IT UNIVERSITY OF COPENHAGEN

Combining BERT and Hand-Crafted Features for Identification of Propaganda Techniques in News Media

Anders Friis Kaas and Viktor Torp Thomsen

Bachelor thesis
Data Science
Anders Friis Kass, anfk@itu.dk
Viktor Torp Thomsen, vikt@itu.dk

May 15, Spring 2020
Supervisor: Barbara Plank, bapl@itu.dk
Project title: Propaganda technique detection
Course code: BIBAPRO1PE

ABSTRACT

The identification of communication techniques in news articles such as propaganda is important, as such techniques can influence the opinions of a large number of people. Most work so far has focused on identification at the news article level. Recently, a new data set and shared task has been proposed for the identification of propaganda techniques at the finer-grained span level. This thesis describes our system submission to the sub-task of technique classification (TC) for the SemEval 2020 shared task on detection of propaganda techniques in news articles. We propose a method of combining neural BERT representations with hand-crafted features via stacked generalization. Our model has the added advantage that it combines the power of contextual representations from BERT with simple span-based and article-based global features. We present an ablation study which shows that even though BERT representations are very powerful for this task, they still benefit from being combined with carefully designed task-specific features.

CONTENTS

PREFACE

We would like to take this opportunity to first of all thank our supervisor, Barbara Plank, for her invaluable guidance and support. Thank you for all the inspiration and the countless ideas even through the difficulties brought on by the nationwide lockdown.

This thesis is an extended version of our system description paper in the SemEval 2020, Task 11 TC sub-task, which can be found in Da San Martino et al. (2020).

1 INTRODUCTION

Propaganda has been widely used throughout history. From the ancient Indian political book Arthashastra written around 300 B.C.E (Boesche, 2003) to the growth of mass media in the late 19th and early 20th century, propaganda has been employed as a communication technique in some form. Even in recent journalism, fragments of propaganda are occasionally used by news managements to shape or misconvey information (Jowett and O'Donnell, 2018, p.1,p.97).

Propaganda is a type of communication which aims to elicit some specific emotional response in the receiver that is desired by the propagandist, or to foster some predetermined agenda (Jowett and O'Donnell, 2018, p.1). Even though the purposes of propaganda and fake news can be quite similar, most definitions of fake news differ from those of propaganda. Fake news is generally defined and understood as fabrication of false news stories (Tandoc Jr et al., 2018), whereas propaganda is defined as deliberate selection or distortion of facts and arguments used to influence public opinion (Smith, 2020).

Recently, topics related to misinformation have seen a rise in attention. For instance, in the 2016 US presidential elections, it has been observed that highly partisan right-wing news outlets were partly responsible for the election of Donald Trump (Faris et al., 2017). Additionally, Post-truth was even announced as the word of the year by the Oxford Dictionary in 2016, as it had transformed from being a peripheral term to frequently being used in both major media publications and in academia (Golovchenko et al., 2018; Oxford-Languages, 2016).

Efforts towards inhibiting the spread of propaganda and misinformation in general have traditionally consisted of manual fact-checking. In recent years, many online services have been launched that promise to deliver non-partisan and transparent fact-checking of news. This includes sites such as American PolitiFact[1] or British BBC Reality Check.[2] However, as the amount of online news content is rising rapidly, it has become increasingly difficult to fact-check even a small fraction of published articles.

In the wake of this explosion of online content, studies were conducted that focused on creating machine learning models that automatically label whole news articles or even entire news outlets as propagandist (Barrón-Cedeño et al., 2019; Rashkin et al., 2017). The reason for this coarse classification of propaganda is that only entire news outlets had been deemed propagandist and labels for individual articles were missing. For this reason, most of the articles from the same news outlet were labeled identically, as they were classified based on their news outlet. These coarse-grained data corpora, which transfer labels from a news outlet to its articles, introduced noise to the models powering the propaganda classifications (Da San Martino et al., 2019). Outlets that were classified as trustworthy could therefore publish propagandist articles and still have them classified as non-propagandist and vice versa.

To increase the granularity of propaganda technique detection, a new corpus was developed in 2019 by (Da San Martino et al., 2019) in collaboration with a professional annotation company called A Data Pro. This data set contains articles from 48 news outlets which have been labeled either propagandist or non-propagandist in nature by *Media Bias/Fact Check*.[3] The data set contains 451 articles and 7,485 total instances of propaganda. As part of the SemEval conference of 2020, a competition involving a similar data set was launched. The competition, called "SemEval

[1] https://www.politifact.com
[2] https://www.bbc.com/news/reality_check
[3] https://mediabiasfactcheck.com

2020, Task 11: 'Detection of propaganda techniques in news articles,'" Da San Martino et al. (2020) contained two sub-tasks. The first sub-task, called span identification (SI), involved detecting in which fragments of news articles instances of propaganda were present. The second sub-task, called technique classification (TC), was a multi-class classification problem in which a system needs to identify the propaganda techniques given the spans of an article that contained propaganda. For instance, when given the span "stupid and petty" the system should classify it as `Loaded_Language`.

The TC sub-task was a shared task and was structured as follows: in September 2019, a training set consisting of articles and the propaganda-carrying spans from those articles were released along with the corresponding labels. Additionally, a development set was released which only contained the articles and their spans. The teams competing in the competition could then submit their predictions on the development set to the competition website where the predictions were evaluated using micro-averaged F1.[4] The development set was thus mainly used to gauge model performance and in turn guide our decisions. A leader board was displayed on the website with which the competing teams could compare their F1 score on the development set against the scores of the other teams. A final test set was then released in March 2020, and the scores on that set represented the final evaluation of the systems of the competing teams. After the test set submission deadline, the development set labels were released.

By participating in the TC sub-task at the SemEval 2020 Task 11, we seek to explore the following questions:

- RQ1: To what extent can modern natural language processing (NLP) techniques, in particular the BERT language model, be used to identify propaganda techniques within news articles?
- RQ2: can traditional machine learning approaches, e.g. feature engineering and model stacking, still be used to improve the performance of BERT, or are the new Transformer based models making these approaches obsolete?

In this thesis we present our final solution and results (team DiSaster, finishing at 11th place) to the TC sub-task.[5] Our system is an ensemble model based on stacked generalization (Wolpert, 1992) which enables the incorporation of both traditional engineered features (Nalini and Sheela, 2014) and the Transformer (Vaswani et al., 2017) based language model BERT (Devlin et al., 2018).

The rest of the thesis is structured as follows: In chapter 2, we will briefly describe similar studies and approaches to fine-grained propaganda detection. Chapter 3 describes the provided data, its distributions and some of the trends. Our model is described in detail in chapter 4 along with a description of our experiments. All results are presented in chapter 5, where we also discuss some of our findings and results. In the last chapter, chapter 6, we present our conclusive remarks along with some ideas for future work and limitations.

[4] Both the SI sub-task and the TC sub-task as well as the evaluation metric are described in detail at https://propaganda.qcri.org/semeval2020-task11/

[5] All code for replication is publicly available at https://github.com/ViktorTorp/SemEval2020-TC

2 BACKGROUND

2.1 EARLY WORK

In some of the first research exploring propaganda identification using NLP, Rashkin et al. (2017) compared the language characteristics used in trustworthy news articles with those of propaganda, satire and hoaxes. In order to analyse these characteristics, they created a corpus containing news articles from trustworthy news outlets along with articles from unreliable news sources. This corpus also allowed them to perform multi-class classification experiments, in which they sought to predict whether an article is trustworthy, satire, a hoax or propaganda. In addition to this coarse-grained propaganda detection on document-level, they also created a model for fine-grained truthfulness prediction on individual statements. This model was trained on a data set collected from fact-checking sites such as PolitiFact, where the truthfulness of statements are rated on a 6-point scale.

These data sets have since been used by Barrón-Cedeño et al. (2019) in order to create the first publicly available system for propaganda detection in news media. The primary goal of this system is to limit the impact of propaganda and disinformation by raising awareness. Their system, called Proppy, works by periodically fetching articles from a variety of news outlets and analyzing them using a maximum entropy classifier which results in a so-called propaganda index between 0 and 1.

