results broken down by propaganda technique. In this setting, a gold span was considered *totally identified* if at least 75% of its characters were labeled as propaganda, *partially identified* if a percentage between 15% and 75% of its characters were labeled as propaganda, *not identified* if less than 15% of its characters were labeled as propaganda.

complete matches, together with the higher *recall* value than *precision* registered in both validation and test results, might suggest that our system tends to predict larger spans than necessary.

An important remark that has to be made is about the changes between the proportion of identified and not identified spans in more (e.g. *Loaded Language* and *Name Calling*) and less (e.g. *Red Herring* and *Reductio ad hitlerium*) frequent techniques. This trend could suggest a direct relation between the number of instances for each propaganda technique and the accuracy achieved by the system in correctly classifying a span that belongs to that technique. This relation can be observed in Figure 5, which reports the distribution of identification proportion for different propaganda techniques.



(a) Causal Oversimplification

(b) Red Herring

(c) Loaded Language

(d) Name Calling

Figure 5: Distribution of identification percentage of gold spans which belong to four different propaganda technique. It can be observed how less frequent techniques in the training set (Figures 5a and 5b) are much harder to label compared to more frequent techniques (Figures 5c and 5d).

However, data in Table 5 is not enough to completely characterize the quality of the predictions of the proposed `SI` model. Because of this, a more in-depth analysis on the similarity between predicted propaganda spans and gold spans was conducted. To evaluate the similarity between predicted spans and gold spans the metric Intersection over Union (`IoU`) was used. `IoU`, also known as *Jaccard index* or *Jaccard similarity coefficient*, is a statistic used for gauging the similarity and diversity of sample sets. It is defined as the size of the intersection divided by the size of the union of the sample sets (i.e. text spans in our setting):

$$IoU(S_1, S_2) = \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|}$$

The analysis was performed by aggregating all predicted spans which matched different thresholds of score, to gain better insights on the distribution of likelihood of our predictions with respect to gold labels. The results of this study are reported in Table 6. As we can see from the data, just a very small percentage of the predicted labels *perfectly* match a gold label, and in this behaviour lies one of the biggest weakness of our proposed architecture. Here we see that one of the fundamental assumption of the proposed approach, that the gold spans are a subset of the predicted spans from `SI`, does not hold in practice.

Threshold	1	≥ 0.5	≥ 0.25	> 0
Percentage	0.041	0.205	0.301	0.397

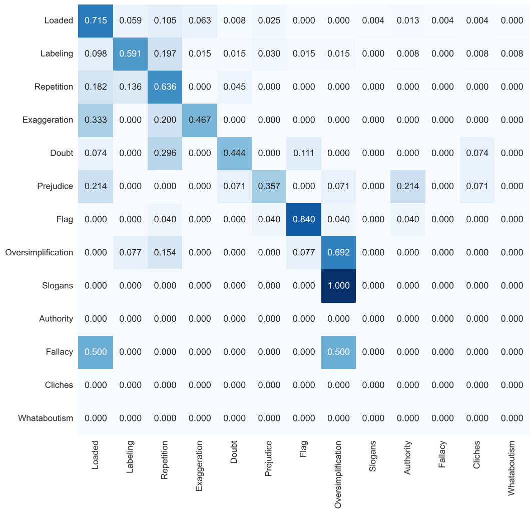
Table 6: Percentages of predicted spans which match different values of `IoU` score.

6.2 Technique Classification Task

As already mentioned in Section 5.2, the `TC` model implemented following the novel approach proposed in this project was not able to produce meaningful results for the Technique Classification subtask. Indeed, performing an error analysis on the results produced by this model would not be interesting.

However, it was still possible to investigate the results obtained with the alternative `TC` classifier. Figure 6 reports the normalized confusion matrix obtained from the analysis of the model prediction on validation data. Interestingly, the most confused classes are *Exaggeration* (primarily with *Loaded Language*), *Doubt* (in this case, mainly with *Repetition*) and *Prejudice* (primarily with *Loaded Language* and *Authority*).

