PROPAGANDA TEXT CLASSIFICATON ANALYSIS NEWS BASED ON THEIR PROPAGANDISTIC CONTENTS

By

Faheem Mohammed Abdul, Adv. Diploma – Data Science & Applications,
Metro College of Technology – Don Mills - Toronto, 2020.

A Major Research Project

Presented to Ryerson University

In partial fulfillment of the requirements for the degree of

Master of Science

in the Program of

Data Science and Analytics

Toronto, Ontario, Canada, 2022

AUTHOR'S DECLARATION FOR ELECTRONIC SUBMISSION OF A MAJOR RESEARCH PROJECT (MRP)

I hereby declare that I am the sole author of this Major Research Paper. This is a true copy of the MRP, including any required final revisions.

I authorize Ryerson University to lend this MRP to other institutions or individuals for the purpose of scholarly research.

I further authorize Ryerson University to reproduce this MRP by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

I understand that my MRP may be made electronically available to the public.

Faheem Mohammed Abdul

PROPAGANDA TEXT CLASSIFICATON ANALYSIS NEWS BASED ON THEIR PROPAGANDISTIC CONTENTS

Faheem Mohammed Abdul

Master of Science 2022

Data Science and Analytics

Ryerson University

ABSTRACT

"Propaganda is a mechanism to influence public opinion, which is inherently present in extremely biased and fake news." If a Celebrity/famous personality has given his/her personal opinion about any upcoming or occurred event organized by a government which is disliked by political parties, organizations, or any individuals then those political parties or individuals can come up with an extreme bias by manipulating original opinion. Here, I propose a model to automatically assess the level of propagandistic content in an article based on different representations, from writing style and certain keywords. I experiment thoroughly with different variations of such a model on a new publicly available corpus, and I show that character n-grams and other style features outperform existing alternatives to identify propaganda based on word n-grams. I make sure that the test data comes from news sources that were unseen on training, thus penalizing learning algorithms that model the news sources used at training time as opposed to solving the actual task. This allows users to quickly explore different perspectives of the same story, and it enables investigative journalists to dig further into how different media use stories and propaganda to pursue their agenda.

Key words:

Propaganda Detection, Text Classification, Linear SVM, Bias News, Social Media, Investigative Journalism. Content, Data Classification, Machine Learning.

ACKNOWLEDGEMENTS

I am very grateful to **Professor Dr. Farid Shirazi** and his team at Ryerson University Social Media Lab as well as virtual for their support and assistance in making this project a reality. Professor Farid was my supervisor for this MRP; he has been a great support throughout the term to guide and direct my research and provide valuable feedback. Social Media Lab played a vital role in labelling the dataset manually with classification codes that I used in this project to train my classifiers. It was a manual process that required time and effort and I appreciate the Social Media Lab team completing that task in time for me to use the dataset for this project.

Thank you, Professor Dr. Farid Shirazi.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Introduction

The landscape of news outlets is wide: from supposedly neutral to clearly biased. When reading a news article, every reader should be aware that, at least to some extent, it inevitably reflects the bias of both the author and the news outlet where the article is published. However, it is difficult to identify exactly what the bias is. It could be that the author himself may not be conscious about his own bias. On the other hand, it could be that the article is part of the author's agenda to persuade readers about something on a specific topic. The latter situation represents propaganda. According to the now classical work from the Institute for Propaganda Analysis, propaganda can be defined as follows:

Definition 1: *Propaganda is expression of opinion or action by individuals or groups deliberately designed to influence opinions or actions of other individuals or groups with reference to predetermined ends.*

Propaganda is most effective when it can go unnoticed. That is, if a person reads a journalistic text, in a formal or an informal news outlet he should not be able to identify it as propagandistic. In that case, the reader is exposed to the propagandistic content without his knowledge and some of his opinions might change as a result. A striking example of the use of propaganda was allegedly put in place to influence the 2016 US Presidential elections. Given the wide landscape of news outlets —from tabloids to broadsheets, from printed to digital, from objective to biased we believe that both news consumers and institutions might benefit from an automatic tool that can detect propagandistic articles.

Here we propose proppy, a system to organize news events according to the level of propagandistic contents in the articles covering them. Proppy is a full architecture that takes a batch of news articles as input, identifies the covered events, and organizes each event according to the level of propaganda in each article. Our major contribution, and the focus of this manuscript, is a supervised model to compute what we refer to as propaganda score: the estimated likelihood of a text document to contain propagandistic mechanisms to deliberately influence the reader's opinion.

1.1 Background

The term propaganda was coined in the 17th century, meaning propagation of the Catholic faith. The term soon took a pejorative connotation, as it was not only intended to spread the faith in the New World, but also to oppose Protestantism; i.e. it was not neutral. Here, we are interested in a journalistic point of view of propaganda: how news management lacking neutrality shapes information by emphasizing positive or negative aspects purposefully. As Jowett and O'Donnell mention, propaganda is frequently considered a synonym of lies, distortion, and deceit. Indeed, all biased messages have been identified as propagandistic, regardless of whether the bias was conscious or not. As a result, if a model is capable of identifying propaganda in a piece of news, it enhances a reader's awareness that she might be facing a biased text. Bias must be considered when addressing people's information needs, as it affects us all and much of the time we are unaware of it.

1.2 Research Question

The MRP would be on Propaganda News/Articles the main objective of this project is to find the Article is a Propaganda Article or not. Propaganda text classification using various machine learning and deep learning models. I can use pertained embedding along with transfer learning techniques to extract information from the text. In addition, model can be deployed as a URL service so that people can check articles to classify whether it is propaganda or not.

1.3 Objective

I will endeavor to incorporate other model(s) as well. My inspiration and reason for Propaganda text classification was propaganda is commonly found in every day news articles and columns. It is dangerous to be ignorant of the propagandized scripts, as they tend to shape information to foster predetermined agendas. So I thought it would be the perfect time to do an analysis articles to classify whether it is propaganda or not. Therefore, I want to propose a model to automatically assess the level of propagandistic content in an article based on the different representations, from writing style and readability level to the presence of certain keywords and lexicon. This would allow users to quickly explore different perspectives on the same story, and it enables investigative journalists to dig further into how different media use stories and propaganda to pursue their agenda.

2. Literature Review

Here I propose proppy, a system to organize news events according to the level of propagandistic contents in the articles covering them. Proppy is a full architecture that takes a batch of news articles as input, identifies the covered events, and organizes each event according to the level of propaganda in each article. The major contribution, and the focus of this manuscript, is a supervised model to compute what I refer to as propaganda score: the estimated likelihood of a text document to contain propagandistic mechanisms to deliberately influence the reader's opinion.

Proppy computes a propaganda score using a maximum entropy classifier. I chose this classifier in order to facilitate direct comparison to previous work and to focus I efforts on improving the representation of the data in terms of features. In, word n-grams were used but, as the authors themselves pointed out, this yielded significant drop in performance when testing on articles from sources that were not seen on training. Here I aim to shed some light about why this could be the case. Therefore, we formulate the following hypothesis:

Hypothesis 1 (H1). Representations based on writing style and readability can generalize better than currently used approaches based on word-level representations.

I argue that this is because word-level representations tend to learn topic and source, rather than whether the target article is propagandistic or not. In order to test the above hypothesis, I first replicated a pre-existing model for propaganda detection. Later on, I compiled a new corpus — QProp— which, unlike most pre-existing corpora, keeps explicit information about the source of each article, thus allowing me to train on articles from some sources and to test on articles from different sources that have not been used for training. I design experiments that involve training and evaluating several supervised models using features based on text readability and style; such features have been widely used in authorship attribution tasks. In my thorough experimentation, i obtain statistically significant improvements over existing approaches in terms of classification performance, especially when testing on articles from unseen sources.

My contributions can be summarized as follows:

1. I experiment with different families of feature representations spanning readability, vocabulary richness, and style in an effective propaganda estimation model, and I demonstrate empirically that they are effective for actually detecting propaganda, as opposed to learning the article's source or its topic as it is the case in most previous work.
2. Allows users to explore the coverage of the current news events based on their propagandistic content.

The remainder of this article is organized as follows:

- Offers a soft introduction to propaganda.
- Related work on (automatic) propaganda identification and authorship-derived representations. Introduces our propaganda detection model.
- The datasets we experiment with, including our new dataset.
- Covers our experiments and discusses the results.
- Describes the full architecture of proppy —as running on the Web—, which includes retrieving the articles, grouping them into events, computing their propaganda score, and displaying the results.
- Finally, concludes and points to possible directions for future work.

Recently, there has been a lot of interest in studying disinformation and bias in the news and in social media. This includes challenging the truthiness of news (Brill, 2001; Finberg, Stone, & Lynch, 2002; Hardalov, Koychev, & Nakov, 2016; Potthast, Kiesel, Reinartz, Bevendorff, & Stein, 2018), of news sources (Baly, Karadzhov, Alexandrov, Glass, & Nakov, 2018), and of social media posts (Canini, Suh, & Pirolli, 2011; Castillo, Mendoza, & Poblete, 2011; Zubiaga, Liakata, Procter, Wong Sak Hoi, & Tolmie, 2016), as well as studying credibility, influence, and bias (Ba, Berti-Equille, Shah, & Hammady, 2016; Baly et al., 2018; Chen, Wu, Srinivasan, & Zhang, 2013; Kulkarni, Ye, Skiena, & Wang, 2018; Mihaylov, Georgiev, & Nakov, 2015; Mihaylov et al., 2018). The interested reader can also check several recent surveys that offer a general overview on "fake news" (Lazer et al., 2018), or focus on topics such as the process of proliferation of true and false news online (Vosoughi, Roy, & Aral, 2018), on fact-checking (Thorne & Vlachos, 2018), on data mining (Shu, Sliva, Wang, Tang, & Liu, 2017), or on truth discovery in general (Li et al., 2016). For some specific topics, research was facilitated by specialized shared tasks such as the SemEval-2017 task 8 on Rumor Detection (Derczynski et al., 2017), the CLEF- 2018 lab on Automatic Identification and Verification of Claims in Political Debates (Nakov et al., 2018), the FEVER-2018 task on Fact Extraction and VERification (Thorne, Vlachos, Christodoulopoulos, & Mittal, 2018), and the SemEval-2019 Task 8 on Fact Checking in Community Question Answering Forums (Mihaylova et al., 2019), among others.

From a modeling perspective, most approaches relied on stylistic and complexity representations, which tend to be topic- and genre-independent. That is, regardless of the event being covered in the target news article or the direction of its bias (if any), the features need to contain the necessary information for the model to be able to make a decision. This is precisely the main design principle of the representations used in authorship attribution —the task of verifying whether a dubious text has been written by the same known author who is behind a number of other texts (Juola, 2012). While factors such as topic and text length play little role for this task, among the most successful representations we typically find character-level n-grams (Stamatatos, 2009). As Hypothesis 1 states, we believe that these representations are robust and are also useful for modeling the degree of bias and propaganda in news articles.

3. Exploratory Data Analysis – EDA

The goal of this research is to build a machine-learning model, as well as preliminary data process and feature extraction algorithms that would allow to successfully identify signs of propaganda in text data and to solve a binary classification task. The task is presented in two forms: article level propaganda detection and sentence level propaganda detection. Each article is marked as either "propaganda" or "non-propaganda".

The dataset also contains unique identifier for each article. Before we start feature extraction process, I need to perform a few particular operations on the data to clean and prepare it for the extraction. First, I need to convert every word in the dataset to lowercase so that in the process of vectorization two semantically identical words, one uppercase and one lowercase, would not considered as separate tokens. Data is presented in the form of text file that consist of tab-separated article content, assigned class and unique article identifier.

3.1 Data Acquisition

Data for this project is acquired from https://zenodo.org/record/3271522#.Yu8DbnbMJPZ. The datasets are open-sourced and compliant with the MRP requirements and have been collected.

3.2 Target and Independent Variables

The corpus contains 52k articles from 100+ news outlets. Each article is labeled as either "propagandistic" (positive class) or "non-propagandistic" (negative class). The labeling was done indirectly using a technique known as distant supervision, i.e. an article is considered propagandistic if it comes from a news outlet that has been labeled as propagandistic by human annotators.

3.3 Data format

We provide the corpus in three tsv files, including training, development, and testing partitions.