2.2 FINE-GRAINED PROPAGANDA DETECTION

In order to overcome the limitations of these coarse models and their potential bias arising from the transfer of labels from news outlets to their articles, Da San Martino et al. (2019) formulated the new task of fine-grained propaganda detection. To solve this task, they developed a new corpus, which would enable models to jointly identify fragments of propaganda within a document and classify their respective propaganda techniques (Da San Martino et al., 2019; Yu et al., 2019).

This corpus consists of a total of 451 articles from 13 propagandist and 36 non-propagandist media outlets as labeled by *Media Bias/Fact Check*. In total, the articles contain around 350,000 word tokens. A total of 82.5% of the articles were pulled from propagandist sources. Professional annotation company A Data Pro was then hired to identify fragments containing propagandist language as well as the specific technique that was used. Each span could contain more than one technique, and in total 7,485 instances of propaganda were identified in the articles.

In addition to formulating the original problem of fine-grained propaganda identification and creating the corpus needed to solve the task, Da San Martino et al. (2019) also designed a multi-granularity network (MGN) as their solution to this task. This model outperformed several strong BERT baseline models in the high granularity fragment-level classification by using information from low granularity classification (e.g. document-level) to drive higher-granularity classification (e.g. paragraph-level). This approach ensured that low-granularity sections of a document, that were classified as very unlikely to contain propaganda, could not contain high-granularity fragments that were classified as likely to contain propaganda.

As part of the NLP for Internet Freedom (NLP4IF) workshop at EMNLP-IJCNLP 2019, a shared task on fine-grained propaganda detection was hosted, which included the two sub-tasks Sentence Level Classification (SLC) and Fragment Level

Classification (FLC) (Da San Martino et al., 2019). The first sub-task, SLC, focused on the binary classification problem of identifying sentences containing propaganda. The objective of the second sub-task, FLC, was to detect all segments of propaganda while simultaneously identifying which of 18 different propaganda techniques they corresponded to. E.g. in the fragment "acted like babies" the span "babies" would be labeled as both `Name_Calling` and `Labeling`.

The winning solution for the FLC sub-task in the NLP4IF'2019 workshop was developed by Yoosuf and Yang (2019). Their solution was based on a fine-tuned BERT model used for token-level prediction, i.e. word-level classification, while using an oversampling strategy to overcome the class imbalance. Their research showed that utilizing a proper oversampling technique can improve a fine-tuned BERT model. In their strategy, sentences containing propaganda were weighted higher than sentences without propaganda, as it helped the model identify propaganda despite the amount of sentences without propaganda in the training data being much larger than the amount of sentences containing propaganda.

Another solution to the FLC task was developed by Alhindi et al. (2019). Their solution to the problem used a stacking of GloVe embeddings (Pennington et al., 2014) and Word2Vec embeddings (Mikolov et al., 2013) trained on Urban Dictionary definitions as input to a BiLSTM-CRF (Huang et al., 2015). Due to the limited amount of training data available, they also created one-hot encoded features representing concepts associated with the individual words such as *vulgar* or *offensive* by using the UBY dictionary from Gurevych et al. (2012). The results from this model showed that the techniques which occurred more frequently and the techniques with strong lexical signals had a much higher F1 score. However, even though `Repetition` was the third most common technique, their model was unable to identify it, and thus its F1 score was 0.0. As a result, Alhindi et al. (2019) concluded that some of the propaganda techniques can not be identified by solely looking at the sentence. Instead, the system needs to include more information from the article in order to understand the context.

2.3 BERT AND CONTEXTUALIZED REPRESENTATIONS

What all of the best performing models in the FLC subtask had in common was that they used a powerful language model as a central component of their learning setup. In particular, the pre-trained BERT model (Bidirectional Encoder Representations from Transformers) or one of its many variants was used in many of the models. BERT was developed and made publicly available by researchers at Google in 2018 (Devlin et al., 2018) and makes use of the Transformer model (Vaswani et al., 2017).

One of the advantages of using BERT over previous static word embedding models is that BERT can generate contextualized representations of input tokens. For a given word, static word embedding models produce the same embedding vector regardless of the context, i.e. they fail to capture polysemy or the semantics of idioms. Contextualized word embedding models, on the other hand, produce different embedding vectors of a given word depending on the context that it is in (i.e. the words that surround it). This means that the semantics of the input tokens are better preserved.

The idea of contextualized representations was introduced by Peters et al. (2018) with a model called ELMo (Embeddings from Language Models). ELMo is a deep bidirectional language model that uses the novel approach of linearly combining the internal states of a bidirectional LSTM model that was trained on next word prediction. At the time of its development and public release, ELMo achieved state-of-the-art results on a number of NLP benchmarks.

Several other models for generating contextualized representations have since been proposed. Some noteworthy models among those are ULMFiT (Howard and

Ruder, 2018), GPT-2 (Radford et al., 2019) (both of which also achieved state-of-the-art results on several benchmarks) and, of course, BERT (Devlin et al., 2018).

BERT uses Transformer encodings and, like ELMo, conditions bidirectionally. However, as that would implicitly allow each word to see itself, a random subset of 15% of the tokens of each input sentence are masked at training time. Furthermore, BERT was trained to predict the relationship between two sentences, e.g. predicting whether sentence B follows sentence A. Additionally, a special classification token known as the [CLS] tag is learned, which can be thought of as a summary of the input sequence. The [CLS] token is useful for sequence classification.

One key difference between BERT and previous contextualized embedding models is that instead of using recurrent neural networks to condition on the context, it uses Transformers. In short, the Transformer (Vaswani et al., 2017) is an encoder-decoder model that uses multi-head attention and positional encoding to condition on the context. As each token in a sequence can be processed independently, the Transformer allows for more parallelization compared to recurrent neural networks which decreases training time.

BERT comes in several different variants, with the two most common called BERT_base and BERT_large. The overall architecture of these two variants is the same, but BERT_large has around three times as many trainable parameters as BERT_base with 340 million parameters against 110 million parameters. Using BERT for downstream tasks usually consists of "fine-tuning" it for the task at hand instead of training the whole network from scratch. Fine-tuning is an approach to machine learning in which all the model parameters of a pre-trained network are adjusted to optimize the model for a particular task. This stands in contrast to the feature-based approach in which the output of a pre-trained model is solely used as a feature in another task-specific architecture. An example of the feature-based approach is to use static pre-trained word embeddings (such as GloVe (Pennington et al., 2014) or Word2vec (Mikolov et al., 2013)) as input to a neural network.

2.4 OTHER WORK

In a recent study by Wang et al. (2020), a new model named LatexPRO (logical and textual knowledge for propaganda detection) was developed which was able to outperform the original MGN from Da San Martino et al. (2019). LatexPRO, works by incorporating knowledge expressed in both natural-language (the text containing the definition of the different propaganda techniques) and first order logic for fine-grained propaganda detection.

The idea of combining BERT with hand-crafted features has been explored in previous studies. In a study from 2019, Zhang and Li (2019) developed a question answering model that augmented BERT with additional linguistic features. Each token in the input sequence was augmented with a named entity label, a part-of-speech tag, the syntactic dependencies and a stop-word tag. It was shown that these features enhanced the model's understanding of the context. Their model was able to outperform the BERT_base model on the SQuAD question answering data set.

3 DATA AND MATERIAL

3.1 COLLECTION AND ANNOTATION

The corpus used in the TC sub-task of SemEval Task 11 consists of 550 articles and 8,979 total instances of propaganda classified into 18 different propaganda techniques. All of the articles had been retrieved with the `news paper3k`[1] library and all of the sentences were separated automatically with the `NLTK sentence splitter`[2] (Da San Martino et al., 2020). The provided data consists of: (1) a 'training' folder that contains all of the training articles as `.txt` files as well as a file that includes the start and end points of all fragments of propaganda as character offsets in the articles along with the corresponding gold labels and article ids (see Figure 1), and (2) two folders called 'development' and 'test' which include all the same data as for the training data except for the gold labels.

id	technique	begin_offset	end_offset
123456	Name_Calling,Labeling	34	40
123456	Black-and-White_Fallacy	299	368
123456	Loaded_Language	400	416
123456	Exaggeration,Minimization	607	653
123456	Loaded_Language	635	653

Figure 1: An excerpt of the label file for the training data, which contains the article id and gold label for propaganda spans. The beginning and end points are given by the characters offset within the entire article. Figure reproduced from https://propaganda.qcri.org/semeval2020-task11/index.html.

