Bachelor's Thesis

Fake News Detection - Natural Language Processing

Chrysovalantis Christodoulou

University of Cyprus

Department of Computer Science

April 2019

Acknowledgment

I would like to express my heartfelt gratitude to my supervisor Dr. George Pallis, Assistant Professor in the Computer Science Department of University of Cyprus, for his endless willing and the opportunity that gave me to work my thesis in one of the biggest new-age problems. He gave me all the support and encouragement throughout the elaboration of my thesis project.

Moreover, I would like to thank the Ph.D. candidate Demetris Paschalidis for his excellent guidance and the chance he gave me to learn an incredible amount of technologies.

Last but not least, I would like to acknowledge and thank my family for their valuable help during my studies. Specifically, I would like to thank my parents Stelios and Maria and my sweetheart Elena for all the psychological and physical aid they graciously provided to me.

Abstract

The recently increased focus on misinformation has spurred research in fact checking, the task of assessing the truthfulness of a claim. People become victims of fake news during their daily lives and support their further spread intentionally or recklessly. The colossal propagation of information makes the necessity for an autonomous fake news detection system more imperative than ever before. Despite the undeniable need, there is not enough coverage because of the problem's complexity. In this thesis, we focus on fake news detection using Natural Language Processing characteristics, and we examine the impact of a variety of different features. Therefore, to do so, we gather an enormous dataset, and we extract an extensive amount of features. We divide those features into three main different domains and six subdomains. Then we have performed a data visualization process in order to get a further understanding of our features, and then we implement five feature selection techniques on them. We end up with the twenty most prominent features, and then we have trained a deep learning model with them. Moreover, we trained the model with different collections of features, and compare those results with the top twenty features. Using the top twenty features, we achieved a state-of-the-art outcome with F1-score over 93%.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Contents

1.1 Motivation

In the recent years, fake news attracted growing interest from the general public and researchers as the distribution of misinformation online advances, particularly in media outlets such as social media feeds, news blogs, and online newspapers. Journalist deal with misinformation spreading since the previous century and for a long time they didn't face tremendous obstacles. The evolution of the Internet reinvents not only the journalist work but also the way people inform. Nowadays, we escape from the daily newspapers and we have 24/7 instant information from a bunch of different sources, many of them unsigned. The collosal propagation of information made the need for an autonomous fake news detection system more imperative than ever before. However, the spark that lights up the increasing interest in misinformation spreading is the U.S 2016 elections.

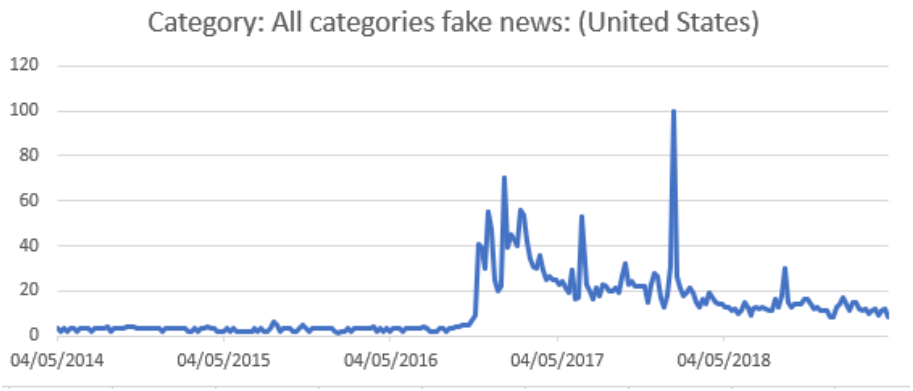


Figure 1.1: Google trends for U.S in category: Fake News

Governments and the general public get awake from the fact that the election of the president of one of the most powerful countries in the word affected by fake news dissemination. Figure 1.1 displays the increase of people searching about fake news in Google and as you can witness the contrast before 2016 and later of that year is crystal clear. The peak of 14/01/2018 is due to the tweets of president Donald Trump, who attacks the Wall Street Journal as 'fake news' over his North Korea comments.

Despite the undeniable need for an autonomous fake news classification tool, there is very little work to cover the demand and the reason for that is the complexity of the problem. It is tough even for humans to detect fake news. It can be claimed that the only way for a person to manually identify fake news is to have a vast knowledge of the covered topic. Furthermore, misinformation identification concerns a lot of different disciplines, which may use inconsistence terminology and may not even know each other. Vlachos and Throne [1] identify the issue and compose an article to bridge the gap between those disciplines and gather all of the various approaches. In addition, they declare the difficulties of each approach and they propose future NLP research on automated fact-checking. In this research, we are focusing on how Natural Language Processing will help us classify fake news articles.