The data is tab-separated. Each line represents one article, with the following information:

1. article_text : the text of the article retrieved via newspaper3k package.
2. event_location : the geographical location - collected from GDELT.
3. average_tone : measures the impact of the event - collected from GDELT

4. article_date : article's publish date - collected from GDELT.
5. article_ID : GDELT ID , unique among the dataset's articles.
6. article_URL : the direct URL for the published article in its source website.
7. MBFC_factuality_label: factuality label for the source from MBFC
8. article_URL
9. MBFC_factuality_label
10. URL_to_MBFC_page
11. source_name
12. MBFC_notes_about_source
13. MBFC_bias_label
14. source_URL
15. propaganda_label

3.4 Data Info.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 35986 entries, 0 to 35985
Data columns (total 15 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   text                    35986 non-null  object
 1   location                35986 non-null  object
 2   tone                    35986 non-null  float64
 3   date                    35986 non-null  object
 4   ID                      35986 non-null  int64
 5   URL                     35986 non-null  object
 6   MBFC_factuality_label   26902 non-null  object
 7   URL.1                   35986 non-null  object
 8   MBFC_factuality_label.1 35986 non-null  object
 9   URL_to_MBFC_page        35986 non-null  object
 10  source_name             35986 non-null  object
 11  MBFC_notes_about_source 29979 non-null  object
 12  MBFC_bias_label         35986 non-null  object
 13  source_URL              35986 non-null  object
 14  propaganda_label        35986 non-null  int64
dtypes: float64(1), int64(2), object(12)
memory usage: 4.1+ MB
```

3.5 Optimal Column Selection.

- The above training dataset has total of 15 columns/features.
- The most important features out of 15 are the input text ("text"), headline ("URL.1"), and the output label (propaganda_label).
- We I am removing the rest columns and keeping the above mention one.

3.6 Data Cleaning.

3.6.1 Handling Missing Values.

```
Empty DataFrame
Columns: [missing_count, missing_percentage]
Index: []
Empty DataFrame
Columns: [missing_count, missing_percentage]
Index: []
Empty DataFrame
Columns: [missing_count, missing_percentage]
Index: []
```

- ❖ **None of the column/feature has missing value present in the dataset.**

3.6.2 Duplicate Row Analysis.