Figures 1 and 2 illustrate an example of a training article where all its propaganda spans are visualized and annotated. The propaganda fragments are indicated by their beginning and end offsets (see Figure 1), which are the absolute positions of the characters within the article (see Figure 2). E.g. the first propaganda span in Figure 1 can be found in the article 123456 (Figure 2) by retrieving the characters from the beginning offset at character number 34 to the end offset at character number 40. Thus, the span "babies" is retrieved which has the propaganda technique `Name_Calling,Labeling`. As shown in these figures, two different techniques can have overlapping spans. E.g. in the last two spans in Figure 1, the beginning and end offsets are overlapping for the two fragments.

3.2 PROCESSING

All of our hand-crafted features, described in section 4.1.2, were extracted both on the original data (i.e. without any pre-processing other than what is described in section 4.1.2) and on a cleaned version of the data. The cleaned version of the data was created by expanding some of the most common English contractions to their full form (e.g. "I'm" was transformed to "I am" and "where's" was transformed to "where is"). In addition to this, all non-alphabetic characters were removed. Both

[1] https://newspaper.readthedocs.io/en/latest/
[2] https://www.nltk.org/api/nltk.tokenize.html

[0]Manchin says Democrats acted like [34]babies[40] at the SOTU (video) Personal Liberty Poll Exercise your right to vote.
Democrat West Virginia Sen. Joe Manchin says his colleagues' refusal to stand or applaud during President Donald Trump's State of the Union speech was disrespectful and a signal that [299]the party is more concerned with obstruction than it is with progress[368].
In a glaring sign of just how [400]stupid and petty[416] things have become in Washington these days, Manchin was invited on Fox News Tuesday morning to discuss how he was one of the only Democrats in the chamber for the State of the Union speech [607]not looking as though Trump [635]killed his grandma[653].
When others in his party declined to applaud even for the most uncontroversial of the president's remarks, Manchin did.
He even stood for the president when Trump entered the room, a customary show of respect for the office in which his colleagues declined to participate.

Figure 2: An example of an article file (e.g. 123456.txt). The superscripts indicating character offsets are not included in the provided articles but are shown here for didactic purposes. Figure reproduced from `https://propaganda.qcri.org/semeval2020-task11/index.html`.

modifications were performed using regular expressions written with the Python library `regex`.[3] However, our final model used only the features obtained from the original data, as they continuously outperformed the features from the cleaned versions.

3.3 DATA

The provided training data for the competition contains a subset of 371 articles in which 6,129 fragments of propaganda are annotated with one of the 18 different propaganda techniques described in Da San Martino et al. (2019). However, due to a low frequency of some of the techniques, similar underrepresented techniques were merged into a superclass, while one of the techniques was eliminated completely. Thus, the TC task was a 14-class classification problem, where two of the classes were superclasses representing several techniques each (Da San Martino et al., 2020). Below, the final 14 classes of propaganda techniques along with examples from the training data are listed (Da San Martino et al., 2020, 2019)

- `Loaded_Language`: emotionally charged phrases. Example: *"Nearly five hours of unprecedented and surreal talks..."*.
- `Name_Calling,Labeling`: labeling a person or an idea as something that evokes negative connotations in the audience. Example: *"brutal dictator"*.
- `Repetition`: Repeating a word or phrase multiple times in order to make it seem true or more important than it is. The repetition may both be within a span or of text occurring elsewhere in the article. Example: the word 'doubt' in *"This person said without a shred of doubt, we knew in 2009 a year before CFIUS ruled that Thomas were engaged in criminality, without a shadow of a doubt, we knew that Russia was trying to gain a corner on the U.S. market, get a strong hold on our uranium, and without a doubt we knew they were using political levers to try to get their way here."*

[3] `https://docs.python.org/3/library/re.html`

- `Doubt`: questioning either the truthfulness of something or the credibility of someone. Example: *"The US is blatantly telling lies."*
- `Exaggeration,Minimisation`: either representing something in an excessive manner, e.g. making it seem more important or bigger than it actually is, or minimizing the importance of something. Example of `Exaggeration`: *"the biggest story of treason and espionage in the recent memory."*
- `Appeal_to_fear-prejudice`: seeking to associate an idea with fear or panic in the audience so as to further an alternative idea. Example: *"war against North Korea would be a slow and bloody slog."*
- `Flag-Waving`: appealing to patriotism or a feeling of belongingness. Example: *"This is not the Soviet Union, this is not Iran or Riyadh – this is America."*
- `Causal_Oversimplification`: assuming one cause when there could be many. Also includes scapegoating which is blaming a single person or group for some outcome. Example: *"There could be only one answer: communists."*
- `Appeal_to_Authority`: stating that a claim is true because an expert has said so without further evidence. Example: *"As Robert Spencer advises, there needs to be legislation that will bar all such groups and affiliated individuals from advising the government or receiving any grants from it."*
- `Slogans`: a short, striking and often emotionally charged phrase. Example: *"Stop islamization of America."*
- `Whataboutism,Straw_Men,Red_Herring`: whataboutism involves dismissing an opponent's argument by charging them with hypocrisy. A straw man is a misrepresentation of the opponents argument. A red herring is introduction of irrelevant facts in order to shift the audience's attention away from the opponent's argument. Example of `Red_Herring`: *"As much as we may be appalled by some of the things that Al Franken has done, the truth is that there are far worse offenders in Washington."*
- `Black-and-White_Fallacy`: presenting two options as being the only possible options when there may be more. Example: *"After the Trumpian revolt, there is no going back."*
- `Thought-terminating_Cliches` Short and often vague phrases intended to inhibit critical thought about a topic. Example: *"Such injustices are bound to occur in any judicial system."*
- `Bandwagon,Reductio_ad_hitlerum`: bandwagon: trying to persuade the audience to take some course of action or believe an idea because "everyone else does." Reductio ad Hitlerum: rejecting an idea or a person by associating it/them with someone unfavorable (originally Hitler but the definition has since expanded). Example of `Reductio_ad_hitlerum`: *"This is really getting surreal in the creepiest and most harrowing Stalinist sense."*

Table 1 contains a list of all the labels along with their respective IDs that we defined.

As illustrated in Figure 1, two different spans may be partly or fully overlapping, e.g. in article 780619695 the span "#*stopthesynod*" (labled as `Slogans`) is overlapping with the span "*stopthesynod*" (labled as `Repetition`). In total, 200 of the 371 training articles had spans that were overlapping, which resulted in 782 instances of overlapping spans. The development set for the competition contained 75 articles, and 45 of those contained overlapping spans. Of the 1062 total spans in the development set, 127 were overlapping. The test set contained 90 articles and 67 of those contained overlapping spans. In total, the test set contained 1789 spans, 494 of which were overlapping.

label	id	support	% w. 1word	Avg #words	Avg one_word_counter	Avg span_sentence_counter
Loaded_Language	8	**2123**	24.78	3.82 (±4.0)	1.65 (±2.0)	0.8 (±1.0)
Name_Calling,Labeling	9	1058	11.25	3.93 (±3.0)	1.62 (±2.0)	0.62 (±1.0)
Repetition	10	621	**43.64**	2.81 (±3.0)	**6.92** (±5.0)	**1.35** (±2.0)
Doubt	5	493	1.42	21.14 (±16.0)	1.0 (±1.0)	0.13 (±0.0)
Exaggeration,Minimisation	6	466	6.22	7.44 (±6.0)	0.97 (±0.0)	0.63 (±1.0)
Appeal_to_fear-prejudice	1	294	3.06	17.05 (±13.0)	1.44 (±1.0)	0.32 (±0.0)
Flag-Waving	7	229	11.79	10.63 (±12.0)	4.33 (±3.0)	0.22 (±1.0)
Causal_Oversimplification	4	209	0.0	21.52 (±13.0)	- (-)	0.1 (±0.0)
Appeal_to_Authority	0	144	0.0	**23.2** (±22.0)	- (-)	0.22 (±0.0)
Slogans	11	129	6.2	4.33 (±3.0)	1.25 (±2.0)	0.23 (±1.0)
Whataboutism,Straw_Men,Red_Herring	13	108	3.7	16.5 (±11.0)	2.25 (±1.0)	0.12 (±0.0)
Black-and-White_Fallacy	3	107	0.0	18.71 (±13.0)	- (-)	0.15 (±0.0)
Thought-terminating_Cliches	12	76	1.32	6.13 (±4.0)	0.0 (±0.0)	0.27 (±0.0)
Bandwagon,Reductio_ad_hitlerum	2	72	0.0	16.44 (±12.0)	- (-)	0.1 (±0.0)

Table 1: Overview of the provided training data for the SemEval 2020 Task 11 competition. The bold font indicates the highest value within a column. In the columns Avg #words, Avg one_word_counter and Avg span_sentence_counter. ±1× standard deviations are included in the parentheses.