1.2 Challenges

By definition, Fake News detection is a challenging issue from every perspective. Even for a journalist, the confirmation of the veracity of a news article constitutes some thorough research around the topic. Moreover, a news article might not be entirely false or maybe is not completely true and that drives to a dilemma about the classification of it. Another challenging part includes the writing of those fraudulent statements by expert journalists who are chosen to bias the common thought to serve political interests as it happens in the U.S 2016 elections. The major issue is that those journalists are experts to avoid common mistakes such as grammatical or syntax and hide their purpose behind strong and powerful words. Thus, our natural language analysis getting harder and harder and we have to take into account those parameters.

In addition, natural language processing is a subfield of artificial intelligence aims to convert a text into a programmer-friendly data structure that describes the initial meaning of the text. Despite all the researches, the nature of language is obscure and sometimes confused, thus the outcome might not be as accurate as we expect to be. We have to be very precise in our metrics and ensure the validity of our conclusions. Natural language processing requires a lot of knowledge to understand and make use of the unlimited information might give you. We had to study a lot of literature to select every possible feature might give us a better understand of the text and help the model classify its content as fake or not.

The collection of data is another challenging part of the research because there are not enough novel datasets to train a deep learning model and you need a lot of "digging" to find something relevant. Then you have to be aware that in a novel dataset some of the data might need preparation or deletion because they might contain irrelevant information. For example, our dataset includes some Russian articles which do not consider our work because we are working on English natural language processing.

Along with the previous challenges, we had to deal with some technical restrictions. As we will mentioned later, our research contributes to Check It which is a plugin for the browser. Check It restriction is to produce every calculation in the client browser because they want to avoid any GDPR issues and ensure the user that nothing from his/her navigation sent to Check It for further calculation. That drives to a serious processing limitation and force us to implement a lightweight, fast and accurate method. Thus we had to turn down high processing power consuming features such as N-grams and TF and ensure the fastest implementation of every feature we used. Another challenge we had to address because of that demand, is the extraction of every feature in JavaScript which is not established for natural language processing and it contains a limited amount of libraries about that topic. Thus, we extract every feature in Python, which is a more suitable programming language for that purpose and then we transformed the code into JavaScript code. Then we compare the features of Python and JavaScript to ensure the validity of our results. Moreover, there was a need to train the model using features extracted from JavaScript and thus we implement a NodeJS project to address it.

1.3 Contributions

The ultimate goal of our research is to examine how natural language processing characteristics will help in this new age problem called Fake News automated detection. Therefore, we analyze tones of news articles and we extract an enormous amount of NLP features. We divide those features into three major classes: Dictionaries, Complexity, and Stylistic features and then we redivide those categories into six subcategories in order to examine separately and combined their contribution. The distribution of our features into different classes aims to organize natural language processing features and help the researches decide a suitable category in their case. Furthermore, we implement some very effective feature selection techniques to analyze our feature as a union and discover the combination of them will produce the best outcome. We showed that feature selection techniques help us gain the best features of our dataset and deliver the best result. In addition, we examine the impact of each subcategory separately and combined by training our model and

compare our outcomes. We provide a visual representation of our experiment results and a table with specific values of each metric for every combination.

Moreover, we have the honor to contribute a big scale project funded by Google called Check It, which is a browser extension aims to inform users about fake news articles. Check IT is a system that combines, in an intelligent way, a variety of signals such as, domain flag-lists, online social network, and natural language processing characteristics, into a pipeline for fake news identification. To sum up, our contributions are as follows:

- Examine the effect of different natural language process features in fake news classification.
- Extract those features to contribute the building of Check It.

Our contribution to the detection of online misinformation has to be effective and help the researches and the public address this disturbing issue.

1.4 Outline Contents

The organization of our contents is as follows.

Chapter 1: Introduction
The introduction defines the motivation of our research and making clear how significant is the addressing of the Fake News problem. Moreover, explains the contribution of our study and the challenges we faced during our research.

Chapter 2: Related Work
Chapter 2 focus on an analytical review of the literature, which consists of work related to fake news classification, more specifically we examine the previous studies on fake news detection using natural language processing characteristics, feature extraction, and feature selection.

Chapter 3: Methodology

The methodology chapter defines our methodology and explains each step we made very precisely. We described how we find our datasets and the role of them in our study. Moreover, we analyze the feature extraction process and how we divide our features into three main different categories which are: Dictionary, Stylistic and Complexity features. Then we present how we visualized our feature to get a further understanding of them and then we mention our five different feature selection methods we implement to analyze the importance of our features. Finally, we described from an abstract point of you the structure of our deep neural network.