```
Empty DataFrame
Columns: [text, URL.1, propaganda_label]
Index: []
Empty DataFrame
Columns: [text, URL.1, propaganda_label]
Index: []
Empty DataFrame
Columns: [text, URL.1, propaganda_label]
Index: []
```

- ❖ **There are no duplicate values present in the dataset.**

3.7 Target Label Analysis.

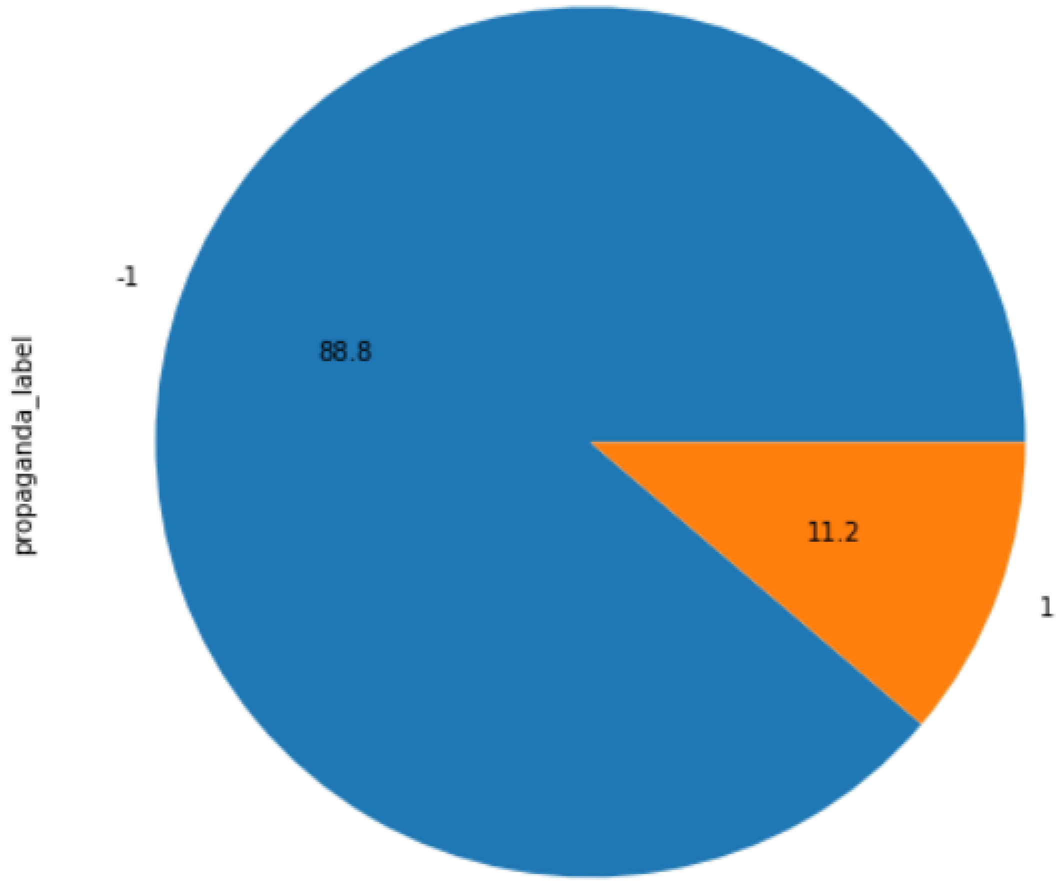


Figure 1: Target Label Analysis

Observations:

- "-1" represents no propaganda, and "1" represents yes propaganda.
- The class is highly imbalance as only 11.2% of text consists of propaganda yes.

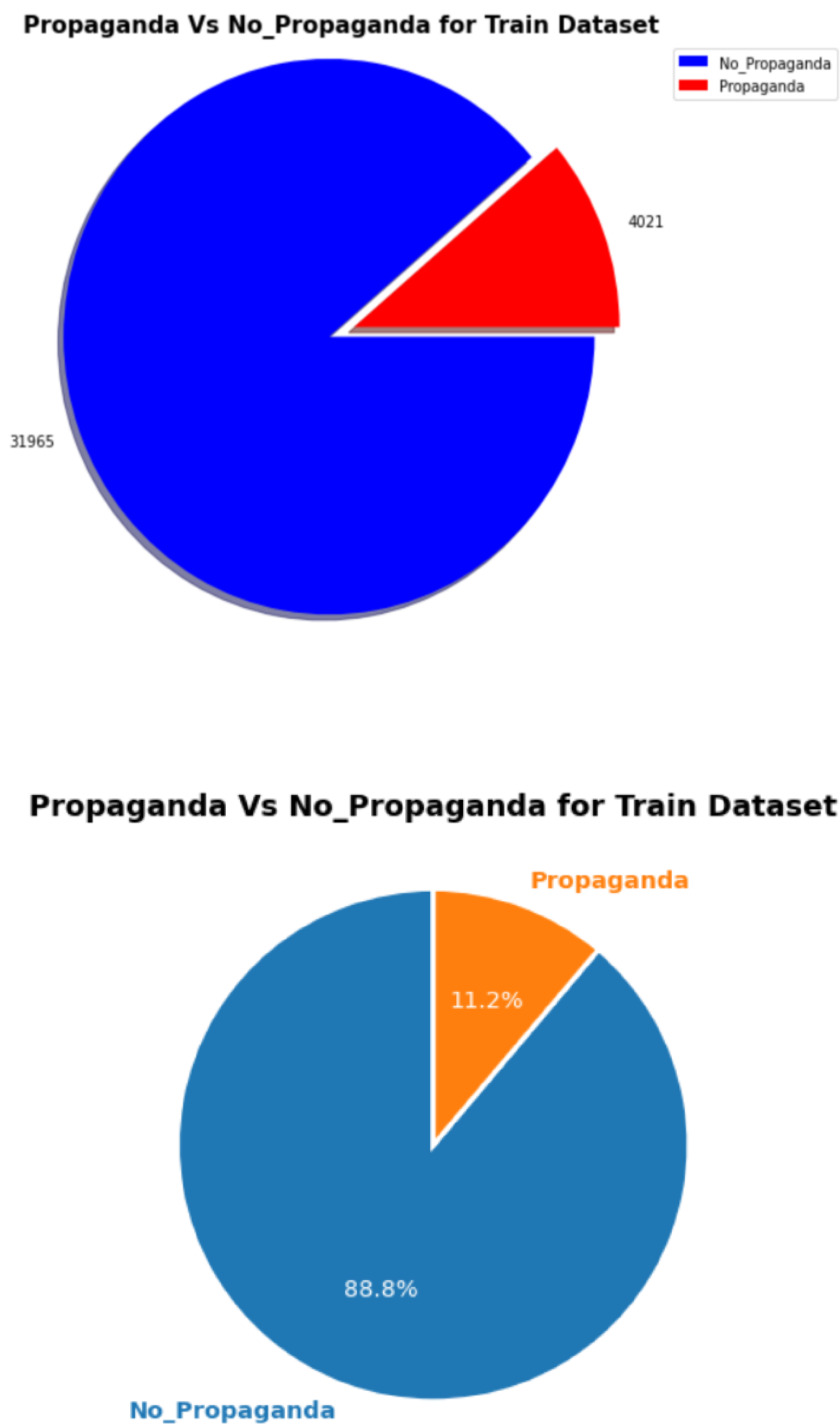


Figure 2: Target Label Analysis of Train Dataset

3.8 Text Preprocessing

'Editorial: Why, Rhode Island, Why? Et tu, Rhody? A recent editorial in the Providence Journal cataloged everything it could find wrong with Connecticut and ended with this suggestion: "Gov. Gi
na Raimondo should see if at least some of those jobs could come to Rhode Island. It is certainly less risky than the Nutmeg State." We beg your pardon. The state with world-famous pension pro
blems and persistent economic issues of its own is "less risky"? The Journal itself reported just a few weeks ago on Rhode Island's own significant economic problems, which in many ways reflect
Connecticut's. Rhode Island enjoys a legacy of corruption that not even Connecticut can match. The ProJo won a Pulitzer Prize in 1994 for uncovering widespread corruption within its own court s
ystem. What, exactly, is to be gained from moving to Rhode Island? Like Connecticut, Rhode Island has an income tax and an estate tax with comparable rates. (Forbes magazine listed it as one o
f the states "Where Not To Die." Connecticut made the list, too.) Connecticut and Rhode Island's interdependence has been limited, with the exception of the interstate economy created by Electr
ic Boat in Groton. There have been no border wars and very little bloodshed. A few jokes about Rhode Island's size, maybe, but if we're being honest, Connecticut doesn't really have a lot going
on in that department either. A little interstate competition is fine, but if Connecticut suffers, so does Rhode Island – and all of New England, for that matter. Connecticut is losing residen
ts at a troubling rate, but Rhode Island has an outmigration problem of its own. From 2015 to 2016, the Ocean State experienced a net loss of about 2,000 tax filers, who took with them more than
$182 million in adjusted gross income. The top destination states for people who fled Rhode Island were Massachusetts, Florida and – wait for it – Connecticut. Connecticut residents moved to Rh
ode Island as well, of course. But Connecticut's population is 3½ times as big as Rhode Island's. So the 1,175 tax filers who left Rhode Island for Connecticut represent a far larger portion of
the Ocean State than the 1,220 who moved from Connecticut to Rhode Island. If any state should be concerned about losing residents to its neighbor, it's Rhode Island. But we don't want to poach
Rhode Islanders. We'd rather celebrate Electric Boat's growth and the burgeoning workforce that supports both states. We'd rather cheer CVS for buying Aetna and keeping it in Hartford than try t
o woo CVS from Woonsocket. A booming Connecticut, especially in the insurance and defense industries, only helps Rhode Island. As Electric Boat – headquartered in Connecticut, might we emphasi
ze – grows over the next decade, the effect on Little Rhody will be profound, as the ProJo's editorial board pointed out. A thriving border economy helps both states as supplier chains develop a
nd as feeder businesses bloom. But for the same reasons that the stain of a Hartford bankruptcy would spread to the suburbs, if Connecticut becomes an economic wasteland, the effects would be f
elt across New England. If Rhode Island and Connecticut want to find a way out of the muck, far better for them to work together. Yes, Connecticut can learn from Rhode Island. Connecticut's pe
nsion problems are similar to those that threatened to swamp Rhode Island, but there are key differences, especially in that Connecticut's pensions are contractual, where in Rhode Island, they w
ere set by state statute. Rhode Island made some tough choices and anticipated a legal battle to solve its problems. Connecticut leaders might have to find the stomach for the same type of stra
tegy. Connecticut and Rhode Island have a lot in common, including language. We both drive around the rotary to get a grinder at Cumbie's, for example. And we are glad that Rhode Island has mad
e progress on its pension issues. But that's no reason to try to poach a few residents. A regional approach would be much wiser.'

Figure 3: Text Analysis

Observations:

- From the above text, we can see that the text consists of stop words, punctuations, special characters, numbers, and combination of alphanumeric values.
- We must perform text-preprocessing methodology to remove the unwanted values and characters.

3.9 Word Cloud

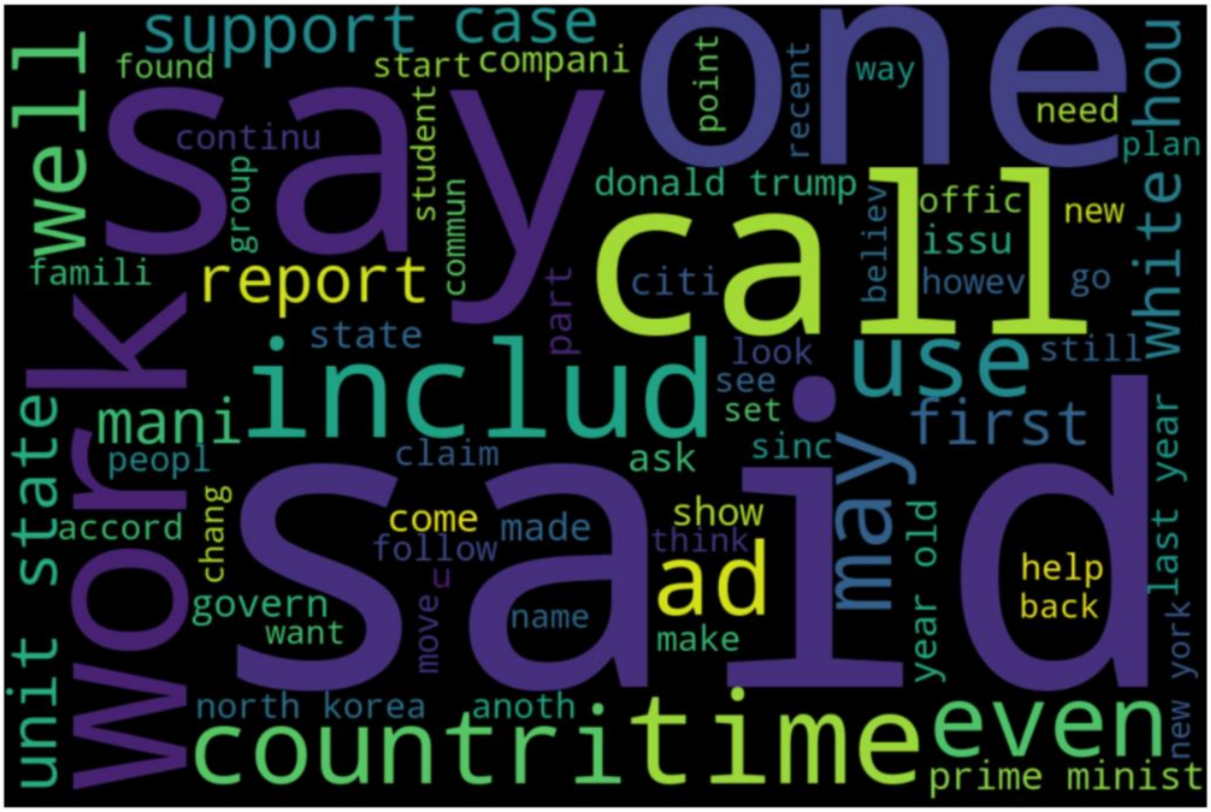


Figure 4: Word Cloud

The word cloud above lists all words with minimum frequency of 200. Word clouds are useful in understanding, which somewhat mostly used in Propagandistic and Non-Propagandistic Articles. So, based on this word cloud I can clearly state that word "Said" was used most followed by "Say" and "One".

3.10 n-gram Analysis

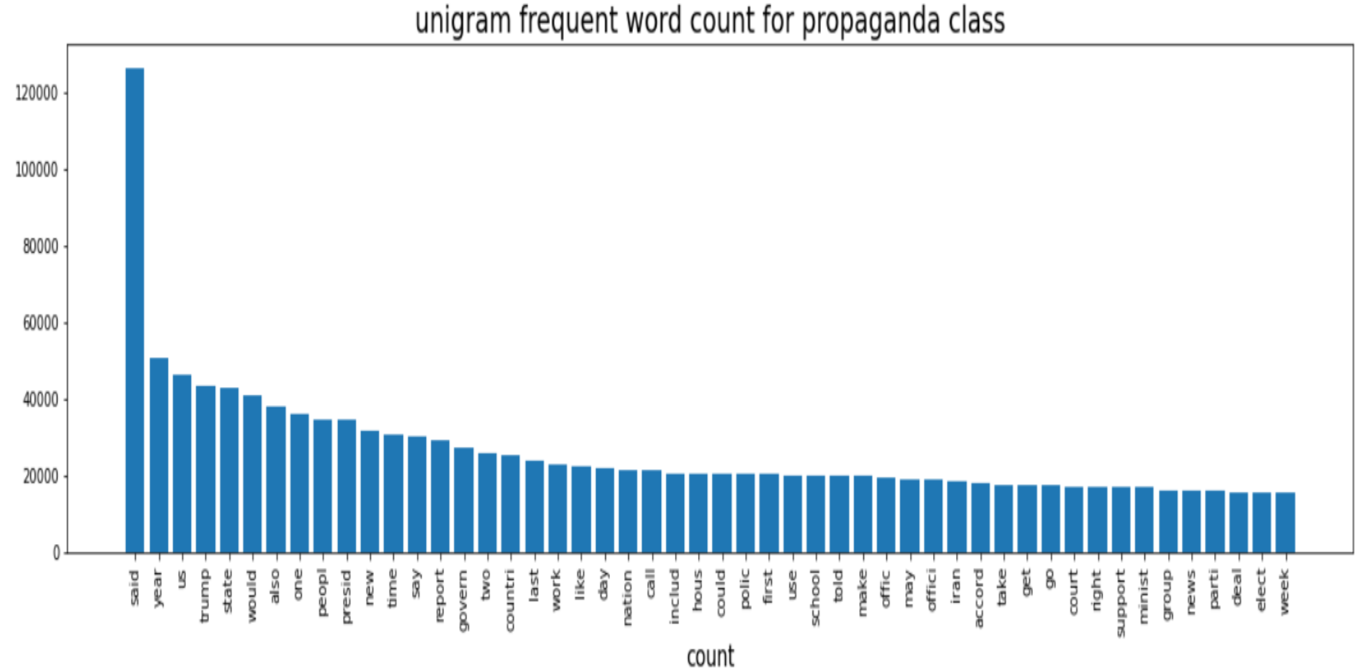


Figure 5: Top 50 of Propaganda Article.

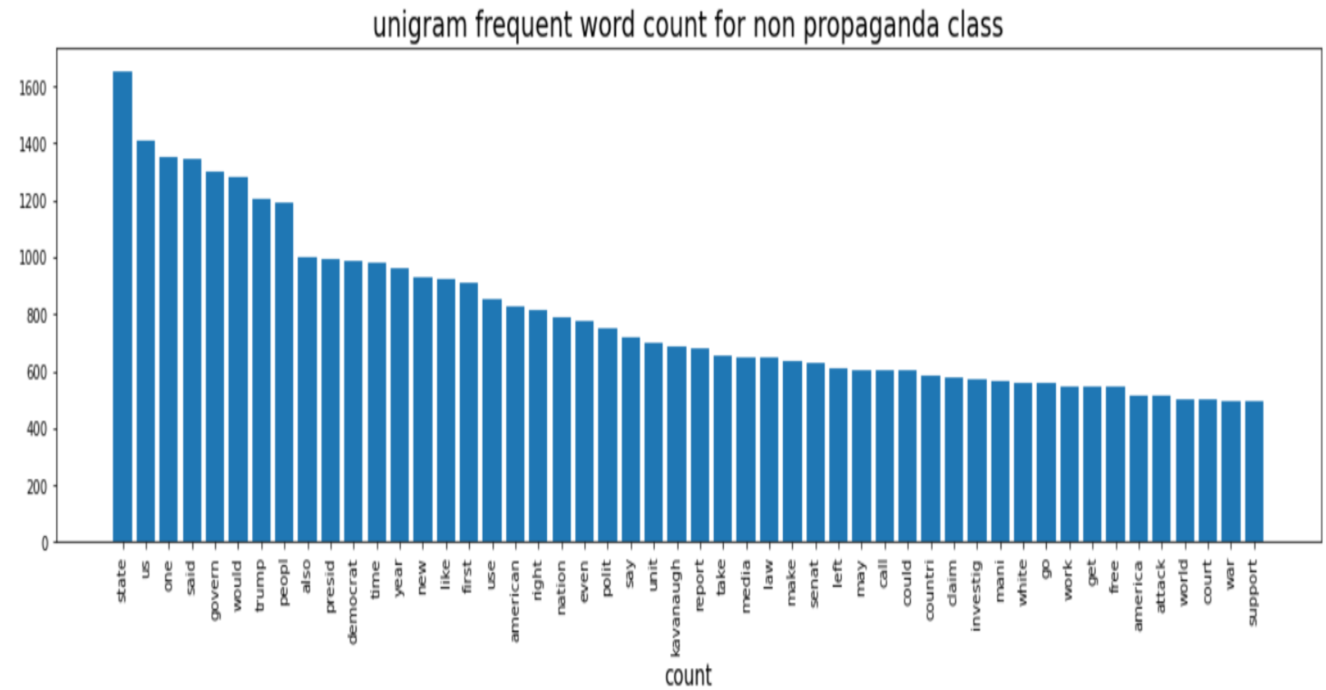


Figure 6: Top 50 used in Non-Propaganda Article.

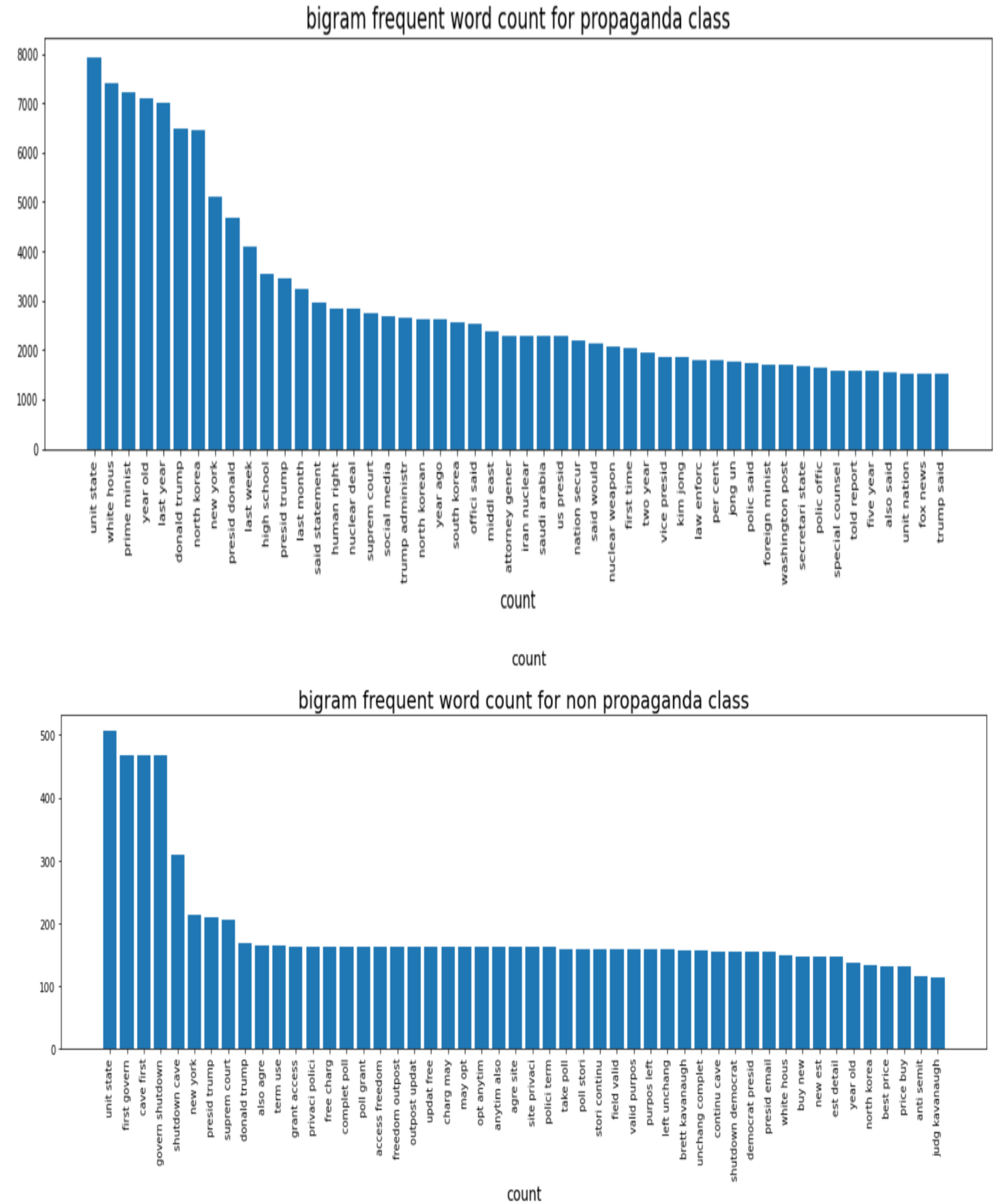
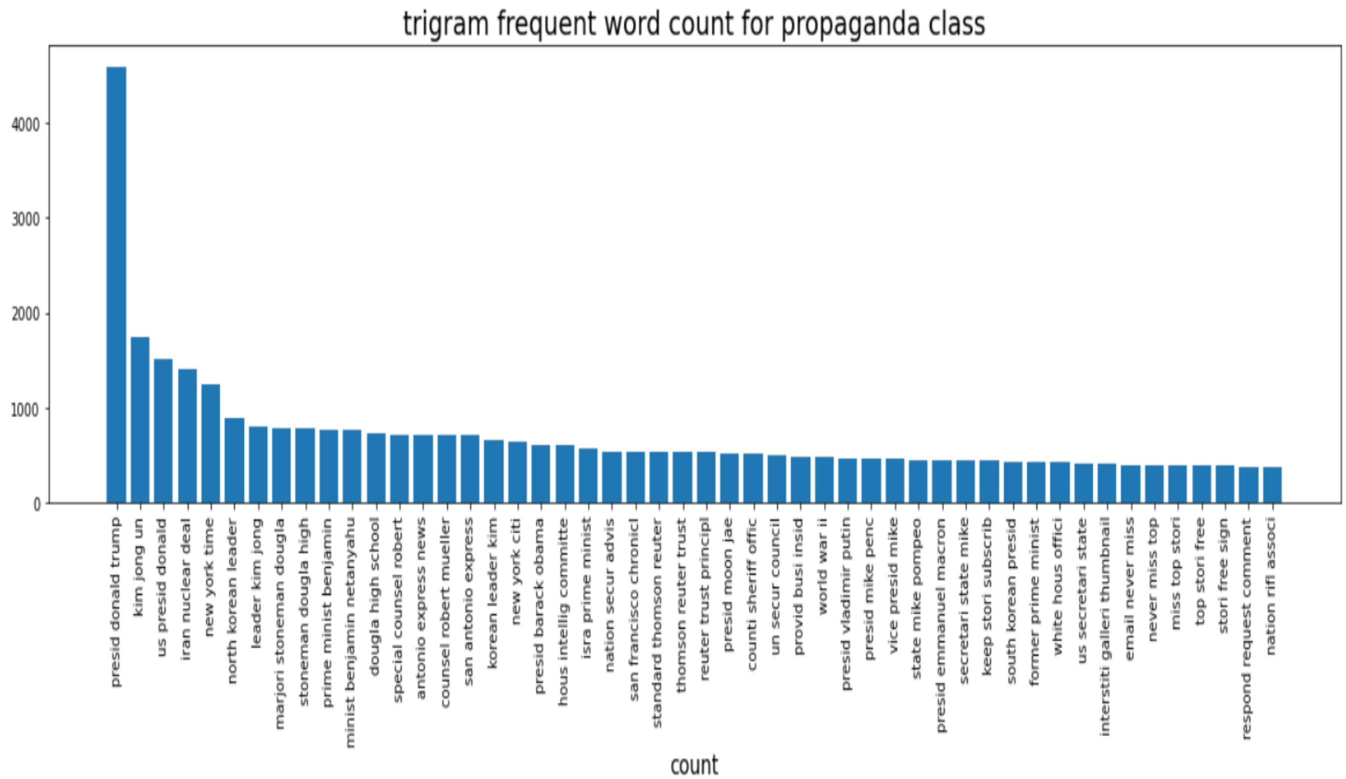


Figure 7: Top 50 word bi-gram in Propaganda and Non-Propaganda Articles.



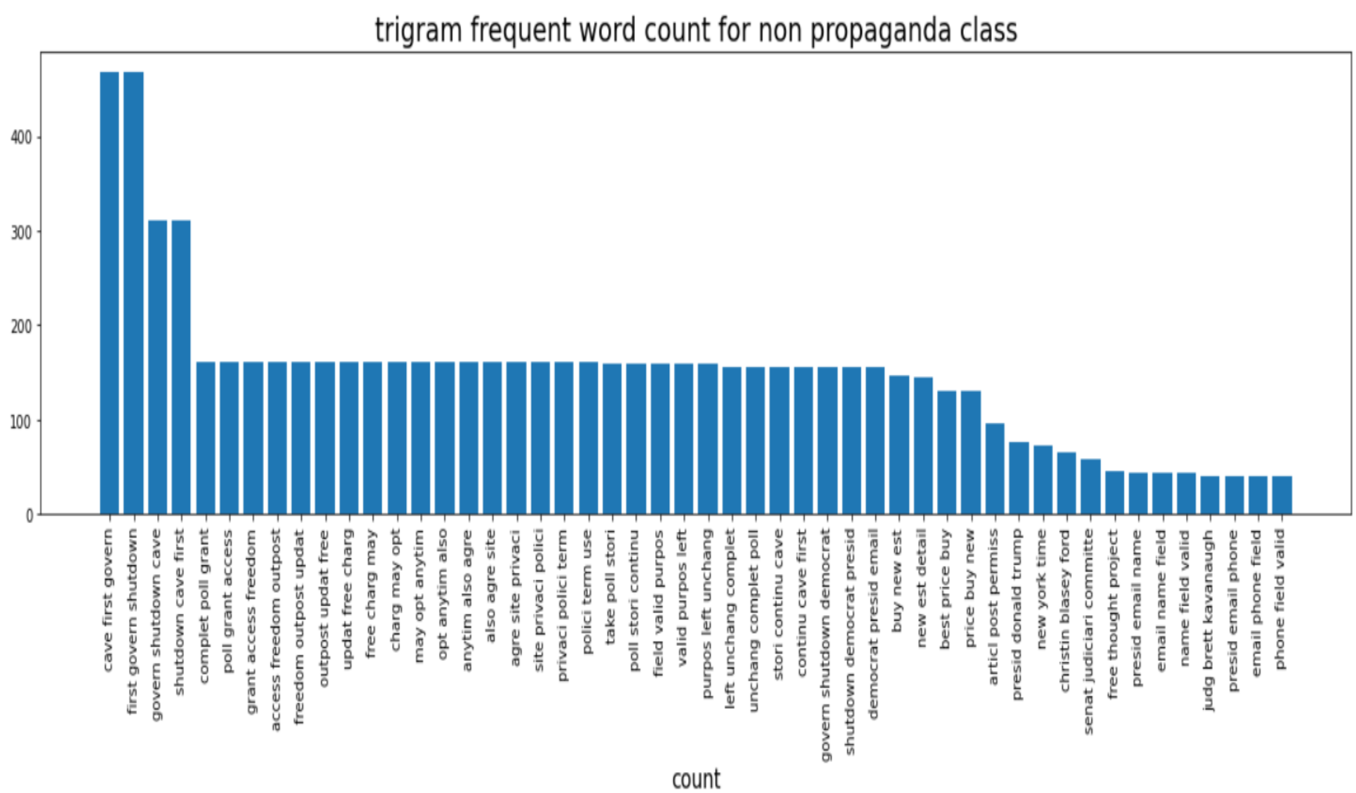


Figure 8: Top 50 word tri-gram in Propaganda and Non-Propaganda Articles.

Methodology and Experiments

Propaganda are generally biased information or knowledge, which are often used for promotions, advertising and in politics.

Pro and Cons of a Propaganda:

- Propagandas can be good when they create or lead the public to an image, which depicts a positive side and where the results are actually positive. They are good when the essence or the gist does not cause conflicts disturbances in a social setting
- Propagandas are good when they do not intend to hinder with the ethical values of individuals.
- If these propagandas direct to a negative idea or promotes misconceptions. When the real core is masked for material benefits that drives people to invest their resources and they promotes confusion

If a propaganda has a positive social goal, then the propaganda is acceptable. In this case the techniques becomes acceptable and ethical only when correspond to a good social connection and sticks to morals and true spirit of the idea.

4. Methodology

We use a maximum entropy classifier with L2 regularization and default parameters to discriminate propagandistic from non-propagandistic articles. We chose it in order to facilitate direct comparison with the work. We consider four families of features, which we describe below.

4.1 Word n-Gram Features

We use tf-idf-weighted word [1; 3]-grams as baseline features, after tokenizing the Text with NLTK. They were used to discriminate trusted vs. propaganda vs. Hoax vs. satire articles.