It is also worth observing that, unfortunately, many less-frequent propaganda techniques does not appear in the validation set, and is therefore impossible to have an unbiased evaluation of the classification performances for those techniques.



	Loaded	Labeling	Repetition	Exaggeration	Doubt	Prejudice	Flag	Oversimplification	Slogans	Authority	Fallacy	Cliches	Whataboutism
Loaded	0.715	0.059	0.105	0.063	0.008	0.025	0.000	0.000	0.004	0.013	0.004	0.004	0.000
Labeling	0.098	0.591	0.197	0.015	0.015	0.030	0.015	0.015	0.000	0.008	0.000	0.008	0.008
Repetition	0.182	0.136	0.636	0.000	0.045	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Exaggeration	0.333	0.000	0.200	0.467	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Doubt	0.074	0.000	0.296	0.000	0.444	0.000	0.111	0.000	0.000	0.000	0.000	0.074	0.000
Prejudice	0.214	0.000	0.000	0.000	0.071	0.357	0.000	0.071	0.000	0.214	0.000	0.071	0.000
Flag	0.000	0.000	0.040	0.000	0.000	0.040	0.840	0.040	0.000	0.040	0.000	0.000	0.000
Oversimplification	0.000	0.077	0.154	0.000	0.000	0.000	0.077	0.692	0.000	0.000	0.000	0.000	0.000
Slogans	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000	0.000	0.000	0.000	0.000	0.000
Authority	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Fallacy	0.500	0.000	0.000	0.000	0.000	0.000	0.000	0.500	0.000	0.000	0.000	0.000	0.000
Cliches	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Whataboutism	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Figure 6: Normalized confusion matrix obtained from results of alternative `TC`. Rows represent the correct labels and columns the predicted ones.

7 Discussion and summary

7.1 Discussion

In this work, a novel approach to tackle the detection and classification of propaganda spans in news article was investigated. The core idea behind it was to develop a tool that would have been able to better adapt to real-world scenarios. Nonetheless, during the development of this tool many flaws were found in the initial approach, that proved to not be as effective as initially expected.

The first problem faced during the development of `SI` regarded the memory complexity of the approach. As already mentioned, the decision to approach the task as a span classification problem lead to the evaluation of a potentially exponential number (in the size of the article) of sequences. This, other than the obvious problem with class imbalance between propaganda and non-propaganda spans, also resulted in a major memory issue with batch embedding computation. The memory issue prevented us from effectively training our models on GPUs — due to the limited memory available. This forced us to train our models on CPUs, which resulted in slower computations and therefore the impossibility to perform hyperparameter tuning and validation of our models with techniques like cross-validation and statistical significance indices.

The second problem was the predicting efficiency of our `SI` model. In the proposed approach the efficiency of `TC` was relying on a good span extraction from `SI`. Since the experiments proved our necessary assumption to be wrong, as discussed in Section 6.1, the `TC` model could not provide satisfactory results.

However our `SI` model still predicts propaganda spans relatively well, even if it does not give exact matches. Proof of this is the test result achieved with `RoBERTa` and our rank in the leaderboard $(8/45)$. Furthermore, our alternative for the `TC` task achieves an F_1 metric of 0.57572, and supports the idea that our original model for `TC` is inherently flawed and not that its failure is due to some implementation error.

7.2 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:

- Fine-tuning our investigated models using different PLMs, changing hyper-parameters and adding regularization methods.

- The exploration of add-on features for the architecture such as conditional random field.

- The error analysis revealed that the propaganda techniques commonly confused in the TC task are the same as the techniques that our model was unable to detect in the SI task. A possible route of improvement for the latter might be deploying data augmentation techniques (such as back translation, random replacement and random insertion) to enrich the number of samples that belongs to less frequent techniques, in order to facilitate their identification.

- Improving our top layer classification algorithm. For the models studied, we explored only linear classification and it might be possible to achieve better results with deeper networks.