Chapter 4: Experiments

Chapter 4 focus on presenting the results of our experiments and describes precisely the scores we used to measure them. We trained our model with some combination of our categories and with the top 20 features of the feature extraction process and then we compare the outcomes. We present a visual comparison of our results and we provide a table with specific values for each experiment. Moreover, we optimize our top 20 features' result to cover the needs of Check It.

Chapter 5: Conclusion

Chapter 5 defines our conclusions thoughts and summarize them to provide a brief outcome. Finally, we propose two significant important future works which will extend our work and help the research with fake news detection.

Chapter 2

Related work

Contents

2.1 Fake News Detection

Fake news detection has earned a lot of attention especially as it is claimed to have a significant impact on 2016 US Presidential Elections [2]. The issue concern not only Computer Scientist but also journalists. Generally, fact-checking is a tough work even for a human and most of the people declared that you can not point something as fake until you have a thorough opinion about the whole topic of the article. Moreover, the assumption in fake news discussion is that it is written to look like real news, tricking the reader who does not check for the authenticity of the sources or the evidence in its content was not really accurate. Regarding, a study from Rensselaer Polytechnic Institute of New York fake news is achieved by heuristics rather than the intensity of the arguments. Using three different datasets they have proved that title structure and the use of NNP (proper nouns) are very

important in differentiating fake from real news [3].

Regarding the difficulty of declaring an article as fake or real some other studies proposed a different approach. For example, the article Five Shades of Untruth: Finer-Grained Classification of Fake News [4] divides the content into five different categories: Factual (true or mostly true), Incomplete Propaganda, Manipulative Propaganda, Hoax, and Irony. This division helped the researches understand the difference between fake news subcategories and identify the variations in the corresponding features. A similar approach is applied by William Yang Wang which divides the fake news into six fine-grained labels for the truthfulness ratings: pants-fire, false, barely- true, half-true, mostly-true, and true. These six subcategories arranged with the help of Politifact users with the evaluation of the editors of the site.

The attempting of defining fake news doesn't stop in the previous approaches. Some very effective approaches analyzing the network characteristics and the spread of fake news through social media to extract features which help us to declare the truthfulness of an article. A serious study in the spreading of true and false news online published in the Science Journal and the results of this study gave us a further understanding of this phenomenon called fake news. Regarding the research which is based on Twitter social media, the diffusion of fake news tweets are significantly farther, faster, deeper, and more broadly than the truth in all categories of information. Moreover, most of the fake tweets are coming from unverified users with young age accounts and less number of followers and followers and the topic of the tweet usually concern politics. Although, the diffusion of fake tweets are faster the life span of them are short due to the revelation of the truthfulness.

All of the above approaches have a significant impact on the declaration of fake news articles however, on this report I am focusing on natural language processing characteristics.

2.2 Natural Language Processing

Natural Language processing is a subfield of artificial intelligence aims to convert a text into a programmer-friendly data structure that describes the original meaning of the text. The studies around natural language processing were dominated by machine-learning approaches that used linear models such as SVM (support vector machine) or logistic regression [5].

Nowadays, this approach tends to be replaced by non-linear neural networks models like the MLP (Multilevel Perceptron) and convolutional neural networks, which are very effective when the input is images or sound files. The replacement of those linear models with these feed-forward networks are proved to be very effective in the final results in a series of works [6]–[9].

Despite the fact that convolutional neural networks most commonly applied to analyzing visual imagery, they have promising results in natural language processing too. Significantly, due to the ability, they have in recognizing patterns independent of their position they can simply determine specific sentences or phrases which are great indicators of the topic of an article [10]. In addition, it seems to have promising results in sentiment classification [11], short-text categorization [12] and modeling the relation between character-sequences and part-of-speech tags [13].

The advantage they have to recognize sentences and phrases despite their position became a disadvantage when we want to keep the structure of the text unaffected. This space can be cover by Recursive Neural Networks [14] and Recurrent Neural Networks [15] which allows working with structure input and have great results in natural language processing area.

Recurrent models have been shown to produce very strong results for language modeling, including as well as for machine translation , dialog state tracking, dependency parsing, response generation, sentiment analysis, sequence tagging, noisy text normalization, and modeling the relation between character sequences and part-

of-speech tags [5].