3.3.1 Data analysis

The class distribution in the training data was very skewed (as the support in Table 1 shows); four of the labels accounted for more than 70% of the training data. As the score for the competition was calculated as the micro-averaged F1 over all the labels, it was crucial to get good predictions on these four classes. In order to create useful handcrafted features that would increase the performance of the model for these techniques, we performed a thorough data analysis whose main results are summarized in Table 1.

Table 1 shows that there is a considerable spread in the average number of words per span among the different techniques. In particular, the spans from the `Repetition` category were much shorter than the spans of the other techniques, as more than 40% of its spans contained only a single word. By examining the instances of `Repetition` in which the span only contained a single word, we found that the Porter stemmed version of the word (Porter et al., 1980) often occurred several times within the article.[4] This effect is displayed as Avg one_word_counter in Table 1, which is the average number of times the stem of single word span occurred within an article. However, this value cannot be calculated for classes in which every span contains more than one word. Furthermore, a similar effect for `Repetition` was discovered when spans with more than one word were examined. The average number of times an entire span with more than one word was repeated within an article was generally much higher for `Repetition` than any other technique. This effect is shown in Table 1 as Avg span_sentence_counter.

During our initial analysis, we found that if a label was in an article, there was a much higher probability of finding another span with the same label elsewhere in the article (see Figure 3). This may be because each individual article only has one defined agenda that it fosters. An article about US politics may for instance use the technique `Flag_waving` or `Slogans` multiple times while not using `Thought-terminating_cliches` at all. Additionally, it can be observed that `Repetition` tends to occur very frequently with other instances of `Repetition`. The reason for this is that if for instance a particular word or phrase is mentioned 9 times throughout an article, each of the 9 repetitions will be labeled as `Repetition`.

[4] E.g. in article 699291100 the stem of the word "threatened" (i.e. "threaten") was repeated 3 times.

Figure 3: Normalized matrix of labels co-occurring in the same article. The diagonal line indicates that articles tend to contain spans with the same label. The IDs correspond to the following techniques (Table 1): 0=*Appeal_to_Auhority*; 1=*Appeal_to_fear_prejudice*; 2=*Bandwagon,Rectio_ad_hitlerum*; 3=*Black-and-White_Fallacy*; 4=*Causal_Oversimplification*; 5=*Doubt*; 6=*Exaggeration,Minimisation*; 7=*Flag-Waving*; 8=*Loaded_Language*; 9=*Name_Calling,Labeling*; 10=*Repetition*; 11=*Slogans*; 12=*Thought-terminating_Cliches*; 13=*Whataboutism,Straw_Men,Red_Herring*.

4 METHODS

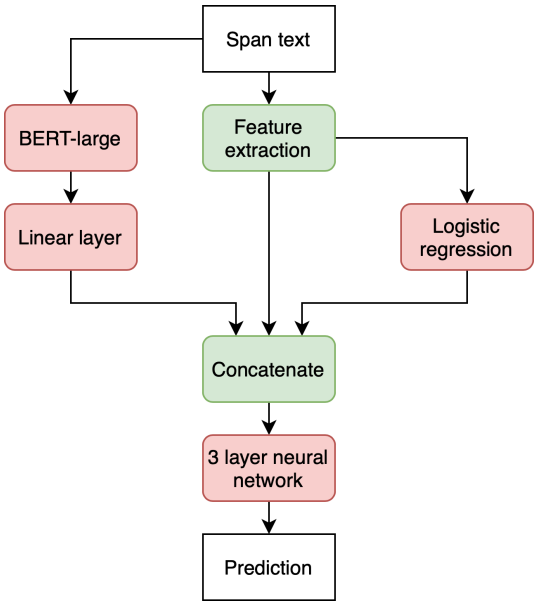

Figure 4: The full model pipeline.

4.1 MODEL OVERVIEW

We tackled the problem of propaganda technique identification as a classification task where we combined three components using stacked generalization (Wolpert, 1992). Our model is illustrated in Figure 4 and consists of: (1) a contextualized embedding representation of the span using BERT, (2) hand-crafted features extracted from both the spans and the global article structure, and (3) the scores of a traditional logistic regression model trained on the hand-crafted features. These components were combined using a feed-forward neural network as the topmost stacking classifier and were implemented in Python using the PyTorch framework (Paszke et al., 2019). All the components are described in the following subsections. In addition to the full model, we also created a simple baseline model which worked by always predicting the label with the highest prior probability. The label with the highest prior probability in the training set was `Loaded_Language`. We compare the results of this baseline model with our final model in Section 5.1, Table 2.

4.1.1 BERT fine-tuning

The BERT component of the pipeline was implemented using the Huggingface library (Wolf et al., 2019) and consists of BERT_large with a single linear layer on top of the last layer hidden-state of the classification token (the [CLS] tag), similarly to the approach described in Devlin et al. (2018). This component was used on the span-level (i.e. the actual propaganda fragments) in order to get 14 dimensional vectors of logits corresponding to the 14 propaganda technique classes. To do this, we tokenized the input sequences using the pre-trained BERT uncased tokenizer (Wolf

et al., 2019). This produced a vector of WordPiece indices (Wu et al., 2016) and a vector of attention masks.[1]

To obtain the logits from BERT for all of the training set spans, a 10-fold stratified learning strategy was employed. We created 10 stratified train/test splits from the training set. BERT_large was then initialized and fine-tuned, as suggested by Devlin et al. (2018), on each of the 10 training sets and made to predict the logits for the corresponding test sets. This was done by alternately training for one epoch on the training set and evaluating on the corresponding test set. We used early stopping to avoid overfitting when the loss of the 10% test set stopped decreasing. We did not use patience in the early stopping as the training was very stable across epochs, and any decrease in model performance would therefore not be due to randomness but to actual model degradation. This method of 10-fold stratified learning ensured that logits were predicted on the whole training set without predicting on data that it was trained on.

The logits for the development and test sets were created by fine-tuning on a stratified 90% subset of the training data and using early stopping to stop training when the loss of the remaining 10% of the training data stopped decreasing, similarly to our approach for the training data.

BERT was optimized on the cross-entropy loss using the AdamW optimizer (Loshchilov and Hutter, 2019) with a learning rate of 2×10^{-5} and an epsilon value of 1×10^{-8} .

4.1.2 Feature extraction

In addition to the BERT logits, we extracted 53 additional features from the data (all features are described in our GitHub repository (Kaas et al., 2020)). 41 of the features were normalized bag-of-words (BOW) representations of the part of speech (POS) tags of the spans. Eight of the features were created by counting the character lengths and word lengths of the spans and the sentences that the spans occurred in. These features were extracted for both the cleaned and the original versions of the data. The last four hand-crafted features were a combination of local features (created solely based on the content of the span) and global features (created based on the content of the entire article). All of the features were created using a mixture of SpaCy (Honnibal and Montani, 2017) and NLTK (Bird et al., 2009). We found that the following five improved the performance of the model:

- If the span is only one word, *article_one_word_counter* (aowc) is a count of how many times the Porter stem of that word appeared in the article. Otherwise it is 0.
- If the span is more than one word long, *article_span_sentence_counter* (assc) is a count of how many times that span appeared elsewhere in the article. Otherwise it is 0.
- *span_word_length* (swl) is a count of the number of words in the span.
- *word_count_span_sent* (wcss) is the number times that a span appears within the sentence it is presented in. E.g. the span "fake news" appears twice in the sentence "it is fake news about a fake news story."
- *word_resemble_factor* (wrf) is the inverse uniqueness of words in a span and is calculated as $\frac{\text{number of words in span}}{\text{number of unique words in span}}$.