- Exploring more PLMs and eventually using ensemble techniques to get even more meaningful embeddings.

7.3 Outro

This very paper, as well as the ETH Zürich Natural Language Processing course's lectures note, were checked using the proposed system, to detect fragments one may suspect to represent one or more propaganda techniques. The results are included in Appendix C.

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.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2020. Xlnet: Generalized autoregressive pretraining for language understanding.

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.

Appendix A `SI` model detailed architecture

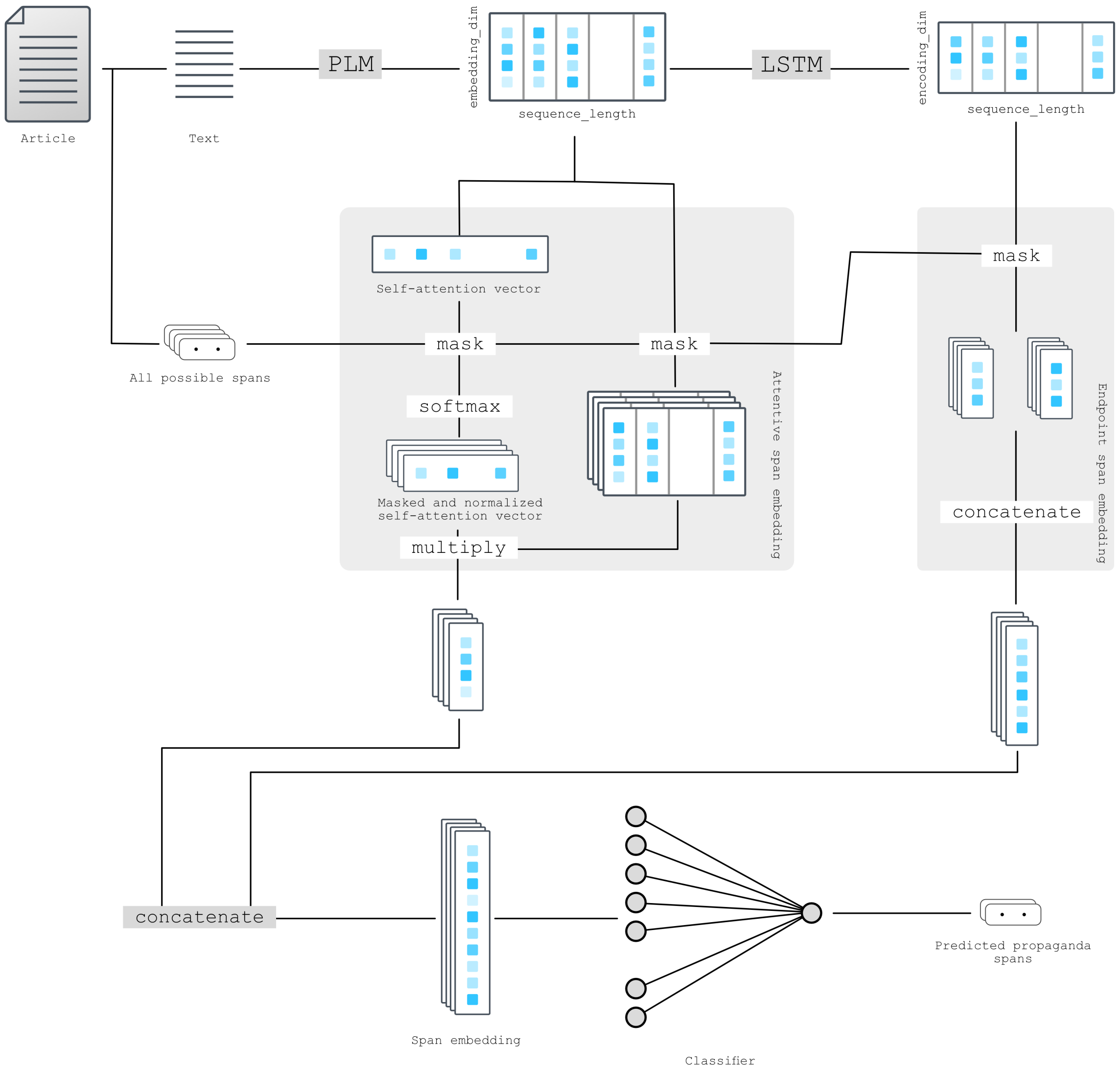


Figure 7: Some illustrations from the `AllenNLP` guide.