Table 1

Source	Lexicon(Example Entry)
Wiktionary	Modal(truly) ● **Action** (accidentally) ● **Manner Adverbs** (foolishly) ●**Comparative**(higher) ● **Superlative Forms** (worst)
LIWC	**First Person Singular**(my) ●**Second Person** (you) ●**Hear** (says) ●**Money** (costs) ●**Negation** (can't) ● **Number** (quarters) ● **See** (Watch) ●**Sexual** (gay) ●**Swear**(dumb)
Wilson et al.	**Strong subjectives** (anti-semites) ● **Weak Subjectives** (extremist)
Hyland	**Hedges** (perhaps)
Hooper	**Assertives** (certain

Table 1: Lexicon sources and lexicon we use for feature extraction with example entries.

4.2 Lexicon Features

Certain kind of vocabulary is common for specific propagandistic techniques (e.g., in name-calling and glittering generalities). We try to capture this by considering representations reflecting the frequency of specific words from a number of lexicons, shown in Table 1. They come from the Wiktionary, the Linguistic Inquiry and Word Count (LIWC) lexicon, Wilson's subjective, Hyland hedges, and Hooper's assertives. For each of the 18 lexicons, we count the total number of occurrences of the words from this lexicon in the text.

The relationship between the occurrences of the words from the above lexicons in different kinds of news articles. They found that the words from some of their lexicons (e.g., swear, see, negation) appear more frequently in propagandistic, satire, and hoax articles than in trustworthy news articles.

Feature	Computation
TTR. Type–token ratio.	$\lvert types \rvert / \lvert tokens \rvert$
Hapax legomena. Word types appearing once in a text.	$\lvert types_1 \rvert$
Hapax dislegomena. Word types appearing twice in a text	$\lvert types_2 \rvert$
Honore's R. Word types, tokens, and hapax legomenæ.	$\frac{100 \cdot log(\lvert tokens \rvert)}{1 - \lvert hapax\_legomena \rvert / \lvert types \rvert}$
Yule's characteristic K. Combination of types appearing with different frequencies and tokens. Assumes that the occurrences of a word follow a Poisson distribution. Here $i = [1, 2, \ldots]$ is the number of word types with a frequency of i in the text.	$10^4 \frac{\sum_i i^2 \lvert types_i \rvert - \lvert tokens \rvert}{\lvert tokens \rvert^2}$

Table 2: Vocabulary Richness Features

4.3. Vocabulary Richness, Readability, and Style

The hyperpartisan outlets tend to use writing style that is different from that of mainstream news media. Thus, we also use features that model style. Different topic-independent features have been proposed in the literature to characterize the vocabulary richness, style, and complexity of a text. Whereas many such features were originally intended to assess the pertinence of teaching materials for different education levels, they have been also found useful for authorship attribution and related tasks. Table 2 shows the five features we use in order to model the vocabulary richness of a news article. We consider the type-token ratio (TTR) as well as the number of types appearing exactly once or exactly twice in the document: the hapax legomena and dislegomena, respectively. We further combine word types, word tokens, and hapax legomena to compute characteristic K.

Table 3 shows the three readability features: the Flesch–Kincaid grade level, the Flesch reading ease, and the Gunning fog index.

Feature	Computation
Flesch–Kincaid grade level. US grade level necessary to understand the text.	$0.39 \cdot \frac{\lvert tokens \rvert}{\lvert syllables \rvert} + 11.9 \cdot \frac{\lvert syllables \rvert}{\lvert tokens \rvert} - 15.59$
Flesch reading ease. A scale in the range $[0, 100]$ representing the complexity of a text. Higher score means easier text.	$206.835 - 1.015 \cdot \frac{\lvert tokens \rvert}{\lvert sentences \rvert} - 84.6 \cdot \frac{\lvert syllables \rvert}{\lvert tokens \rvert}$
Gunning fog index. Number of years of formal education necessary to understand the text. Here, $tokens_c$ stands for complex tokens: those with three syllables or more.	$0.4 \left(\frac{\lvert tokens \rvert}{\lvert sentences \rvert} + 100 \cdot \frac{\lvert tokens_c \rvert}{\lvert tokens \rvert} \right)$

Table 3: Readability Features

We can argues that in tasks in which the topic is not relevant, character-level representations are more sensitive than token-level ones. So, we considers that "the most frequent character n-grams are the most important features for stylistic purposes". Our style representation consists of tf -idf-weighted character 3-grams. These representations capture different style markers, such as prefixes, suffixes, and punctuation marks.

4.4. NELA

Recently, The NEws LAndscape features (NELA): 130 content-based features collected from the literature that measure different aspects of a news article such as sentiment, bias, morality, and complexity, among others. We integrated the NELA features into our model and experiments. They are categorized in six subgroups, which are included in Table 4 (a seventh subgroup Facebook engagement reported was not included in their software release).

NELA Features Table

Subgroup	Description
Structure	Part of Speech
Sentiment	Emotions: positive, negative, affect, etc. from LIWC ●happiness score
Topic specific	Biological process ●relativity: motion, time, and space words ● personal concerns: work, home, leisure etc. (all from LIWC)
Complexity	SMOG readability measure ●average word length ●word count ●cognitive process words from LIWC
Bias	Several bias lexicons ●subjectivity probability in the text
Morality	Features based on the Moral Foundation Theory

Table 4: NELA Features

4.5 Experiments and Evaluation

We designed three experiments to verify hypothesis H1. The first one aims at comparingour features, and thus we experimented with a 4-way classifier: trusted vs. propaganda vs. hoax vs. satire. The second experiment focuses on our main 2-way classification task: propaganda vs. non-propaganda. We perform this experiment on both the TSHP-17 and the QProp corpora. As we observe a sizable drop in performance when testing on news coming from sources never seen during training, we further run a third experiment to test whether this is due to representations misleading the algorithm to model the media source instead of solving the actual task.

We replicate the experimental setup by using a Maximum Entropy classifier with L2 regularization and default parameters (C=1). This allows us to compare to them directly, and to focus on the effectiveness of the different representations: word n-grams, lexicon, vocabulary richness, readability, and character n-grams. Note that, since we fixed the hyper-parameters of the algorithm, there is no need for a separate tuning dataset. We also tried using support vector machines. The results with the linear kernel varied slightly with respect to the Maximum Entropy classifier and they were much worse when using the polynomial and RBF kernels. Thus, we decided to report results for the Maximum Entropy only.