We compare and discuss the importance of the features in Section 5.2.

[1] The attention masks were not strictly necessary for this application of BERT, but they were required as an input to the Huggingface implementation of BERT that we used (Wolf et al., 2019).

4.1.3 Logistic regression

A logistic regression was performed over the hand-crafted features alone using a similar stratified learning strategy as for BERT, and the resulting 14-dimensional output was used downstream in our pipeline.

The importance of the output of the logistic regression is also discussed in Section 5.2.

4.1.4 Feed-forward network

The last component of our model is a fully connected feed-forward neural network (FFNN) with three hidden layers each consisting of 500 neurons. The input layer and the three hidden layers were activated with the GELU activation function (Hendrycks and Gimpel, 2016). The network was optimized using the AdamW optimizer with a learning rate of 2×10^{-5} and an epsilon value of 1×10^{-8} on the cross-entropy loss. As illustrated in Figure 4, the network was fed a concatenation of BERT logits, the hand-crafted features and the output of the logistic regression over the features. The resulting output was a 14 dimensional vector of logits corresponding to the 14 propaganda classes.

Similarly to our approach for training BERT, we alternately trained the network for one epoch on a stratified 90% subset of the training data and evaluated on the remaining 10%. In this instance, the loss of the 10% development set was less stable across epochs. To find the best model parameters while not overfitting, we therefore trained the model for 150 epochs while evaluating on the development set after each one. Additionally, after each epoch we logged the parameter configuration along with its corresponding loss on the development set. Afterwards, we simply used the configuration with the lowest loss.

4.1.5 Computational considerations

The most compute-intense step of our model was extracting the BERT task-specific representations for the entire dataset (as outlined in Section 4.1.1). Fine-tuning BERT and obtaining the representations took roughly 12 hours using the Tesla K80 GPU available on Google Colab. However, once the model was trained, new representations could be obtained in seconds. The extraction of global features was quicker, taking less than 30 minutes for both the training, development and test set on a 2017 MacBook Pro with 3,1 GHz Quad-Core Intel Core i7 processor. A clear advantage of our model is its simplicity. Once the BERT features are extracted, our stacked model can be trained in about five minutes.

4.2 ABLATION STUDY

To test the importance of the different components, a feature ablation study was performed. The study was performed by ablating a feature or component (such as the logistic regression) from the data set before running the data through the final FFNN. A 10-fold stratified cross-validation was then performed on the training set and the micro-average F1 score was recorded. Additionally, we created predictions for the development set using the ablated model and recorded the scores we received after uploading the predictions to the official competition website.

5 RESULTS AND DISCUSSION

5.1 RESULTS

The performance of our model as measured in F1 per class and micro-averaged F1 across the official development and test sets as well as the micro-averaged F1 score of the cross-validation on the training set can be seen in Table 2. The results of our ablation study are shown in Table 3. A confusion matrix for the predictions on the development set can be seen in Figure 6 and a confusion matrix for the predictions on the cross-validation of the training set can be seen in Figure 7.

5.2 DISCUSSION

As evident from the results of the ablation study (Table 3), the most important component of our learning setup was BERT. However, as BERT is used at the span-level, it is only able to predict a label based on the tokens in a given span. Due to this local behavior, BERT alone was struggling to correctly predict the `Repetition` class. This was most likely because the words or phrases that were repeated were not necessarily in the span, but spread throughout the article, which the data exploration in Section 3.3 also supports. However, this was a problem as `Repetition` was the third most frequent class in the training set.

It was partly for this reason that we decided to extract and use additional global (article-level) features from the data set. The most important extracted feature for `Repetition` was the *article_one_word_counter*. This feature directly tells the final FFNN if a word has been repeated in the article, and removing this global feature shows its importance as the F1 score for `Repetition` drops by 0.025 (Table 3). This is also supported by our data analysis (Section 3.3, Table 1) which shows that the Avg one_word_counter is significantly higher for `Repetition` compared to the other labels. These results also align with the conclusion from Alhindi et al. (2019), in which they suggest that global features would be necessary in order to get good results for techniques such as `Repetition`. With regard to RQ2, the fact that we obtained better quality predictions by augmenting BERT-predictions with additional information about the text shows that feature engineering is still a relevant discipline as other recent research also suggests (Wang et al., 2020; Zhang and Li, 2019; Wu et al., 2018).

The augmented BERT approach worked well on both the training set and the development set, but our score dropped significantly when predicting on the test set (0.628 dev set micro F1 $\rightarrow$ 0.566 test set micro F1). As we do not have access to the test set labels, a detailed error analysis is difficult for now and is left for future work. However, by comparing the F1 scores from the official test set with the cross-validation scores in Table 2, we do see a particularly large drop in F1 for the `Repetition` category (from 0.646 cross-validation $\rightarrow$ 0.204 test). This drop in `Repetition` F1 can also be observed for the other participants in the competition. This may be due to overfitting the model to the training and development sets. It may also be due to the test set having a slightly different distribution than the training and development sets.

In fact, even between the training set and the development set, we see a rather large difference in the distribution of labels (see Figure 5). We performed a Chi-squared goodness of fit test with 13 degrees of freedom to see if the development

	Cross validation training set		development set		Test set
	Baseline	Full model	Baseline	Full model	Full model
Micro-averaged F1 score	0.346	0.672	0.31	0.628	0.566
Appeal_to_Authority	0.000	0.341	0.000	0.300	0.512
Appeal_to_fear-prejudice	0.000	0.447	0.000	0.353	0.353
Bandwagon,Reductio_ad_hitlerum	0.000	0.162	0.000	0.286	0.204
Black-and-White_Fallacy	0.000	0.200	0.00	0.000	0.267
Causal_Oversimplification	0.000	0.433	0.000	0.231	0.146
Doubt	0.000	0.640	0.000	0.562	0.591
Exaggeration,Minimisation	0.000	0.530	0.00	0.483	0.306
Flag-Waving	0.000	0.622	0.000	0.752	0.583
Loaded_Language	0.515	0.796	0.468	0.767	0.745
Name_Calling,Labeling	0.000	0.792	0.000	0.773	0.681
Repetition	0.000	0.646	0.000	0.471	0.204
Slogans	0.000	0.518	0.000	0.351	0.426
Thought-terminating_Cliches	0.000	0.343	0.000	0.091	0.190
Whataboutism,Straw_Men,Red_Herring	0.000	0.157	0.000	0.056	0.043

Table 2: Results. The row in the top section shows the micro-averaged F1 scores across the official development test sets and the micro-averaged F1 score on the 10-fold cross-validation of the training set. The rows in the bottom section are the individual F1 scores per class from the cross-validation of the training set and on the development set.