Appendix B TC model detailed architecture

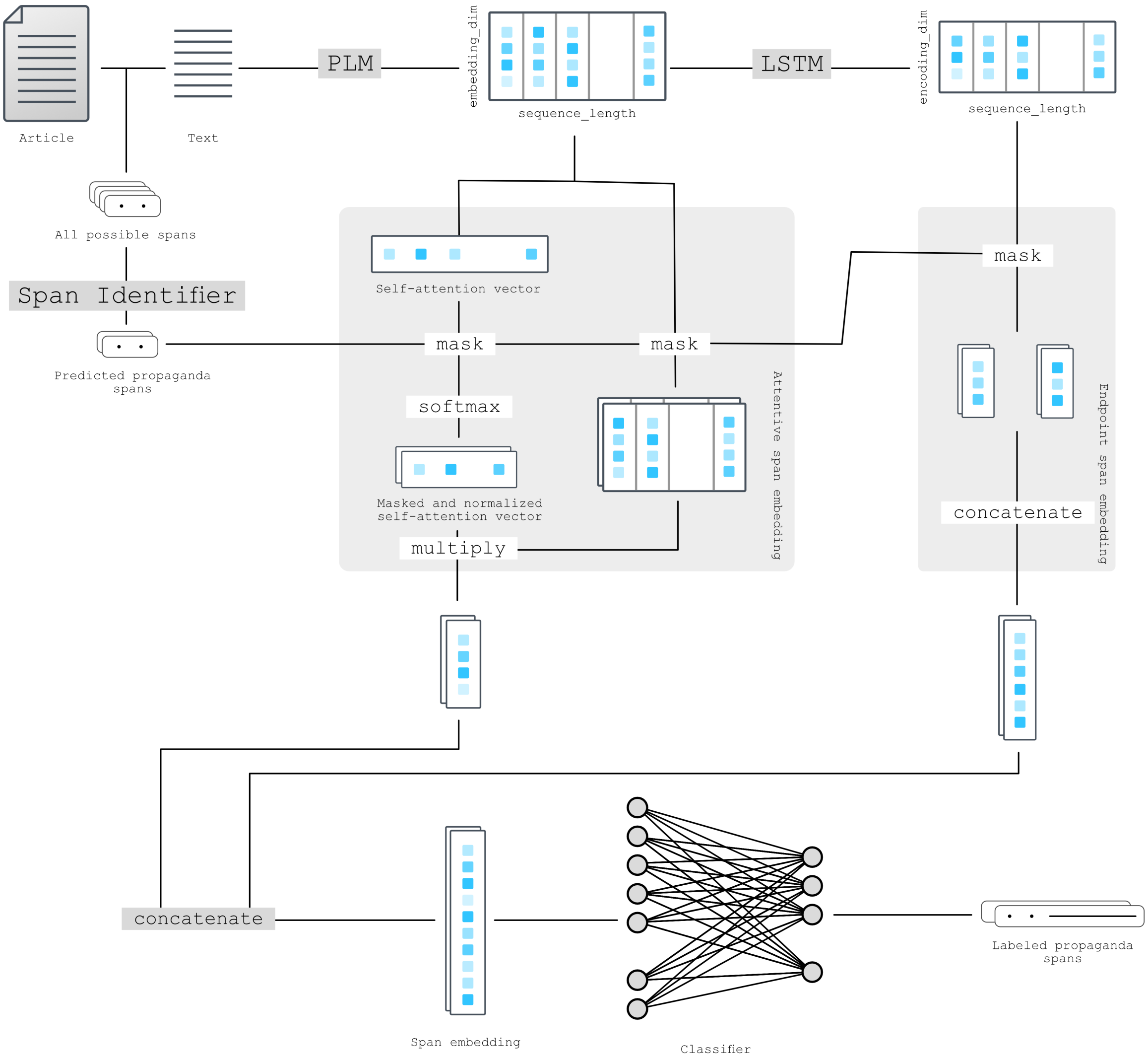


Figure 8: Some illustrations from the AllenNLP guide.

Appendix C Predictions on lecture notes and this paper

C.1 Propaganda detection on the NLP Course notes[4]

```
highly inefficient Loaded_Language
comes in handy . Loaded_Language
redundancies . Loaded_Language
cumbersome Loaded_Language
the sleeker zn . Name_Calling,Labeling
inductive hypothesis ) Loaded_Language
exponential Loaded_Language
redundant . Loaded_Language
the magic of backpropogation . Loaded_Language
the cheap gradient principle : Name_Calling,Labeling
cheap Loaded_Language
log - linear models Repetition
analytically tractable . Loaded_Language
log - linear ,Name_Calling,Labeling
completely BROKEN ! " Loaded_Language
broken ,Loaded_Language
spam classification example Name_Calling,Labeling
gradient descent . Loaded_Language
log - softmax ,Name_Calling,Labeling
exponential families . Repetition
our estimate ,Loaded_Language
very convenient Loaded_Language
The crux of the argument Loaded_Language
non - linear boundary ? Name_Calling,Labeling
log - linear models Name_Calling,Labeling
non - linear decision boundaries , by Repetition
log - linear models : Name_Calling,Labeling
vanishing gradients . Loaded_Language