We used two basic evaluation measures: F1-measure and accuracy. For the multiclass setting in experiment 1, we report macro-averaged F1, while for the binary setting in experiments 2 and 3, we take propaganda as the positive class and we compute F1 with respect to that class (no macro-averaging). In order to better analyze the results, we used the McNemar statistical test. This is a non-parametric test that computes statistics based on the comparison between the number of instances in which the predictions of two classifiers differ. Such statistics approximate a $\chi 2$ distribution, assuming that the number of instances in which the two predictors differ is greater than 20, a condition which we checked was always satisfied in our experiments. We selected the standard value of $\alpha =$ 0:05. Therefore, whenever we use the term statistically significant, we refer to McNemar's test at 95% confidence level.

4.5.1. Experiment 1: Four-Way Classification on the TSHP-17 Corpus

Whereas identifying propagandistic articles is our main objective, here we replicate. Thus, we use a Maximum Entropy classifier to discriminate between the four classes in the TSHP-17 corpus: trusted, hoax, satire, and propaganda relied on word n-gram features only. We also use these representations for this and the other experiments, and we consider them as a baseline. Our results using word n-grams on the original in- and out-of-domain partitions of the TSHP-17 corpus —including the void instances we discard for the rest of the experiments but we consider the model to have been successfully replicated. The evaluation results on the filtered corpora are slightly higher.

We performed an ablation study: using (i) each feature family in isolation and (ii) all but one. We study the performance of the resulting multi-class models when testing on articles from seen (in-domain) vs. unseen (out-of-domain) sources.

4.5.2. Experiment 2: Two-Way Classification on TSHP-17 and QProp

Since we are interested in the binary task of distinguishing propaganda vs. nonpropaganda, we asked ourselves whether the same drop between in-domain and out of-domain articles manifests in the binary classification setting as well. We perform our analysis on both corpora. For the TSHP-17 corpus, we do one vs. the rest by converting trusted, hoax and satire articles into the negative class and we test on the in-domain partition only. QProp is already a two-way classification corpus.

The corpora are highly imbalanced, and thus we will not show accuracy values. We first focus on the TSHP-17 corpus. The baseline word n-grams hold their status as a simple yet powerful representation, achieving an F1 of 90.76. Nevertheless, whereas the other representations show a performance from average to poor, one representation stands out: character n-grams yield an F1 of 96.22 (+5.46 with respect to word n-grams). The results on the QProp corpus. On a corpus with ten propagandistic sources, character n-grams outperform word n-grams by five or more points in both partitions —82.93 (+8.51) and 82.13 (+6.58). These differences between the word and character n-gram are statistically significant. The feature combination improves the performance significantly, i.e. in most cases the different feature families capture different aspects. On the TSHP-17 corpus, combining word and character n-grams boosts the performance by one point absolute with respect to the model using character n-grams only. The results on the development and on the test partitions of QProp vary: the best combination on development is character n-grams and NELA, whereas adding lexicon and vocabulary richness on top of them works best on test. Nevertheless, the difference between the results with this combination and the character n-grams alone is not statistically significant.

4.5. Experiment 3: Learning Propaganda vs. Learning the Source

In this experiment, we aim at analyzing whether our models learn to distinguish propagandistic vs. non-propagandistic articles as opposed to learning to recognize the news source an article is coming from. In order to do that, we first evaluate our models trained on the TSHP-17 corpus on its out-of-domain partition; i.e. on articles from unseen sources.

Features clearly improve with respect to the word n-grams F1, and the improvements are statistically significant.

The information available in our QProp corpus regarding the source of each article allows for a more sophisticated experiment. In particular, we reshape QProp by performing the following steps: (i) we merge the training, the development, and the testing partitions into one single collection; (ii) we randomly split the positive (negative) instances into two subsets: $Qprop_1^+$ and $Qprop_2^+$ ($Qprop_1^-$ and $Qprop_2^-$); and (iii) we compose a new training set by mixing $Qprop_1^+$and $Qprop_1^-$and a new testing set by mixing $Qprop_2^+$ and $Qprop_2^-$. We apply a number of constrains when producing this redistribution. First, we make sure there is no intersection between the sources in the new training and testing partitions. Second, we include an equal number of propagandistic and non-propagandistic sources in each partition. Third, we force the two propagandistic sources with less than 100 instances to be part of the test set. We perform several random samplings in order to come out with partitions as balanced as possible. We perform a number of experiments with an increasing number of instances on the training side, sampling subsets of positive instances according to their source. The procedure is as follows. Let s1……5 be the five propagandistic sources in the training set D^*_{tr}. We select at random $k \leq 5$ propagandistic sources and we keep only those documents belonging to the selected sources, resulting in D^*_{tr} . The negative instances are sub-sampled as well in order to resemble the distribution of the data in the original QProp, but regardless of their sources. We then train a model on the resulting D^*_{tr} and we evaluate it on the testing partition. We keep the test set untouched in all cases as, regardless of the sub-sampling, the models are always tested on articles whose sources, both propagandistic and non-propagandistic, were not seen during training. We repeated this experiment with all possible combinations of $k \in [1; 5]$ propagandistic sources and with all feature families.

4.6 Measuring Classifier Performance

Precision and Recall were the main measures used to evaluate classifier performance. Although the accuracy was also looked at, precision was a more significant measure because the dataset was imbalanced and a high accuracy when everything gets assigned to the majority class is misleading.

4.7 Algorithm Comparison and Selection

To compare the algorithms, statistical summary was calculated for each algorithm based on the results obtained for each code. The algorithm with optimal mean, median, max and min was selected for further study and as the final recommended model to use.

Results

5.1.Machine Learning Experiment Results

These machine-learning uses programmed algorithms that receive and analyses input data to predict output values within an acceptable range. As new data is fed to these algorithms, they learn and optimize their operations to improve performance, developing 'intelligence' over time.

5.1.1. Naïve Bayes Classifier – Count Vectorizer