	- BERT	- LR	- HCF	- HCF & - LR	- Finetuning	- wrf	- aowc	- assc	- swl	- wcss
Cross validation training set	-0.229	-0.003	-0.004	-0.005	-0.205	-0.001	-0.002	-0.003	-0.005	-0.001
Development set	-0.192	-0.013	-0.030	-0.037	-0.326	-0.011	-0.024	-0.019	-0.010	-0.009
Appeal_to_Authority	-0.326	-0.025	-0.033	-0.037	-0.298	+0.001	+0.016	-0.018	-0.048	+0.010
Appeal_to_fear-prejudice	-0.447	+0.015	-0.005	+0.005	-0.284	+0.028	+0.008	+0.025	+0.001	+0.009
Bandwagon ,Reductio_ad_hitlerum	-0.162	-0.004	+0.042	+0.052	-0.162	-0.010	+0.006	+0.042	-0.002	+0.012
Black-and-White_Fallacy	-0.200	-0.002	+0.079	+0.055	-0.200	-0.010	+0.036	-0.027	+0.081	+0.040
Causal_Oversimplification	-0.433	-0.028	-0.009	+0.008	-0.377	-0.008	+0.023	+0.001	+0.018	+0.004
Doubt	-0.256	+0.002	-0.001	-0.007	-0.214	+0.003	+0.010	-0.002	-0.004	-0.004
Exaggeration,Minimisation	-0.530	+0.005	+0.002	+0.016	-0.378	+0.001	-0.005	-0.006	-0.004	-0.002
Flag-Waving	-0.622	-0,010	-0.015	-0.016	-0.249	-0.005	+0.011	-0.002	-0.018	0.000
Loaded_Language	-0.166	-0.003	0.000	-0.003	-0.162	-0.003	-0.006	-0.007	-0.006	0.000
Name_Calling,Labeling	-0.454	-0.002	-0.005	-0.003	-0.389	-0.005	-0.006	-0.007	-0.007	-0.004
Repetition	-0.105	-0.008	-0.027	-0.031	-0.103	-0.008	-0.025	-0.012	-0.008	-0.008
Slogans	-0.518	-0.004	+0.002	+0.012	-0.346	-0.002	+0.012	-0.008	+0.008	+0.011
Thought-terminating_Cliches	-0.343	-0.025	-0.002	-0.012	-0.343	-0.059	-0.022	-0.050	-0.022	-0.044
Whataboutism,Straw_Men ,Red_Herring	-0.157	-0.001	-0.016	+0.005	-0.157	+0.016	-0.013	-0.026	-0.030	-0.056

Table 3: Results of the ablation study. The values are absolute differences in F1 with respect to the full model (Table 2). The columns are ablated features and components. The rows in the top section are the differences in micro-averaged F1 scores on the cross-validation of the training set and on the official development set. The rows in the bottom section are the differences in individual F1-scores per class from the cross-validation of the training set. The abbreviations in the columns are: LR (logreg), HCF (hand-crafted features), wrf (word_resemble_factor), aowc, (article_one_word_counter), assc (article_span_sentence_counter), swl (span_word_length), wcss (word_count_span_sent).

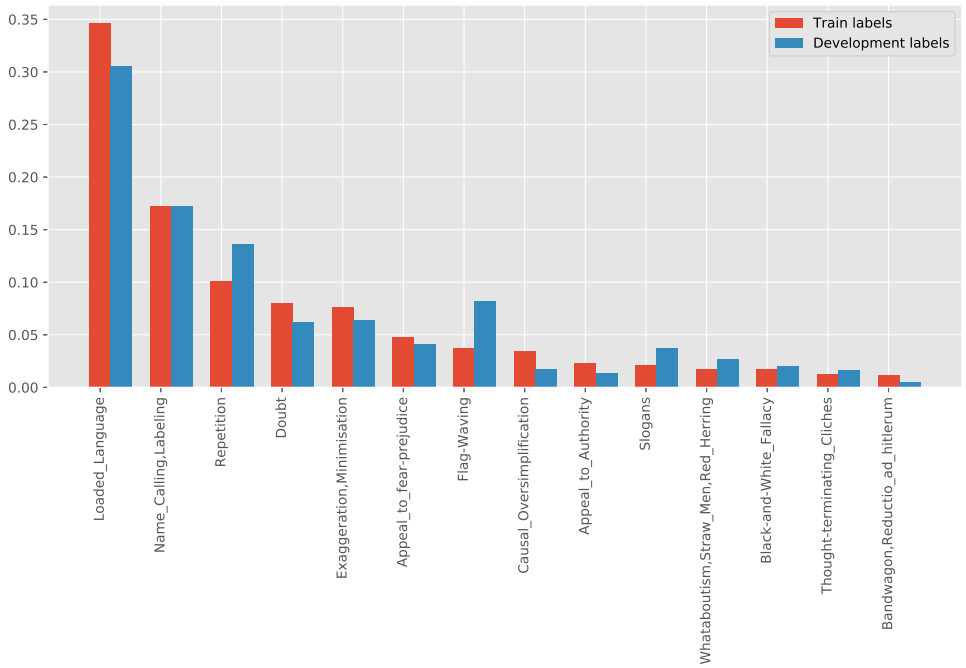

Figure 5: Label distribution in the training data and the development data.

set labels came from the same distribution as the training set labels. This resulted in a test statistic of 125.18 and a p-value of 1.90×10^{-20}, indicating that the two distributions are indeed different.

Figure 6 visualizes the confusion matrix from our full model on the official development set and is thus illustrating the performance of the model. From this figure we see that the model generally performs quite well on the larger classes such as `Loaded_Language`, `Name_Calling,Labelling` and `Exaggeration,Minimisation`. Similar to the results from Alhindi et al. (2019), the model also performs really well on the `Flag-Waving` category. Even though `Flag-Waving` is not one of the larger classes, the model is still able to recognise it well with a recall of 0.79, as the language used in these fragments carries strong lexical signals (Alhindi et al., 2019). Usually, the `Flag-Waving` spans contain words pertaining to nationality such as "America" or "Russian", e.g. *"This is not the Soviet Union, this is not Iran or Riyadh – this is America."*.

In addition, the F1 score for `Repetition` dropped from 0.646 on the 10-fold cross-validation to 0.47 on the official development set (see Table 2). In particular, the recall of `Repetition` on the development set is much lower than on the training set (see Figures 6 and 7). This can be interpreted as our model failing to recognise when a label actually is `Repetition`, and that it often mistakes it for other techniques. Overall, 47.6% of all true instances of `Repetition` in the development set were wrongly classified as either `Loaded_Language` or `Name_Calling,Labelling` (Figure 6), whereas in the results obtained from the cross-validation of the training data (Figure 7), only 27.2% of the true instances of `Repetition` were misclassified as `Loaded_Language` or `Name_Calling,Labelling`, which is remarkably lower. This could indicate that the instances of `Repetition`, `Loaded_Language` and `Name_Calling ,Labelling` in the development set differ somehow from the instances in the instances in training data.

These differences could cause:

1. Our hand-crafted features to no longer represent some trend in `Repetition` alone but rather some behavior found in both `Repetition`, `Loaded_Language` and `Name_Calling,Labelling`