one - hot encoding scheme . Name_Calling,Labeling
skip - gram model Repetition
naive computation Loaded_Language
structured prediction . Repetition
structured prediction task ,Repetition
bigram language model Loaded_Language
bigram assumption Loaded_Language
smoothing . Loaded_Language
neural n - gram models Name_Calling,Labeling
log - linear models Name_Calling,Labeling
illustrating example Loaded_Language
inherently limited Loaded_Language
explode Loaded_Language
I duck ‟ Repetition
A duck ‟ ,Name_Calling,Labeling
shortest - path - search problem Loaded_Language
computationally intractable . Loaded_Language
our problem . Flag-Waving
additively decomposable Loaded_Language
additively decomposable ,Loaded_Language
an annihilator Name_Calling,Labeling
semirings Loaded_Language
overwhelming evidence Loaded_Language
structured hierarchically . Loaded_Language
hierarchical nature Loaded_Language
caviar Loaded_Language
a cleft sentence ,Loaded_Language
How does papa eat caviar ? Doubt
With a spoon ] Loaded_Language
red car ” Repetition
red car ” Repetition
red car ” Name_Calling,Labeling
single - root constraint . Name_Calling,Labeling
projective and non - projective . Repetition
no edges will cross each other . Loaded_Language
An Intractable Problem Loaded_Language
```