According to research at the University of Stanford Recursive Neural Network combined with PCFGs (Probabilistic context-free grammar) improves the previous Stanford POS (part of speech) tagger parser by 3.8% to obtain an F1 score of 90.4%. The results are outstanding and if are not state-of-the-art is near state-of-the-art.[16]. Moreover, another study again from the University of Stanford increase the state of the art in a single sentence positive/negative classification from 80% up to 85.4%. The precision of predicting fine-grained sentiment labels for all phrases reaches 80.7%, a rise of 9.7% over the pack of features baselines [16]

2.3 Feature Extraction

Feature Extraction is one of the most critical parts of the machine learning process because starts from an initial set of data and convert them into meaningful and non-redundant information, which facilitates the subsequent learning and generalization steps. In our case, feature extraction is base on natural language characteristics and there are a plethora of features to arrange.

Most of the times when we have to extract textual features we have to apply some data preparation methods. According to Detection of Online Fake News Using N-Gram Analysis and Machine Learning Techniques [17] the text needs to be subjected to certain refinements like stemming, stop-word removal and tokenization. Tokenization can be either word tokenization, sentence tokenization even paragraph tokenization and help us to split the text into tokens of individual words, sentences or paragraphs. Stop words removal is an action after tokenization as the article says and the goal of that step is to remove insignificant words of a language like ”a, about, an, are, as, at, be, by, for, from, how, in, is, of, on, or, that, these, this, too, was, what, when, where, will, etc ”. Sometimes those words may create noise in text classification because they are commonly used and usually doesn't have anything important to offer in the text classification process.Now stemming, is the process of transforming the tokens into a standard form which means that change the word

from its current form into its original form. The outcome of that process is the reduction of word types or classes in the data. For example, the words "Running", "Ran" and "Runner" will be reduced to the word "run." The authors use Porter stemmer, which is the most commonly used and trustful stemming algorithm.

A famous feature identification and analysis approach commonly used in Natural Language processing areas are N-grams. This method applied in a series of works regarding fake news detection [17]–[19]. In the paper Detection of Online Fake News Using N-Gram Analysis and Machine Learning Techniques, the authors focus on word-based n-gram to represent the context of the document and generate features to classify the document. They produce uni-grams, bi-grams, tri-grams, and four-grams with various sizes and they have reached 92% accuracy using unigram features with a simple LSVM (Lagrangian Support Vector Machine) classifier.

However, linguistic features don't stop on n-grams. Another popular feature regarding [20] is the analysis of punctuations in a text. In the journal article (Automatic Detection of Fake News) the authors calculate the number of periods, commas, dashes, question marks and exclamation marks in the text and the results confirm Rubin. In the same article, they have also extract Psycholinguistic, Readability and Syntax features. Regarding Psycholinguistic features, the authors used the famous LIWC (Linguistic Inquiry and Word Count software) which is a large lexicon help us to understand the sentiment of the text. Readability features indicate text understandability using the number of complex words, the number of syllables and some widely used readability index metrics such as Flesch-Kincaid, Flesch Reading Ease, Gunning Fog, and the Automatic Readability Index (ARI). Moreover, they have also produced CFG (Context Free Grammar) derived features which are a set of recursive rewriting rules used to generate patterns of strings.

The authors of the article trained a model first using each feature category individually and then combined all the features. Then they have compared the results of the model and they ended with some conclusions. The combination of all features gave them the 2nd best accuracy 74%, but the first places held by Readability index

features 78%. The worst accuracy belonged to Psychological features with 56%.

2.4 Feature Selection

Feature Selection is the process of discovering and selecting the most informative and relative features from a dataset. Moreover according to Feature Extraction: Foundations and Applications book the feature selection process offers general data reduction, to limit storage demands and increase algorithm speed, feature set reduction, to save resources in the next round of data collection or during utilization, model performance improvement, to gain in predictive accuracy, and data understanding, to help you increase your knowledge about the process that generates the data.

There are several feature selection techniques such as filter methods which provide a complete order of the features using a relevance index. Some classical test statistics are T-test, F-test, Chi-squared test are considered as filter methods. Those methods, according to the book, select features without optimizing the performance of a predictor. On the other hand, wrappers and embedded methods involve predictor as part of the selection process. Wrappers divide the features into subsets and using the predictor calculates the predictive accuracy of each subset. Embedded methods perform feature selection in the process of training and are usually specific to given learning machines.

A famous method for feature selection is Individual relevance ranking which means that we are testing how much an individual feature affects the model prediction. However, this method carries a serious disadvantage. Features that are not individually relevant may become relevant in the context of others and features that are individually relevant may not all be useful because of possible redundancies. Thus we justify the use of multivariate methods, which make use of the predictive power of features considered jointly rather than independently. The Relief method considers as one of the most commonly used methods to calculate the impact of subsets on a model. Moreover, it has the advantage to take into account the feature redundancy

and generate more compact subsets of features.