```
----MultinomialNB with Count Vectorizer----

Overall accuracy score: 0.9307
Overall f1 score: 0.7349
Overall precision: 0.644
Overall recall: 0.8557

Classification Report:
              precision    recall  f1-score   support

          -1       0.98      0.94      0.96      4550
           1       0.64      0.86      0.73       575

    accuracy                           0.93      5125
   macro avg       0.81      0.90      0.85      5125
weighted avg       0.94      0.93      0.93      5125
```

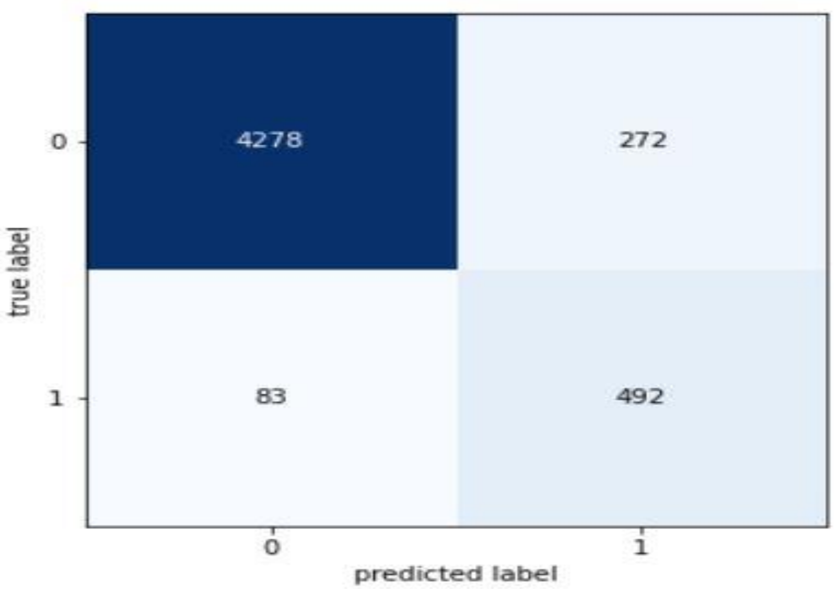


Figure 9: Naïve Bayes Classifier – Count Vectorizer

5.1.2. Naïve Bayes Classifier – tfidf Vectorizer

```
----MultinomialNB with tfidf Vectorizer----

Overall accuracy score: 0.9543
Overall f1 score: 0.7955
Overall precision: 0.7996
Overall recall: 0.7913