[4] https://drive.google.com/file/d/1esgbEGgF2TYwr0wCE5Sb18YLJ3A9eDgp/view

```
inefficient ,Loaded_Language
greedy decisions Loaded_Language
suboptimal . Repetition
greedy graph . Name_Calling,Labeling
our greedy solution ,Name_Calling,Labeling
greedy Name_Calling,Labeling
greedy Name_Calling,Labeling
greedy Name_Calling,Labeling
greedy Repetition
brute force approach Loaded_Language
greedy Name_Calling,Labeling
greedy graph Loaded_Language
greedy Loaded_Language
fancy tricks and efficient data structures Loaded_Language
greedy Loaded_Language
greedy Repetition
greedy Repetition
greedy Repetition
greedy Repetition
humans incrementally build meaning in sequential order ,Loaded_Language
Appealingly ,Loaded_Language
Colorless green dreams sleep furiously . ” Loaded_Language
Everybody loves someone else . " Slogans
The space of all strings is huge ! Loaded_Language
Everyone loves someone else . " Slogans
undecidable . Loaded_Language
Alex likes Brit . ” Name_Calling,Labeling
Alex likes some teacher . Name_Calling,Labeling
linguistically expressive formalism : Loaded_Language
Expressive power Loaded_Language
inductively defined Loaded_Language
closes the loop ” Loaded_Language
Semiring – ify matrix multiplication . Repetition
Shortest – path matrix multiplication . Loaded_Language
brute – force computation Loaded_Language
slow sequential recurrence ,Loaded_Language
Beam Search works competitively well Loaded_Language
boiled down Loaded_Language
our problem ? Flag-Waving
Probabilistic models . Name_Calling,Labeling
Non – probabilistic models . Name_Calling,Labeling
Discriminative . Loaded_Language
Asymptotic efficiency . Loaded_Language
Interestingly ,Doubt
non – convex or even non – differentiable . Loaded_Language
Intrinsic Evaluation Loaded_Language
intrinsic . Repetition
incorrectly classified Repetition
incorrectly classified Repetition
null hypothesis . Repetition
powerful tool Loaded_Language
In a nutshell ,Loaded_Language
Empirical comparison Loaded_Language
Empirical Methods Loaded_Language
Empirical Methods Loaded_Language
Empirical Methods Loaded_Language
```

C.2 Propaganda detection on this very paper

```
novel Loaded_Language
novel approach Loaded_Language
relaxed assumptions Loaded_Language
epoch - making Loaded_Language
epoch - making Transformer - based masked language model . Loaded_Language
fine - tuned Name_Calling,Labeling
state - of - the - art results . Exaggeration,Minimisation
a state - of - the - art Name_Calling,Labeling
state - of - the - art Exaggeration,Minimisation
state - of - the - art Exaggeration,Minimisation
state - of - the - art deep learning models Name_Calling,Labeling
robust Loaded_Language
propagandist argument ,Name_Calling,Labeling
propaganda likelihood Repetition
slackness Loaded_Language
very heavy Loaded_Language
very heavy class imbalance Loaded_Language
propagandist argument . Name_Calling,Labeling
manually selected spans . Name_Calling,Labeling
propaganda . Repetition
propaganda . Repetition
our approach Loaded_Language
propagandist argument . Name_Calling,Labeling
propaganda ,Name_Calling,Labeling
propagandist argument Name_Calling,Labeling
propagandist argument ,Name_Calling,Labeling
Not Propaganda " Slogans
propaganda spans . Name_Calling,Labeling
skew our model ’s predictions . Loaded_Language
propaganda argument Name_Calling,Labeling
seems flawed . Loaded_Language
flawed . Loaded_Language
Not propaganda " Name_Calling,Labeling
our strategy Loaded_Language
Not Propaganda " class Name_Calling,Labeling
limitation ,Loaded_Language
Not Propaganda " Slogans
Not Propaganda " Name_Calling,Labeling
our 15th class " Not Propaganda " . Name_Calling,Labeling
our 15th class " Not Propaganda " . Name_Calling,Labeling
Not Propaganda " . Slogans
totally identified Loaded_Language
totally identified and entirely missed propaganda spans Exaggeration,Minimisation
completely identified Exaggeration,Minimisation
Red Herring Loaded_Language
Red Herring Appeal to Authority Loaded_Language
totally identified Exaggeration,Minimisation
indepth analysis Loaded_Language
very small Exaggeration,Minimisation
one of the biggest weakness of our model . Exaggeration,Minimisation
our model . Doubt
Interestingly , the model . Doubt
propaganda spans Name_Calling,Labeling
potentially exponential number Loaded_Language
achieves very good scores Loaded_Language
very good scores Loaded_Language
our work ,Doubt
```