Two well know greedy methods are Forward and Backward selection algorithms. Both of them are greedy algorithms and consume a lot of time and resources to deliver an outcome. Although, both methods are time and resources consuming they provide very good results because they examine every single combination of features and give us the one which produces the best results. The only difference between these methods is the way they build the subsets. The forward algorithm chooses a subset of the predictor variables for the final model, instead of the backward algorithm which begins with the full least squares model containing all predictors, and then iteratively removes the least useful predictor, one-at-a-time.

Chapter 3

Methodology

Contents

3.1 Methodology Overview

Our methodology is base on four major pillars: Data Collection, Feature Extraction, Data Visualization, Feature Selection, and the Deep Neural Network. An overview of our architecture depicted in Figure 3.1. Firstly, we have to collect our data and then extract features from those data. Then, we have to examine our features and pass to the neural network only the most essential features. In this research, we are

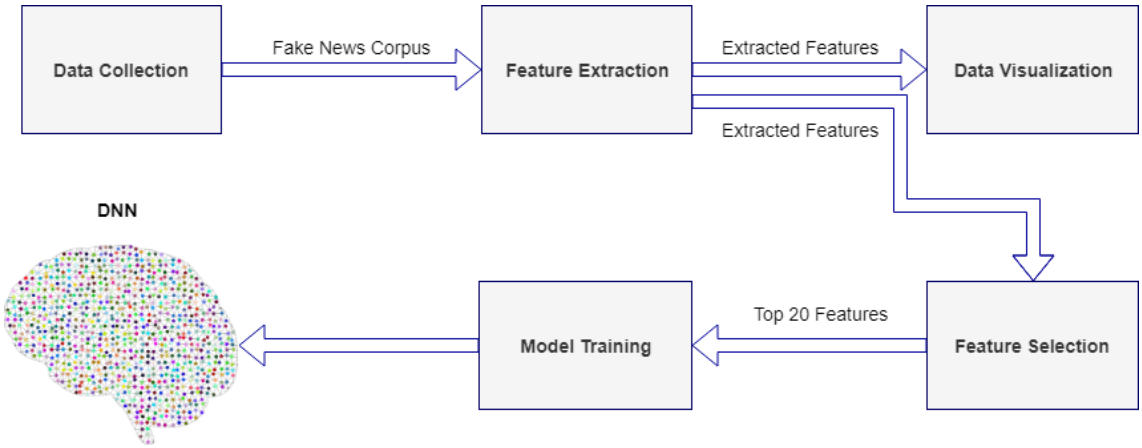


Figure 3.1: An overview of our methodology

not focusing on constructing and training of the deep neural network and we are using it as a black box. However, our job is to ensure the validity of the extracted features and by examining the importance of the features feed the model with the most significant features. Achieving these steps we will increase the accuracy of the model and reduce the possible noise of irrelevant features. In conclusion, we want to guarantee the best possible training for the deep neural network model.

3.2 Data collection

As we mention in the methodology overview, our first step is the data collection. Finding the best dataset for fake news detection is almost impossible due to the nature of the problem. We searched from different sources and we conclude on five datasets mention in Table 3.1.

Index	Reference	Name	Rows	Download
1	[21]	Politifact	240	
2	[21]	Buzzfeed	182	
3	[22]	FakeNewsNet	422	https://github.com/KaiDMML/FakeNewsNet
4		Fake News	20,172	https://www.kaggle.com/c/fake-news/data
5		Fake News Corpus	9,408,908	https://github.com/several27/FakeNewsCorpus

Table 3.1: Data sets we used in the research

Politifact and Buzzfeed datasets used only for evaluating the correctness of our feature extraction process and our deep learning model. Both datasets have an inadequate number of records for training a machine learning model. However, they are very useful for testing our model because they are tagged very precisely by journalist and we are utterly positive for the veracity of each article.

Kaggle is a platform for predictive modeling and analytics competitions in which statisticians and data miners compete to produce the best models for predicting and describing the datasets uploaded by companies and users. The dataset we found in Kaggle contains 20.8k of balanced fake and reliable news articles label using the B.S detector plugin. B.S. Detector searches all links on a given webpage for references to unreliable sources, checking against a manually compiled list of domains. It then provides visual warnings about the presence of questionable links or the browsing of questionable websites. Kaggle dataset contains a quite large amount of reliable and unreliable news articles. An advantage of this dataset is that includes the title and the content of an article and give us the opportunity not only investigating the content of an article but also the combination of title and content. We used this dataset for training and testing the model.