Classification Report:
              precision    recall  f1-score   support

          -1       0.97      0.97      0.97      4550
           1       0.80      0.79      0.80       575

    accuracy                           0.95      5125
   macro avg       0.89      0.88      0.88      5125
weighted avg       0.95      0.95      0.95      5125
```

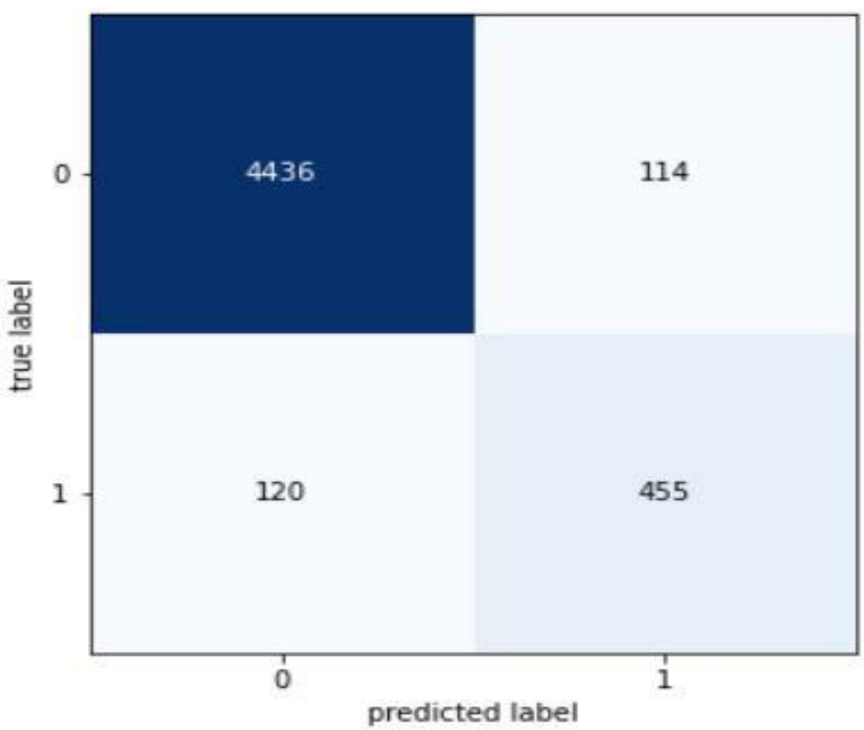


Figure 10: Naïve Bayes Classifier – tfidf Vectorizer

5.1.3. Logistic Regression – Count Vectorizer

```
----Logistic Regression with Count Vectorizer----

Overall accuracy score: 0.968
Overall f1 score: 0.8551
Overall precision: 0.8689
Overall recall: 0.8417

Classification Report:
              precision    recall  f1-score   support

          -1       0.98      0.98      0.98      4550
           1       0.87      0.84      0.86       575

    accuracy                           0.97      5125
   macro avg       0.92      0.91      0.92      5125
weighted avg       0.97      0.97      0.97      5125
```

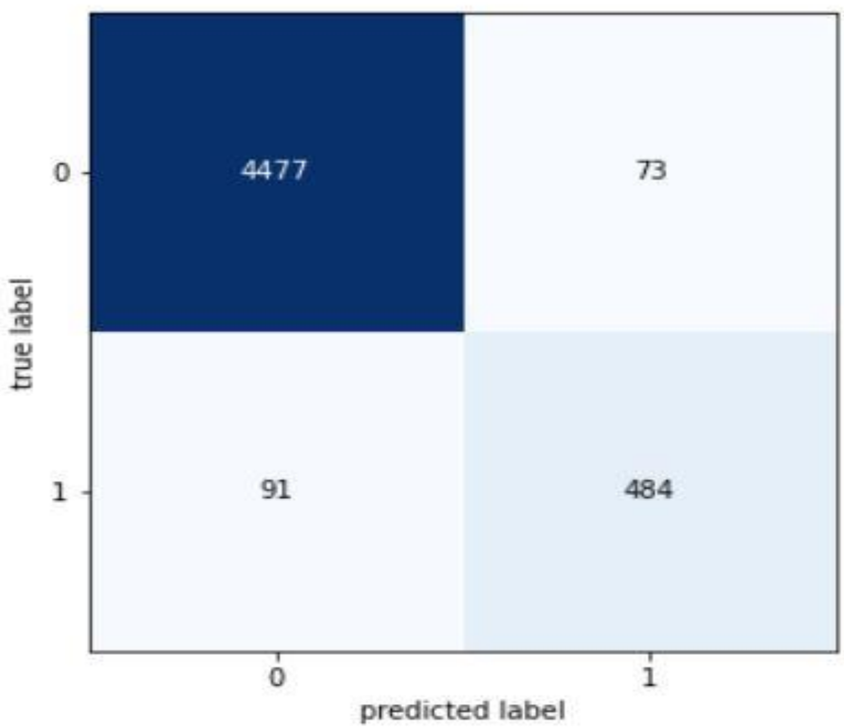


Figure11: Logistic Regression – Count Vectorizer

5.1.4. Logistic Regression – tfidf Vectorizer

```
----Logistic Regression with tfidf Vectorizer----

Overall accuracy score: 0.9727
Overall f1 score: 0.873
Overall precision: 0.9127
Overall recall: 0.8365

Classification Report:
              precision    recall  f1-score   support

          -1       0.98      0.99      0.98      4550
           1       0.91      0.84      0.87       575

    accuracy                           0.97      5125
   macro avg       0.95      0.91      0.93      5125
weighted avg       0.97      0.97      0.97      5125
```

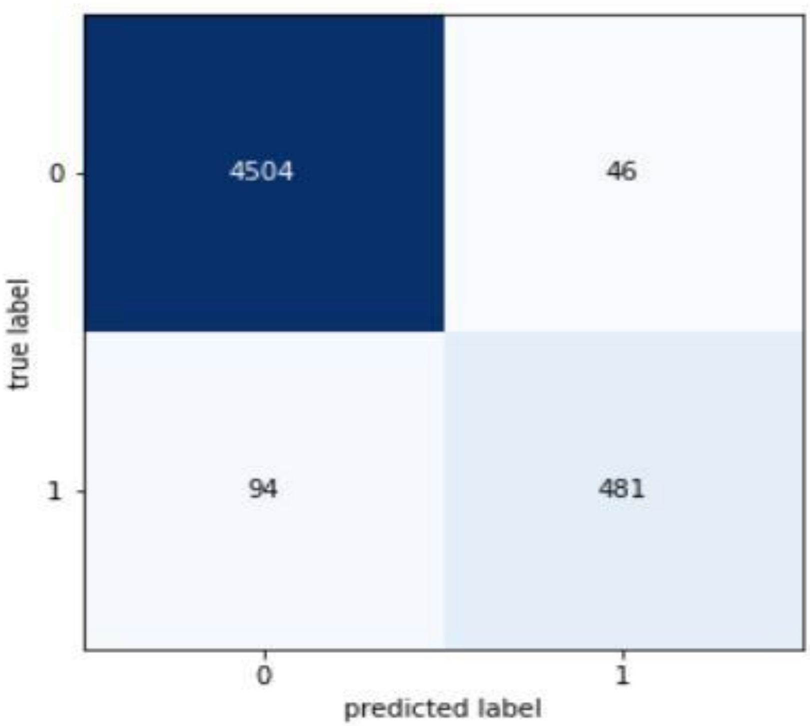


Figure12: Logistic Regression – tfidf Vectorizer

5.1.5. Support Vector Machine – Count Vectorizer

```
----SVM with Count Vectorizer----

Overall accuracy score: 0.9659
Overall f1 score: 0.8472
Overall precision: 0.8509
Overall recall: 0.8435

Classification Report:
              precision    recall  f1-score   support

          -1       0.98      0.98      0.98      4550
           1       0.85      0.84      0.85       575

    accuracy                           0.97      5125
   macro avg       0.92      0.91      0.91      5125
weighted avg       0.97      0.97      0.97      5125
```

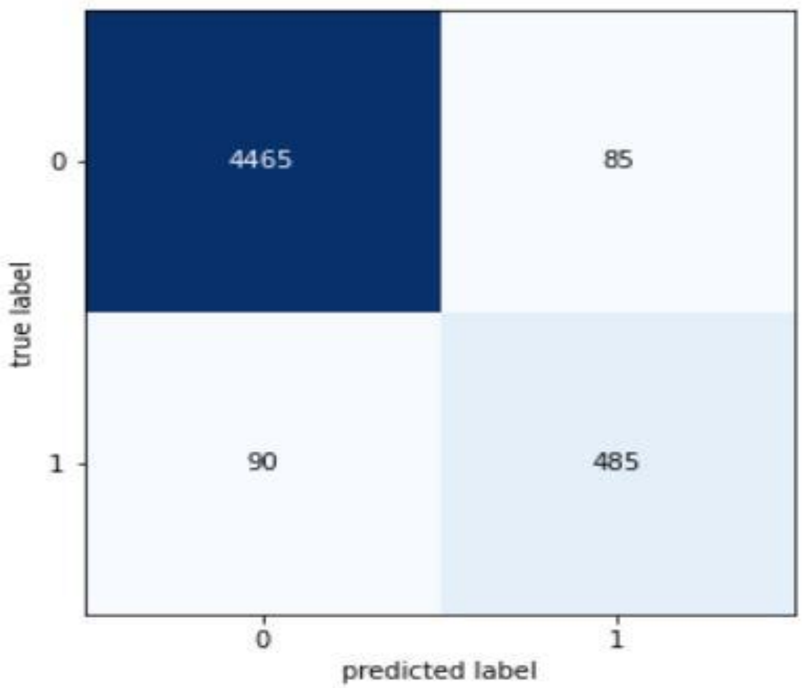


Figure13: Support Vector Machine – Count Vectorizer

5.1.6. Support Vector Machine – tfidf Vectorizer

```
----SVM with tfidf Vectorizer----

Overall accuracy score: 0.9721
Overall f1 score: 0.8731
Overall precision: 0.8913
Overall recall: 0.8557

Classification Report:
              precision    recall  f1-score   support

          -1       0.98      0.99      0.98      4550
           1       0.89      0.86      0.87       575