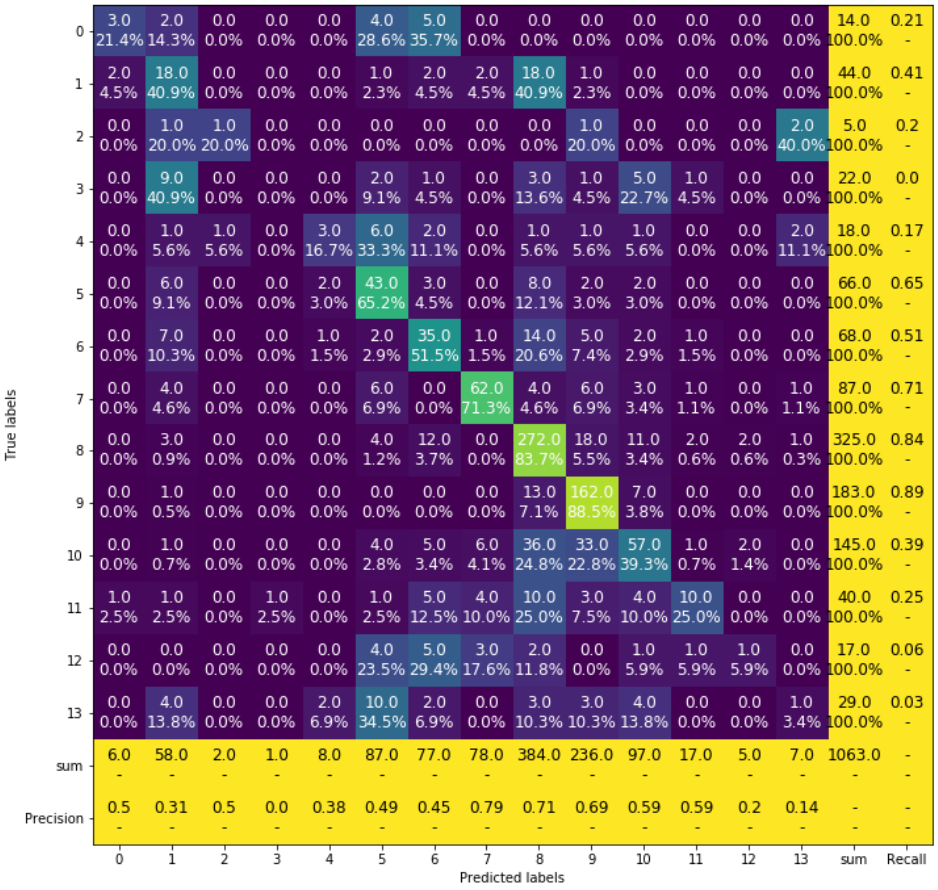

Figure 6: Normalized confusion matrix from the full models prediction on the official development set. The IDs correspond to the following techniques (Table 1): 0=*Appeal_to_Auhority*; 1=*Appeal_to_fear_prejudice*; 2=*Bandwagon,Rectio_ad_hitlerum*; 3=*Black-and-White_Fallacy*; 4=*Causal_Oversimplification*; 5=*Doubt*; 6=*Exaggeration,Minimisation*; 7=*Flag-Waving*; 8=*Loaded_Language*; 9=*Name_Calling,Labeling*; 10=*Repetitiono*; 11=*Slogans*; 12=*Thought-terminating_Cliches*; 13=*Whataboutism,Straw_Men,Red_Herring*.

2. Some of the text within the spans of these instances to be more similar in the development set, making the distinction between the techniques more difficult for BERT.

3. A combination of the two effects.

These differences could also explain the drop in precision for `Loaded_Language` and `Name_Calling,Labelling` from the results obtained by the training set (Figure 7) to the results for the development set (Figure 6).

By examining Figure 6, we see that the model is quite bad at identifying the classes with a low frequency such as `Black-and-White_Fallacy` and `Whataboutism`, `Straw_Man,Red_Herring`. Our model failing to identify the correct label in these classes indicates that the performance of the model could be enhanced by collecting more articles that contain propaganda techniques from the smaller classes.

5.2.1 Attempts at exploiting label co-occurrence information

As mentioned in Section 3.3 and visualized in Figure 3, the labels were not distributed uniformly throughout the articles. In particular, if a technique was used in an article, there was a much higher chance than expected of finding it elsewhere in

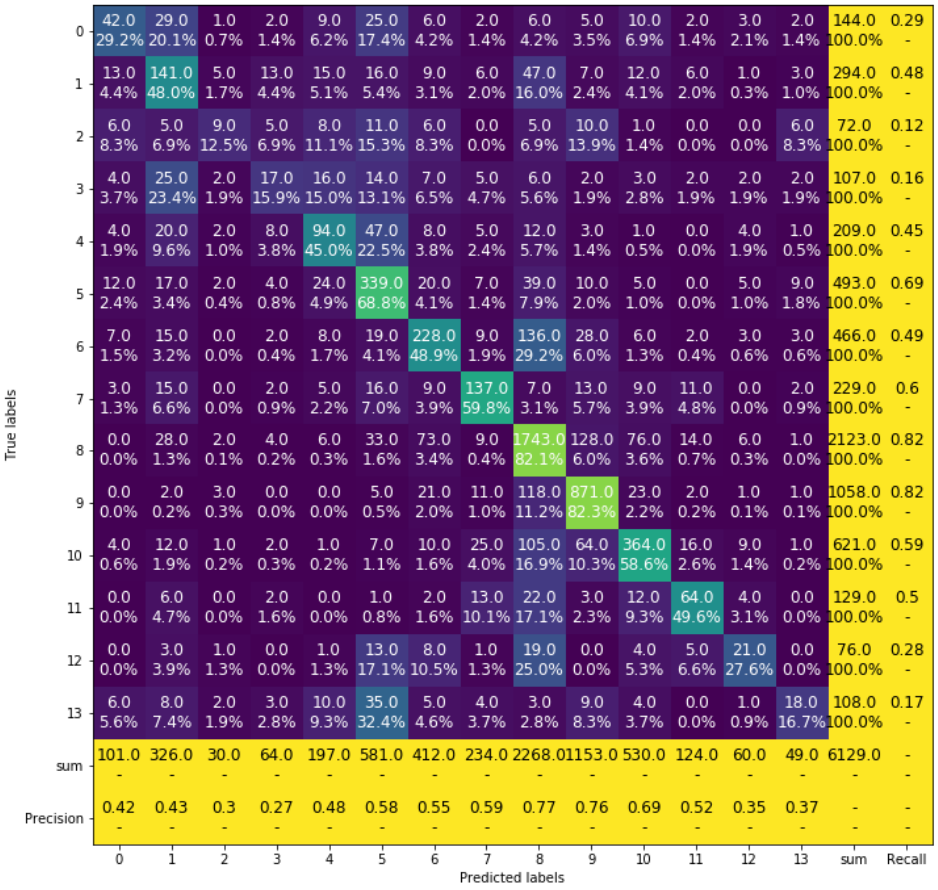

Figure 7: Normalized confusion matrix of the predictions of the full model on the 10-fold cross-validation of the training set. The IDs correspond to the following techniques (Table 1): 0=*Appeal_to_Auhority*; 1=*Appeal_to_fear_prejudice*; 2=*Bandwagon,Rectio_ad_hitlerum*; 3=*Black-and-White_Fallacy*; 4=*Causal_Oversimplification*; 5=*Doubt*; 6=*Exaggeration,Minimisation*; 7=*Flag-Waving*; 8=*Loaded_Language*; 9=*Name_Calling,Labeling*; 10=*Repetitiono*; 11=*Slogans*; 12=*Thought-terminating_Cliches*; 13=*Whataboutism,Straw_Men,Red_Herring*.

the article. We explored several approaches towards exploiting this phenomenon. Our first attempt was to include the average of the BERT logits from all propaganda fragments within an article into the concatenation that was fed to the final FFNN. I.e. if a label occurred in an article with n total labels, we included the vector $\frac{1}{n}\sum_{i=1}^{n}\mathbf{v}_i$, where $\mathbf{v}_i$ was the predicted vector of BERT logits of the i'th span in the article.

Secondly, we experimented with several different types of sequential neural networks on top of the outputs from the full model that was described in Chapter 4. This included network structures such as a bidirectional LSTM and an encoder-decoder model with attention.

The most promising model, though, was a many-to-one FFNN which used the same input as the network described in section 4.1.4. However, this model did not only use the information from the current instance of propaganda but instead had a window size of 5, meaning that it also used the information from the two prior and the two subsequent fragments of propaganda within the article. This model achieved a micro-averaged F1 score of 0.68 on a 5-fold stratified cross-validation on the training set, but it only managed a F1 score 0.54 on the official development set. This significant drop in performance from the training set to the development set

seems to be caused by the model overfitting on the training data, which may be due the increased complexity of the model.

We hypothesized that these approaches would work well as the order of the spans would be included as additional information. Unfortunately, we were unable to improve the performance using any of these approaches.

6 CONCLUSION

6.1 CONCLUSIVE REMARKS

Propaganda as a communication strategy can be used to influence the opinions of a large number of people. In order to maintain an objective public discussion, it is therefore important to catch and correct instances of propaganda usage. Until recently, this task has largely been carried out manually by individual people or organizations. However, as the tools and computational resources have become available, we have seen an emergence of research into automating this process.

The first work on automated propaganda detection focused on labeling entire articles or outlets as propagandist (Rashkin et al., 2017). However, initial research by Da San Martino et al. (2019) and their development of the corpus for fine-grained propaganda detection enabled for the first time the possibility of exploring automated propaganda technique identification.

In this thesis, we sought to explore the following research questions: RQ1: to what extent can modern natural language processing (NLP) techniques, in particular the BERT language model, be used to identify propaganda techniques within news articles? And RQ2: can traditional machine learning approaches, e.g. feature engineering and model stacking, still be used to improve the performance of BERT or are the new Transformer-based models making these approaches obsolete?