ID: 18
Title: FBI Closes In On Hillary!
Author: The Doc
Text: We now have 5 separate FBI cases probing the Hillary-Bill Clinton inner circle. We now have 5 separate FBI cases probing the Hillary-Bill Clinton inner circle...
Label: 1

(a) Fake Article

ID: 1
Title: FLYNN: Hillary Clinton, Big Woman on Campus - Breitbart
Author: Daniel J. Flynn
Text: Ever get the feeling your life circles the roundabout rather than heads in a straight line toward the intended destination...
Label:true 0

(b) Reliable Article

Figure 3.2: Some random excerpts from the Kaggle dataset

Fake news corpus dataset is the largest dataset we found and contains more than 9 million articles. However, these articles originate from a curated list of 1001 domains collected from opensources.co. The entries are divided into 12 groups: *fake news*, *satire*, *extreme bias*, *conspiracy theory*, *rumor mill*, *state news*, *junk science*, *hate news*, *clickbait*, *political*, and *credible*. In this research we collect only fake articles which are 928,083 and credible articles which are 1,920,139.

3.3 Features Extraction

Feature Extraction is one of the most challenging parts of this research because we tried to extract all the features we found in the literature. Unfortunately, due to the restriction of processing time, we didn't extract features such as n-grams, TF (Term Frequency) and TF-IDF (Term Frequency-Inverse Document Frequency). The extracted features can be classified into three main categories: *Dictionaries, Complexity and Stylistic features.*

3.3.1 Dictionary Features

Dictionary features are based on well-studied word counts that are associated with various psychological processes and basic sentiment analysis. We use several dictionaries such as Laver & Garry, Loughran Mcdonald, Martindale's Regressive Imagery Dictionary (RID) and AFINN dictionary to analyze the tone, the sentiment even the personal concerns of the writer. The Laver & Garry dictionary has been developed to estimates the policy positions of political actors in the United Kingdom by comparing their speeches and written documents to keywords found in the British conservative and Labour manifestos of 1992. Words have been decided semantically based on how they were related to specific content categories as well as empirically based on how specific they were to a specific political party. The dictionary holds 415 words and word patterns stored in 19 categories. Loughran-McDonald dictionary classifies the words into 7 sentiment categories (Negative, Positive, Uncertainty, Litigious, Strong Modal, Weak Modal, Constraining). The English Regressive Imagery Dictionary (RID) is composed of about 3200 words and roots attached to 29 divisions of primary process cognition, 7 categories of secondary process cognition, and 7 categories of emotions. Moreover, it's very important to mention the AFINN dictionary which is a list of English terms manually rated for valence with an integer between -5 (negative) and +5 (positive) by Finn Arup Nielsen. AFINN lexicon looks to plays a significant role in fake news classification.

3.3.2 Complexity Features

Complexity features supply us some truly valuable features using some obscure techniques to extract features that show us the readability index and the vocabulary richness of the text. **Readability index** is the comfort which a reader can understand a written text. In natural language, the readability of text depends on its content. It focuses on the words we choose, and how we put them into sentences and paragraphs for the readers to comprehend. The further we use complex vocabulary and syntax the further our readability score is decreased because the text becomes understandable for fewer people. Some well known methods to calculate readability index are the following: *Flesch, Flesch_kincaid, McLaughlin's SMOG formula, Cole-*

man_liau, Automated Readability Index, Dale–Chall formula, Gunning fog formula, Linsear formula . **Vocabulary Richness** is highly correlated with the word's frequency in a text. There are several techniques to measure vocabulary richness such as *TTR, BrunetsW, HonoreR, SichelS, and Yule*. Most of the techniques take into account the number of unique words in a text.

3.3.3 Stylistic Features

Stylistic features reflect the style of the writer and help you understand the syntax of the text. In this research, we extract more than 90 stylistic features and discovered that some of them play a significant role in fake news classification. Moreover, only stylistic features provide us the ability to study the title's impact in the veracity of an article. Dictionary and complexity features based on extensive texts to provide accurate results and they are not suitable for the title's content. Some examples of stylistic features are *the number of lines, the number of words, the average number of words begin with a capital letter, the radio of digits and the number of stopwords in a text.*

A great subcategory of stylistic features is part of speech tagging. Part of speech tagging helps us to label each word in a corpus with a tag. The tag defines the role of the word in the text, for example, the most commonly appeared tags are N for the noun, V for the verb and A for the adjective. There is a great variate of autonomous POS taggers which reaching up to 98% accuracy and the mistakes are limited. In our Python implementation of features extraction, we are using NLTK pos tagger to calculate the tag of each word. In our JavaScript implementation, we are using a GitHub opensource library which is an optimized version of Eric Brill's POS tagger.