    accuracy                           0.97      5125
   macro avg       0.94      0.92      0.93      5125
weighted avg       0.97      0.97      0.97      5125
```

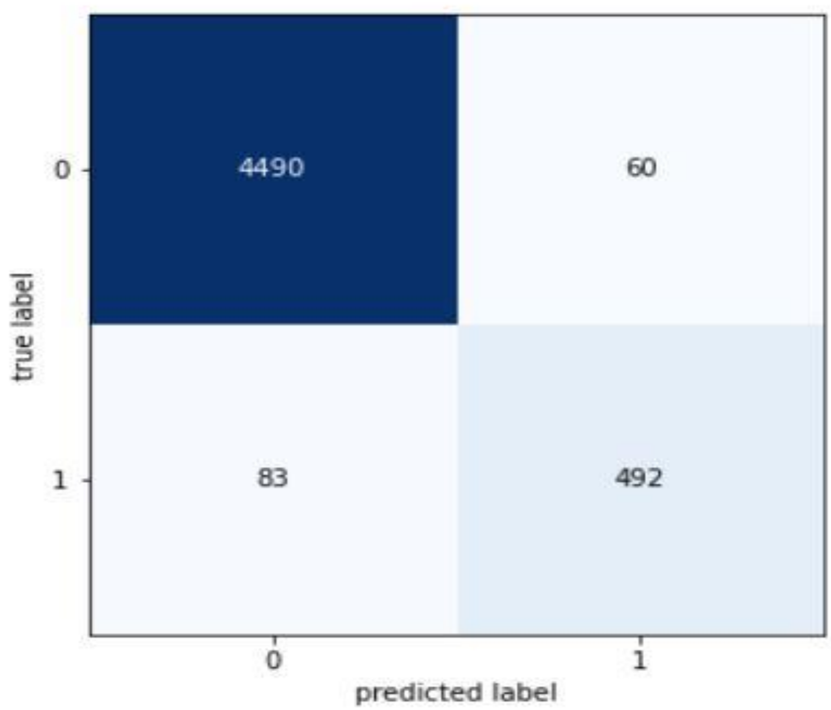


Figure14: Support Vector Machine – tfidf Vectorizer

5.2.Deep Learning Experiment Results

5.2.1. Long Short Term Memory – LSTM Model

5.2.1.1. Plots

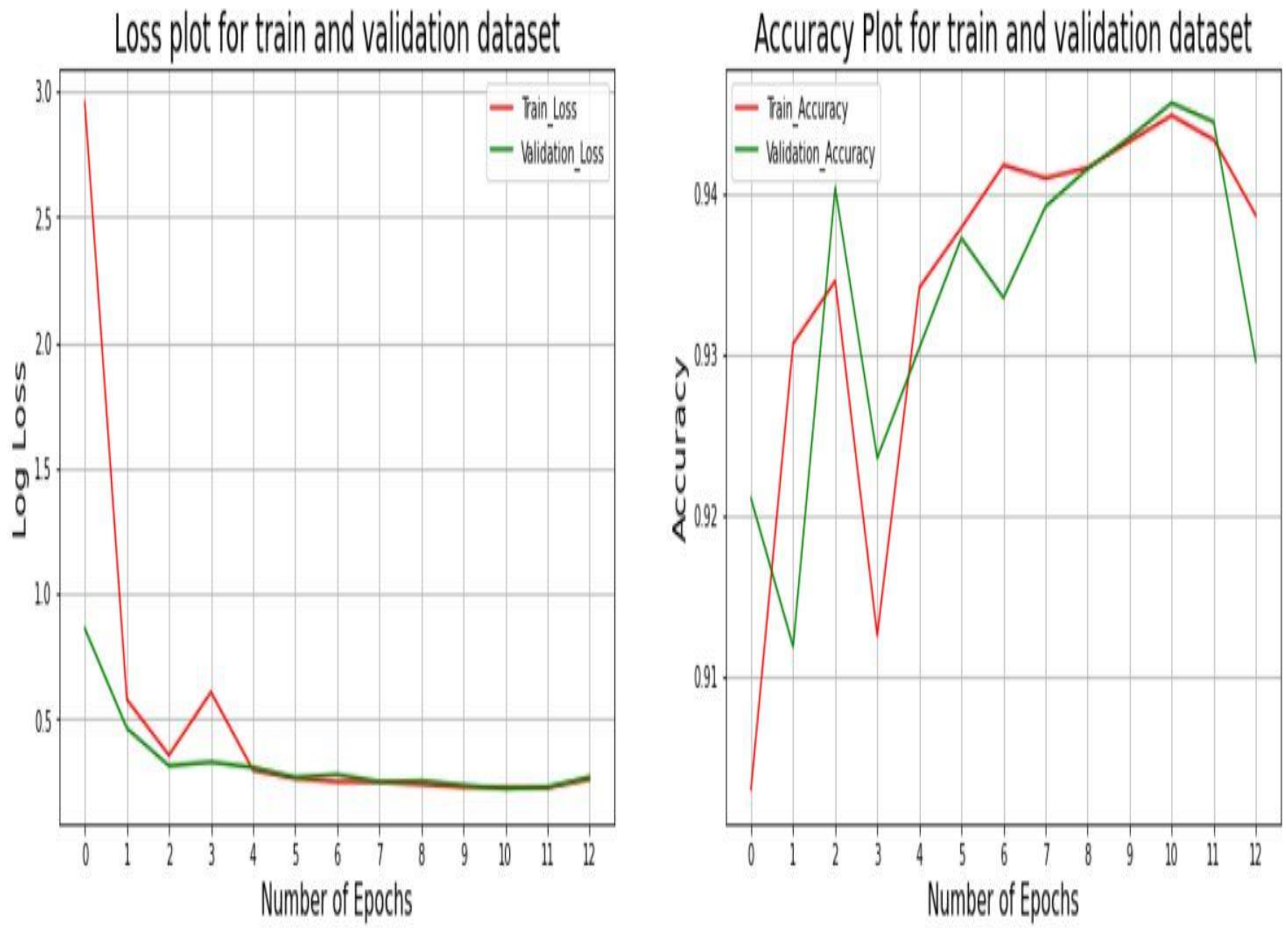


Figure15: LSTM – Loss & Accuracy for Train and Validation Dataset

5.2.1.2. Classification Report

Classification Report:

	precision	recall	f1-score	support
0	0.96	0.96	0.96	4550
1	0.69	0.69	0.69	575
accuracy			0.93	5125
macro avg	0.82	0.82	0.82	5125
weighted avg	0.93	0.93	0.93	5125

Figure16: LSTM – Classification Report

5.2.1.3. Confusion Matrix – Prediction.

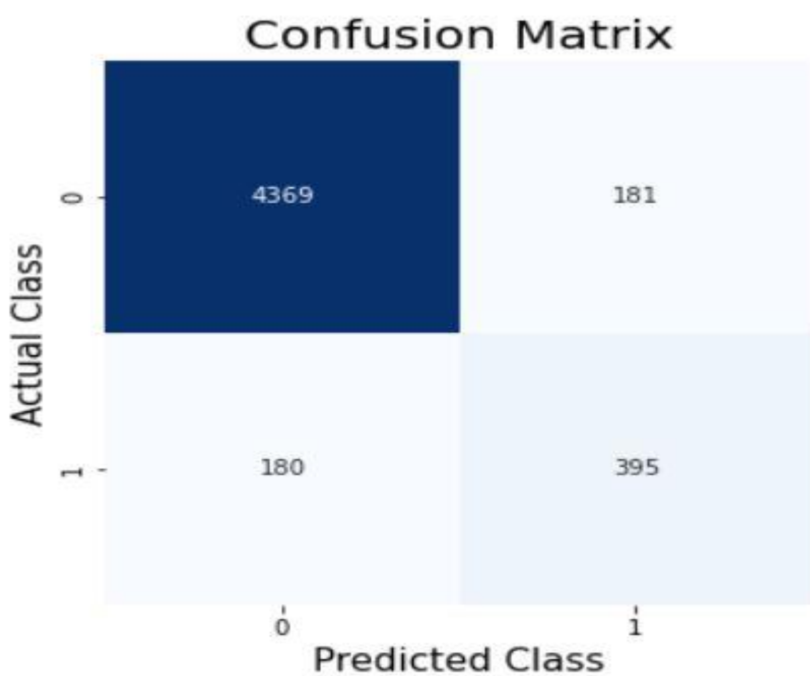


Figure17: LSTM – Confusion Matrix

5.2.2. Attention Long Short Term Memory Model

5.2.2.1. Plots

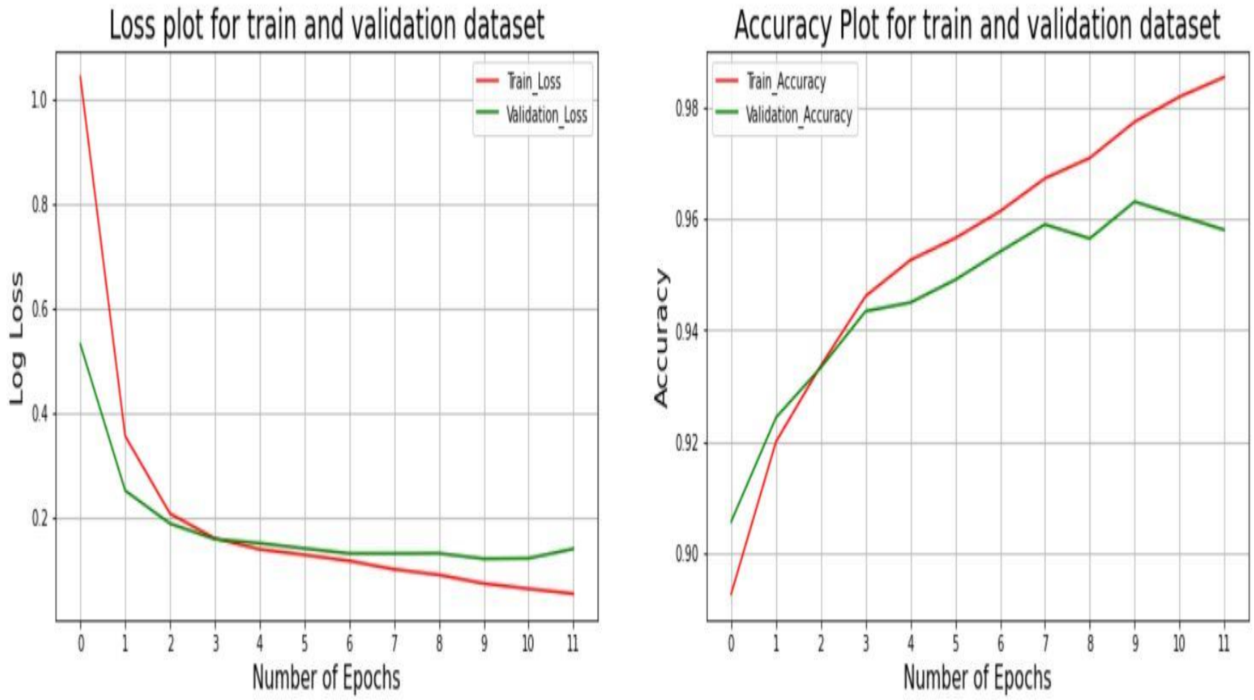


Figure18: Attention LSTM – Loss & Accuracy for Train and Validation Dataset

5.2.2.2. Classification Report

Classification Report:

	precision	recall	f1-score	support
0	0.98	0.97	0.98	4550
1	0.80	0.83	0.82	575
accuracy			0.96	5125
macro avg	0.89	0.90	0.90	5125
weighted avg	0.96	0.96	0.96	5125

Figure19: Attention LSTM – Classification Report

5.2.2.3. Confusion Matrix – Prediction.

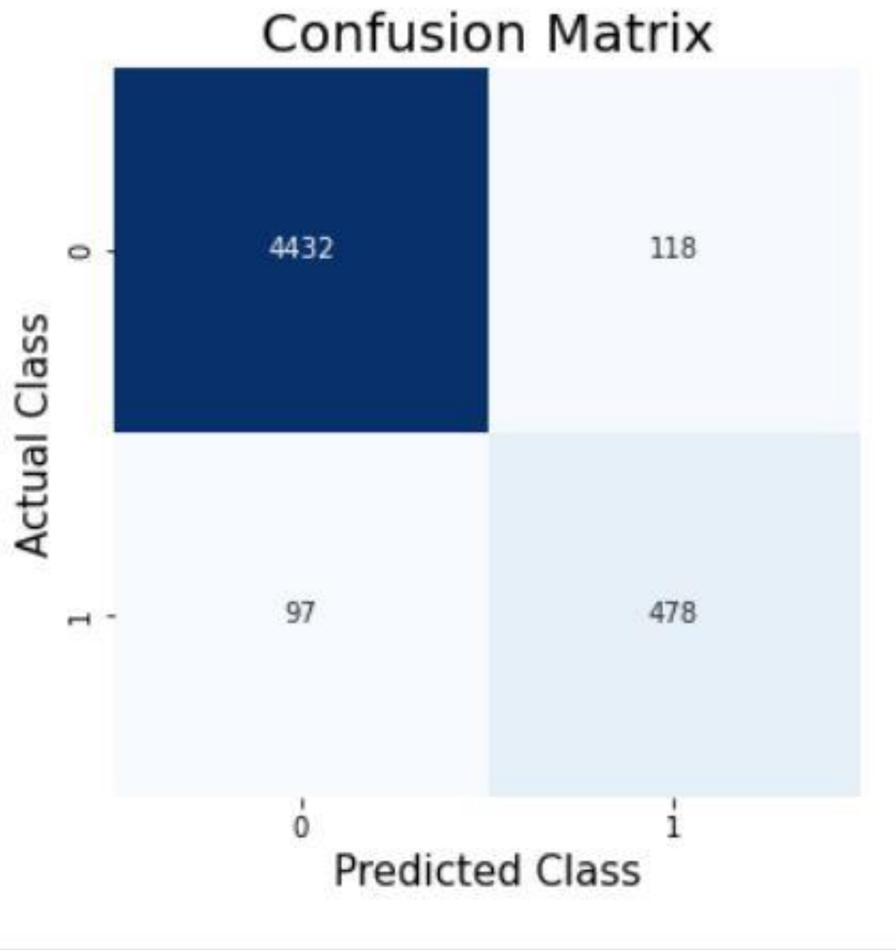


Figure20: Attention LSTM – Confusion Matrix

Conclusion and Future Work

I performed a thorough experimentation into propaganda detection at the news article level. My experimental results show that representations modeling writing style and text complexity are more effective than word *n*-grams, which model topics. My comparison against existing models corroborates this hypothesis: models that consider stylistic features, such as character *n*-grams always outperform alternative representations, which are typically used in topic-related tasks. Different from previous approaches, this is true also when trying to classify articles from sources unseen on training. This is a key asset when dealing with the never ending spawn of news outlets: propagandistic vs. other.

Further presented a system that organizes news articles into events and, for each event, shows articles according to their level of propagandistic content. The system is designed with the aim of raising awareness into individual readers as well as providing tools for organizations to monitor large amounts of news articles. Finally, I published an interface where our system organizes events according to propaganda, I also released the source code used in these experiments as well as my new corpus. I believe that these three resources are valuable for further research on propaganda detection, and that they will be also appreciated by the research community as well as by the general public.

Interesting avenues for future research include going into the fragment level and training models to identify specific propaganda techniques. That would allow for the creation of models able to explain their decisions and to give the user a clearer picture of what propagandistic techniques have been put in use.

Appendix A. Analysis of the most relevant word *n*-grams

In this appendix, I look at the most informative word *n*-grams as considered by the classifier to differentiate between propagandistic and non-propagandistic articles. In order to do that, I built a binary classifier on the QProp corpus only with word *n*grams and we retrieved the strings that the model assigned the highest weights to —both for the propaganda and for the non-propaganda classes.

Table A.1

Top-18 most significant word n-grams for the propaganda class (stop-word instances not shown); b=block of (semantically)-related instances (it links to the examples in Table A.15), w= weight assigned by the classifier.

b	w	*n*-gram	b	w	*n*-gram
1.	1.17	with permission	4.	0.49	american
	1.09	permission from	5.	0.47	the left
	0.99	article posted	6.	0.46	muslim
	0.98	article posted with	7.	0.46	obama
	0.97	posted with permission		0.45	clinton
	0.95	posted with	1.	0.43	originally published by
2.	0.63	the best of		0.43	whole article
	0.62	best of	8.	0.43	united states
3.	0.52	actually	1.	0.43	the whole article

Table A.1

Tables A.1 and A.2 show the most important word *n*-grams that help the classifier to decide whether a text should be classified as propagandistic or not. As Table A.1 shows, strings that refer to posting a piece of news after getting proper permission from another source (block 1) are among those with the highest weights. This may reflect that propagandistic articles tend to be re-posted in different media. Other strings are more related to superlatives. Also, three blocks include strings associated with people profiling or to specific characters. It is worth noting that the characters mentioned in block 7 have less media presence nowadays; therefore, relying on them to identify propaganda is a time-sensitive issue.

Table A.2

Top-18 most significant word n-grams for the negative class (instances with stop-words not shown); b=block of (semantically)-related instances (it links to the examples in Table A.16), w= weight of the classifier.

b	w	*n*-gram	b	w	*n*-gram	b	w	*n*-gram
1.	−1.69	said	1.	−0.39	told	3.	−0.33	saturday
	−0.63	said the	4.	−0.38	minister		−0.31	week
2.	−0.47	after	3.	−0.35	wednesday		−0.31	monday
1.	−0.44	he said		−0.34	tuesday	5.	−0.31	photo
2.	−0.43	last	1.	−0.33	said in	6.	−0.29	provided
3.	−0.41	thursday	3.	−0.33	friday	7.	−0.29	read more

Table A.2

On the other extreme, Table A.2 shows the highest-weighted word *n*-grams for the negative class non-propaganda. It is interesting to note that strings with “said” and related verbs are among those with the highest weights. This might reflect than non-propagandistic articles tend to quote the actors or reporters of the events. Having most weekdays reflects something similar: it is more likely that non-propagandistic news will cover a punctual event occurring at a specific time, rather than columns and other kinds of pieces. Tables A.3 and A.4 show some instances of these strings in context. This small subset of examples shows that indeed the *n*-grams associated with propagandistic articles tend to occur in propagandistic text snippets, whereas those associated with non-propaganda tend to occur in more neutral and objective sentences.

Table A.3

Collocation-like examples including the word n-grams in Table A.13 (linked by the number on the left of each block).

1.	Article	**posted with permission**	from Robert Spencer
	party or has been republished	**with permission**	from the author
	Article	**posted with permission**	from End of the American Dream
	This article was	**originally published**	by Adam Taggart at PeakProsperity.com
	This report was	**originally published**	by Jeremiah Johnson at Tess Pennington's
2.	their anger don't represent	**the best of**	America, they represent the worst of
	and gave us	**the best of**	medieval law
	after fellow venture members to	**the best of**	his ability
3.	If the NRA	**actually**	cared about the Second Amendment
	thereby endangering them, when	**actually**	all I did was respond to published
	the family noted that Roberson was	**actually**	wearing security attire
4.	Speaking at an African	**American**	church in Boynton Beach
	disgraceful in all of this is that the	**American**	people were promised a special prosecutor
	the increasing balkanization of the	**American**	body politic
5.	now expressing hatred (which	**the left**	does so well) rather than love
	the greatest existential threat	**the left**	has ever faced in America
	How has	**the left**	elite handled these allegations?
6.	the more	**muslim**	savages we allow into america
	There is no such thing as a moderate	**muslim**	and there never will be
	Ally will be the first	**muslim**	male Judge in New York
7.	Barack Hussein	**Obama**	Soetoro Sobarkah
	Americans praised him under	**Obama**	and demonize him under Trump
	Why aren't they going after Hillary	**Clinton**	with her emails and with the dossier
8.	If that actually happened in the	**United States**	of America and everything each and every
	the Missile Defense Agency and the	**United States**	government in their ballistic missile defense
	this "sticks in the craw" of the	**United States**	and the Western Financial,

Table A.4

Collocation-like examples including the word n-grams in Table A.14 (linked by the number on the left of each block).

1.	With that	**said**	, while President Donald Trump
	Republican Congressman Trey Gowdy	**said**	he thought it was politically smart
	Tempe Police Department,	**said**	the women were arrested
2.	video footage released	**after**	the official meeting
	As TFTP reported	**last**	week, Carol Davidsen
	This monstrous slaughter took place	**last**	October, and still the FBI has nothing
3.	Miami collapsed on	**Thursday**	, possibly killing several motorists
	shooter drills at the school that very	**week**	and that they would be firing
	President Trump on	**Wednesday**	voiced support for confiscating guns
4.	A Lutheran	**minister**	and early Nazi supporter
	Does the	**Minister**	agree that Tommy Robinson
	by Home Office	**Minister**	Ben Wallace
5.	that he posted a	**photo**	on Facebook
	Relying on a	**photo**	posted on Collins Facebook
	Then after the	**photo**	was taken
6.	tested for a rape kit and she	**provided**	a written account
	Brandon Curtis at Concealed Nation	**provided**	some thoughts
	was not based on any information	**provided**	to her by Obama himself
7.		**Read more**	about the Thursday activities here
		Read more	about that by clicking linked
	document here, and	**read more**	about it here

Appendix B

Dataset Links

Dataset: https://zenodo.org/record/3271522#.YgK9ie5BzDI

https://github.com/mofasa-20/PROPAGANDA-TEXT-CLASSIFICATON-ANALYSIS-/tree/main/Input-Data

Code File

https://github.com/mofasa-20/PROPAGANDA-TEXT-CLASSIFICATON-ANALYSIS-/tree/main/Codes

GitHub Link

https://github.com/mofasa-20/PROPAGANDA-TEXT-CLASSIFICATON-ANALYSIS-

References

ALPAC- Automatic Language Processing Advisory Committee

Ba, M. L., Berti-Equille, L., Shah, K., & Hammady, H. M. (2016). VERA: A platform for veracity estimation over web data. Proceedings of the 25th International Conference Companion on World Wide WebWWW '16159–162 Montréal, Québec, Canada Baeza-Yates, R. (2018).

Recursive hetero-associative memories for translation

Bias on the web. Communications of the ACM, 61(6), 54–61.

Recurrent Continuous Translation Models

Baly, R., Karadzhov, G., Alexandrov, D., Glass, J., & Nakov, P. (2018). Predicting factuality of reporting and bias of news media sources. Proceedings of the 2018 Conference on Empirical Methods in nAtural Language ProcessingEMNLP '183528–3539.

Sequence to Sequence Learning with Neural Networks

Barrón-Cedeño, A., Da San Martino, G., Zhang, Y., Ali, A., & Dalvi, F. (2018). Qlusty: Quick and dirty generation of event videos from written media coverage. Proceedings of the Second International Workshop on Recent Trends in News Information Retrieval27–32 Grenoble, France

Neural Computation - Long Short-Term Memory - LSTM

Bazerman, C. (2010). The informed writer: Using sources in the disciplines. Fort Collins, CO, USA: The WAC Clearinghouse. ALPAC- Automatic Language Processing Advisory Committee

JANUS: a speech-to-speech translation system using connectionist and symbolic processing Strategies

Neural Machine Translation by Jointly Learning to Align and Translate

Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation

Convolutional Sequence to Sequence Learning

Bird, S., Loper, E., & Klein, E. (2009). Natural language processing with python. O'Reilly Media Inc.

Brill, A. M. (2001). Online journalists embrace new marketing function. Newspaper Research Journal, 22(2), 28.

Canini, K. R., Suh, B., & Pirolli, P. L. (2011). Finding credible information sources in social networks based on content and social structure. Proceedings of the IEEE International Conference on Privacy, Security, Risk, and Trust, and the IEEE International Conference on Social ComputingSocialCom/PASSAT '111–8 Boston, Massachusetts, USA

Castillo, C., Mendoza, M., & Poblete, B. (2011). Information credibility on Twitter. Proceedings of the 20th International Conference on World Wide WebWWWZ'11675–684 Hyderabad, India

Chen, C., Wu, K., Srinivasan, V., & Zhang, X. (2013). Battling the Internet Water Army: Detection of hidden paid posters. Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and MiningASONAM '13116–120 Niagara, Ontario, Canada

Conserva, H. (2003). Propaganda techniques. United States: AuthorHouse.

Cristianini, N., & Shawe-Taylor, J. (2000). An Introduction to Support Vector Machines and other kernel-based learning methods. Cambridge University Press.

Derczynski, L., Bontcheva, K., Liakata, M., Procter, R., Wong Sak Hoi, G., & Zubiaga, A. (2017). SemEval-2017 Task 8: RumourEval: Determining rumour veracity and support for rumours. Proceedings of the 11th International Workshop on Semantic EvaluationSemEval '1760–67 Vancouver, Canada

Ellul, J. (1965). Propaganda: The formation of men's attitudes. United States: Vintage Books.

Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. Proceedings of the Second International Conference on Knowledge Discovery and Data MiningKDD '96226–231 Portland, OR, USA

Graham, J., Haidt, J., & Nosek, B. A. (2009). Liberals and conservatives rely on different sets of moral foundations. Journal of Personality and Social Psychology, 96(5),1029.

Gunning, R. (1968). The technique of clear writing. McGraw-Hill.

Hardalov, M., Koychev, I., & Nakov, P. (2016). In search of credible news. Proceedings of the 17th International Conference on Artificial Intelligence: Methodology, systems, and applicationsAIMSA '16172–180 Varna, Bulgaria

Hooper, J. (1975). On assertive predicates. 4. Academic Press, New York.

Horne, B. D., & Adal, S. (2017). This just in: Fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news. Proceedings of the international workshop on news and public opinion at ICWSM Montreal, Canada

How to Detect Propaganda (1938). In Institute for Propaganda Analysis (Ed.). Propaganda analysis. volume i of the publications of the institute for propaganda analysis (pp. 210–218). New York, NY

Jaccard, P. (1901). Étude comparative de la distribution florale dans une portion des Alpes et des Jura. Bulletin del la Société Vaudoise des Sciences Naturelles, 37, 547–579.

Japkowicz, N., & Shah, M. (2011). Evaluating learning algorithms: A classification perspective. New York, NY: Cambridge University Press.

Translation Engines

Potthast, M., Kiesel, J., Reinartz, K., Bevendorff, J., & Stein, B. (2018). A stylometric inquiry into hyperpartisan and fake news. Proceedings of the 56th annual meeting of the Association for Computational LinguisticsACL '18231–240 Melbourne, Australia

Rashkin, H., Choi, E., Jang, J. Y., Volkova, S., & Choi, Y. (2017). Truth of varying shades: Analyzing language in fake news and political fact-checking. Proceedings of the 2017 Conference on Empirical Methods in Natural Language ProcessingEMNLP '172931–2937 Copenhagen, Denmark

Shu, K., Sliva, A., Wang, S., Tang, J., & Liu, H. (2017). Fake news detection on social media: A data mining perspective. SIGKDD Explorations Newsletters, 19(1), 22–36.

Sproule, J. M. (2001). Authorship and origins of the seven propaganda devices: A research note. Rhetoric & Public Affairs,, 4(1), 135–143.

Stamatatos, E. (2009). A survey of modern authorship attribution methods. Journal of the American Society for Information Science and Technology, 60(3), 538–556.

Thorne, J., & Vlachos, A. (2018). Automated fact checking: Task formulations, methods and future directions. Proceedings of the 27th International Conference on Computational LinguisticsCOLING '183346–3359 Santa Fe, NM, USA

Thorne, J., Vlachos, A., Christodoulopoulos, C., & Mittal, A. (2018). FEVER: A large-scale dataset for fact extraction and VERification. Proceedings of the 2018 conference of the North American chapter of the Association for Computational Linguistics: Human language technologiesNAACL-HLT '18809–819 New Orleans, LA, USA

Vosoughi, S., Roy, D., & Aral, S. (2018). The spread of true and false news online. Science, 359(6380), 1146–1151.

Wilson, T., Wiebe, J., & Hoffmann, P. (2005). Recognizing contextual polarity in phrase-level sentiment analysis. Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language ProcessingNAACL-HLT '05347–354 Vancouver, Canada

Zubiaga, A., Liakata, M., Procter, R., Wong Sak Hoi, G., & Tolmie, P. (2016). Analysing how people orient to and spread rumours in social media by looking at conversational threads. PLoS ONE, 11(3), 1–29.