With respect to RQ1, Da San Martino et al. (2019) presented the first promising results showing that BERT can be used to identify propaganda techniques, as they successfully developed a multi-granularity network revolving around BERT for this particular task. Later studies have shown similar results using neural networks with BERT as a central component to classify propaganda techniques within news articles (Wang et al., 2020; Yoosuf and Yang, 2019).

As part of our study on how well modern NLP techniques can be used to identify propaganda techniques, we participated in the SemEval 2020 competition Task 11: "Detection of Propaganda Techniques in News Articles". With regards to RQ1, our initial results indicated that BERT proved to be a strong language model for technique classification, which inspired us to examine the effect of combining BERT with traditional machine learning techniques (RQ2). Our final proposed model for the competition consists of several components, the most important one being BERT. We combined BERT with valuable global and local features extracted from the articles and the propaganda spans. In order to measure the effect of the additional features, an ablation study was conducted. Concerning RQ2, our results and the ablation study have shown that the combination of BERT and additional features improved the predictive power of our model compared to using BERT alone.

We ended up with a micro-averaged F1 score of 0.56648 on the official test set, earning us an 11th place (out of 32 international teams) overall in the competition.

In connection with RQ1, based on our results and prior research, we can therefore conclude that the modern Transformer-based model BERT can be used as a tool for propaganda technique identification. As to RQ2, we also argue that feature-engineering and feature selection have not been rendered obsolete as a result of powerful language models such as BERT for the task of propaganda technique classification.

6.2 LIMITATIONS

As the field of NLP contains a wide range of problems, with anything from chat bots to the task of multilingual sentiment analysis, our findings on this particular task are not enough to firmly conclude that feature engineering will always improve the performance of powerful language models (such as BERT). However, we have shown that it is beneficial for propaganda technique classification, and similar results have been obtained from studies in other domains of NLP (Zhang and Li, 2019).

Also, even though we are arguing that these modern NLP techniques can be used to identify propaganda techniques within news articles to some extend, we also argue that with their current performance, the predictions of propaganda detection models, such as ours, should only be used as guidance and not as an absolute truth. Wrongly classifying news as propaganda would not only have consequences for the individual authors but also for society at large.

6.3 FUTURE WORK

We still believe that the model can be improved by including new features that contain information about the trends of the different labels. Furthermore, we would also like to exploit the tendency that a label within an article has a higher chance of occurring later in the same article. In other future work, we would like explore the effect of combining modern and traditional NLP approaches on a broader range of tasks.

BIBLIOGRAPHY

Alhindi, T., J. Pfeiffer, and S. Muresan (2019, November). Fine-tuned neural models for propaganda detection at the sentence and fragment levels. Association for Computational Linguistics.

Barrón-Cedeño, A., G. Da San Martino, I. Jaradat, and P. Nakov (2019, Jul). Proppy: A system to unmask propaganda in online news. *Proceedings of the AAAI Conference on Artificial Intelligence 33*, 9847–9848.

Bird, S., E. Klein, and E. Loper (2009). *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. Beijing: O'Reilly.

Boesche, R. (2003, 01). Kautilya's "arthaśāstra" on war and diplomacy in ancient india. *The Journal of Military History 67*, 9–37.

Da San Martino, G., A. Barrón-Cedeño, H. Wachsmuth, R. Petrov, and P. Nakov (2020, September). SemEval-2020 task 11: Detection of propaganda techniques in news articles. SemEval 2020, Barcelona, Spain.

Da San Martino, G., A. Barrón-Cedeño, and P. Nakov (2019, November). Findings of the NLP4IF-2019 shared task on fine-grained propaganda detection. Hong Kong, China. Association for Computational Linguistics.

Da San Martino, G., S. Yu, A. Barrón-Cedeño, R. Petrov, and P. Nakov (2019, November). Fine-grained analysis of propaganda in news articles. EMNLP-IJCNLP 2019, Hong Kong, China.

Devlin, J., M.-W. Chang, K. Lee, and K. Toutanova (2018). Bert: Pre-training of deep bidirectional transformers for language understanding.

Faris, R. M., H. Roberts, B. Etling, N. Bourassa, and E. Zuckerman (2017, Aug). Partisanship, propaganda, and disinformation: Online media and the 2016 u.s. presidential election.

Golovchenko, Y., A. Nielsen, and H. Meilvang (2018). Introduktion: Løgn og sandhed – fakta og fiktion. *Politik 21*(1), 1–7.

Gurevych, I., J. Eckle-Kohler, S. Hartmann, M. Matuschek, C. M. Meyer, and C. Wirth (2012, April). UBY - a large-scale unified lexical-semantic resource based on LMF. Avignon, France. Association for Computational Linguistics.

Hendrycks, D. and K. Gimpel (2016). Gaussian error linear units (gelus).

Honnibal, M. and I. Montani (2017). spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing.

Howard, J. and S. Ruder (2018). Universal language model fine-tuning for text classification.

Huang, Z., W. Xu, and K. Yu (2015). Bidirectional LSTM-CRF models for sequence tagging. *CoRR abs/1508.01991*.

Jowett, G. and V. O'Donnell (2018). *Propaganda & Persuasion*. SAGE Publications.

Kaas, A. F., V. T. Thomsen, and B. Plank (2020). Disaster at semeval-2020 task 11. https://github.com/ViktorTorp/SemEval2020-TC.

Loshchilov, I. and F. Hutter (2019, January). Decoupled weight decay regularization. Ernest N. Morial Convention Center, New Orleans.

Mikolov, T., K. Chen, G. Corrado, and J. Dean (2013). Efficient estimation of word representations in vector space.

Nalini, K. and D. L. J. Sheela (2014, Jul). Survey on text classification. *International Journal of Innovative Research in Advanced Engineering (IJIRAE) 1*.

Oxford-Languages (2016). Word of the year 2016. https://languages.oup.com/word-of-the-year/2016/.

Paszke, A., S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. 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 (Eds.), *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc.

Pennington, J., R. Socher, and C. D. Manning (2014). Glove: Global vectors for word representation.

Peters, M. E., M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer (2018). Deep contextualized word representations.

Porter, M. F. et al. (1980). An algorithm for suffix stripping. *Program 14*(3), 130–137.

Radford, A., J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever (2019). Language models are unsupervised multitask learners.

Rashkin, H., E. Choi, J. Y. Jang, S. Volkova, and Y. Choi (2017, September). Truth of varying shades: Analyzing language in fake news and political fact-checking. Copenhagen, Denmark. Association for Computational Linguistics.

Smith, B. L. (2020, Mar). Encyclopædia britannica.

Tandoc Jr, E. C., Z. W. Lim, and R. Ling (2018). Defining "fake news" a typology of scholarly definitions. *Digital journalism 6*(2), 137–153.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin (2017). Attention is all you need.

Wang, R., D. Tang, N. Duan, W. Zhong, Z. Wei, X. Huang, D. Jiang, and M. Zhou (2020). Leveraging declarative knowledge in text and first-order logic for fine-grained propaganda detection. *ArXiv 2004.14201*.

Wolf, T., L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, and J. Brew (2019). Huggingface's transformers: State-of-the-art natural language processing. *ArXiv abs/1910.03771*.

Wolpert, D. (1992, 12). Stacked generalization. *Neural Networks 5*, 241–259.

Wu, M., F. Liu, and T. Cohn (2018, October-November). Evaluating the utility of hand-crafted features in sequence labelling. Brussels, Belgium. Association for Computational Linguistics.

Wu, Y., M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, Łukasz Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. *ArXiv 1609.08144*.

Yoosuf, S. and Y. Yang (2019, November). Fine-grained propaganda detection with fine-tuned BERT. Hong Kong, China. Association for Computational Linguistics.

Yu, S., G. D. S. Martino, and P. Nakov (2019). Experiments in detecting persuasion techniques in the news. *ArXiv 1911.06815*.

Zhang, Y. and J. Li (2019). The death of feature engineering? —— bert with linguistic features on squad 2.0. Technical Report CS224n, Stanford University.