3.4 Data Visualization

An initial understanding of the features can be achieved via data visualization techniques. Data visualization helps the data analyst to extract some quick conclusions

and offer him/her a better awareness of the features. We used the R programming language to produce our visualize data with the help of the ggplot library. We produce two types of graphs: Bar Mean Plot and Box Plot. Bar mean plot displays the central value of a discrete set of numbers. Boxplots can inform you about your outliers and what their values are. It can also tell you if your data is symmetrical, how tightly your data is grouped, and if and how your data is skewed. Figure 3.3 shows you exactly the form of a box plot.

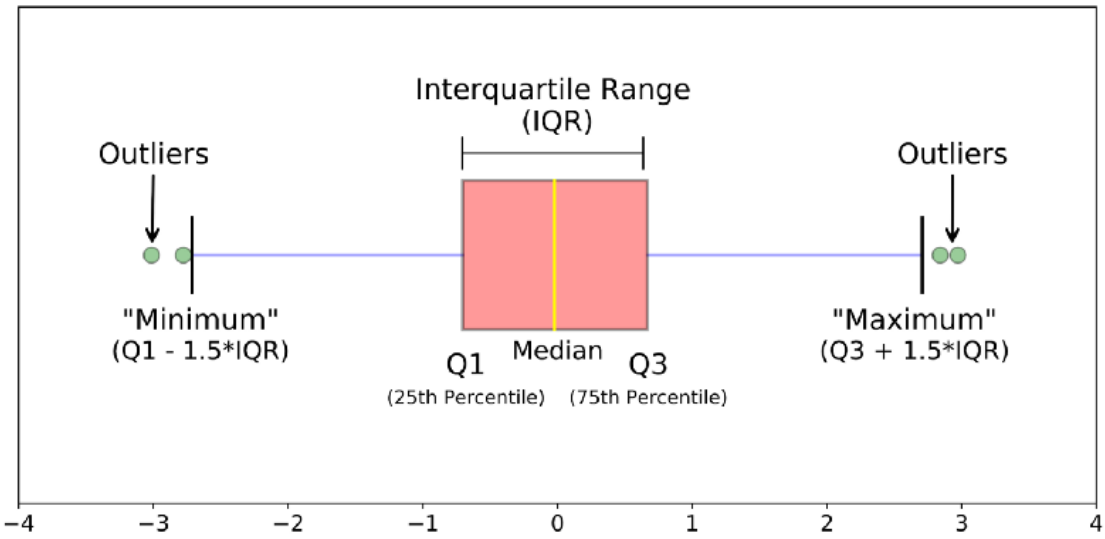
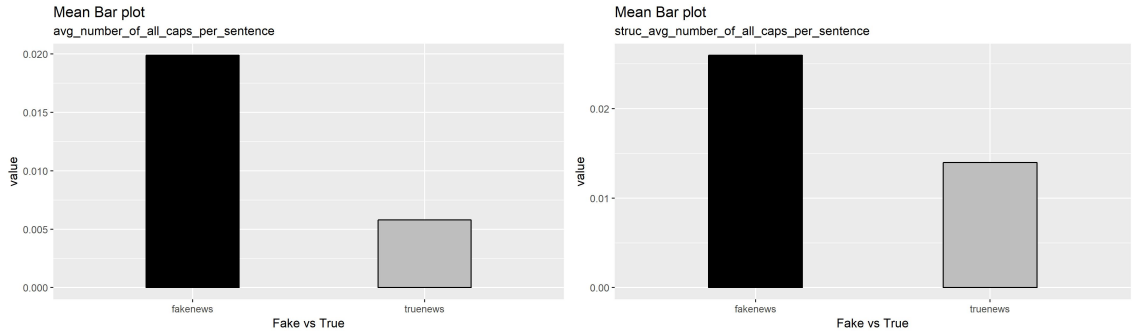


Figure 3.3: Different parts of a boxplot

A feature that stands out is the average number of all capital words in a sentence showing that fake news articles may use more capital words in order to attract the user's attention. Figure 3.4 shows the average number of all capital words in a sentence in both Politifact dataset 3.4a and Kaggle dataset 3.4b.
Both datasets agree that fake articles include a larger average number of uppercase words in a sentence and that can be explained because fake articles may don't have enough arguments to persuade a user and try to impress him/her using capital words. Capital words are connected with shouting and strong modal. Furthermore, capital words usually used in titles to attract users to click an article and steal their attention from their regular activity.

The Feature 3.5 sketches the box plot for AFFIN word score for PolitiFact dataset.



(a) Politifact Mean (b) Kaggle Mean

Figure 3.4: Average Number of all capital words in a sentence

As the graph displays, fake news interquartile range is greater than reliable and lies on the x-axis which means most of the fraudulent data possess AFFIN word score near zero. In contrast, reliable news favor to the positive side of the x-axis and show us that most of the real articles have positive AFFIN word score. Moreover, fake news appears to have more outliers and the distribution of values are sparser than real news which maybe means that real news has homogeneity instead of fake news. Those conclusions can easily be extracted from data visualization and despite the

Figure 3.5: AFFIN word score of Politifact dataset

feature selection process also helps you understand your data, data visualization is

an essential and very important step in our research.

3.5 Feature Selection

Feature selection is the process of finding and selecting the most useful and informative features in a dataset. It is considered to be a crucial step in the machine learning pipeline. Unnecessary features can have side-effects during a model's training, decreasing training speed, decreasing model interpretability, and decreasing generalization performance. The feature selection process is able to i) give us a better understanding of our features and ii) help the DNN to learn from these features as well as make the training faster since this process will result in the removal of sparse features that do not contribute to the model. Regarding the feature selection process, we utilize the following methods:

1. **Features with a high percentage of missing values:** Responsible for finding features with a fraction of missing values above a specified threshold e.g. 60%. Such features are not useful for the classification tasks as they do not carry any information and also can affect the performance of the model by adding unnecessary noise.

2. **Collinear - Highly correlated features:** Highly correlated features may lead to reduced generalization performance on the test set due to high variance and less model interpretability. Such a method is able to detect collinear features based on a specified correlation coefficient value.

3. **Features with zero importance in a tree-based model:** Finds features that have zero importance according to a gradient boosting machine learning model. Such models are tree-based machine learning models, that can find feature importance. In a tree-based model, the features with zero importance are not used to split any nodes, and so they can be removed without affecting model performance.

4. **Features with low importance:** The same feature importance used in the above method, are also utilized here. Features with the lowest importance do

not contribute to specified total importance. This can be done using Principal Components Analysis (PCA) where it is common to keep only the PC needed to retain a certain percentage of the variance, such as 95%. The percentage of total importance accounted for is based on the same idea.

5. **Features with a single unique value:** It is a basic method that detects features with only one unique value. Such features cannot be useful for training a machine learning model because they have zero variance. For example, a tree-based model can never make a separation on a feature with only one value since there are no groups to divide the observations into.

During the feature selection process for fake news, the aforementioned methods were applied on the dataset, individually for articles' titles and content. This separation was performed due to the fact that titles and content are different in nature and combining them may affect the performance of the selection. The first method does not produce results either for content or title, meaning that the dataset analyzed does not contain any missing values. The unique values method detected 97 features to be removed in the content features and 104 features to be removed from the title features Figure 3.6.

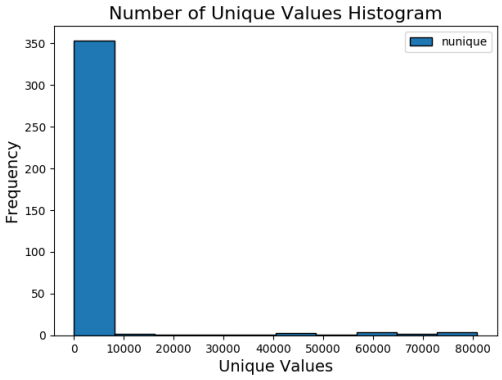


Figure 3.6: Unique Values Histogram for News Articles for both title and content

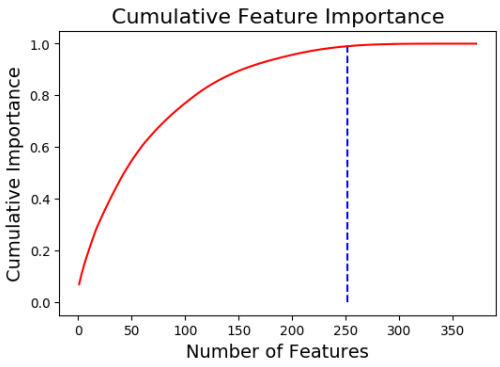


Figure 3.7: Cumulative Feature Importance

The collinearity method detected 13 content features and 4 title features that are highly associated with correlation magnitude greater than 0.975 Figure 3.8. Highly correlated content features include the total number of characters, total number of

words and the total number of words beginning with a lowercase letter.

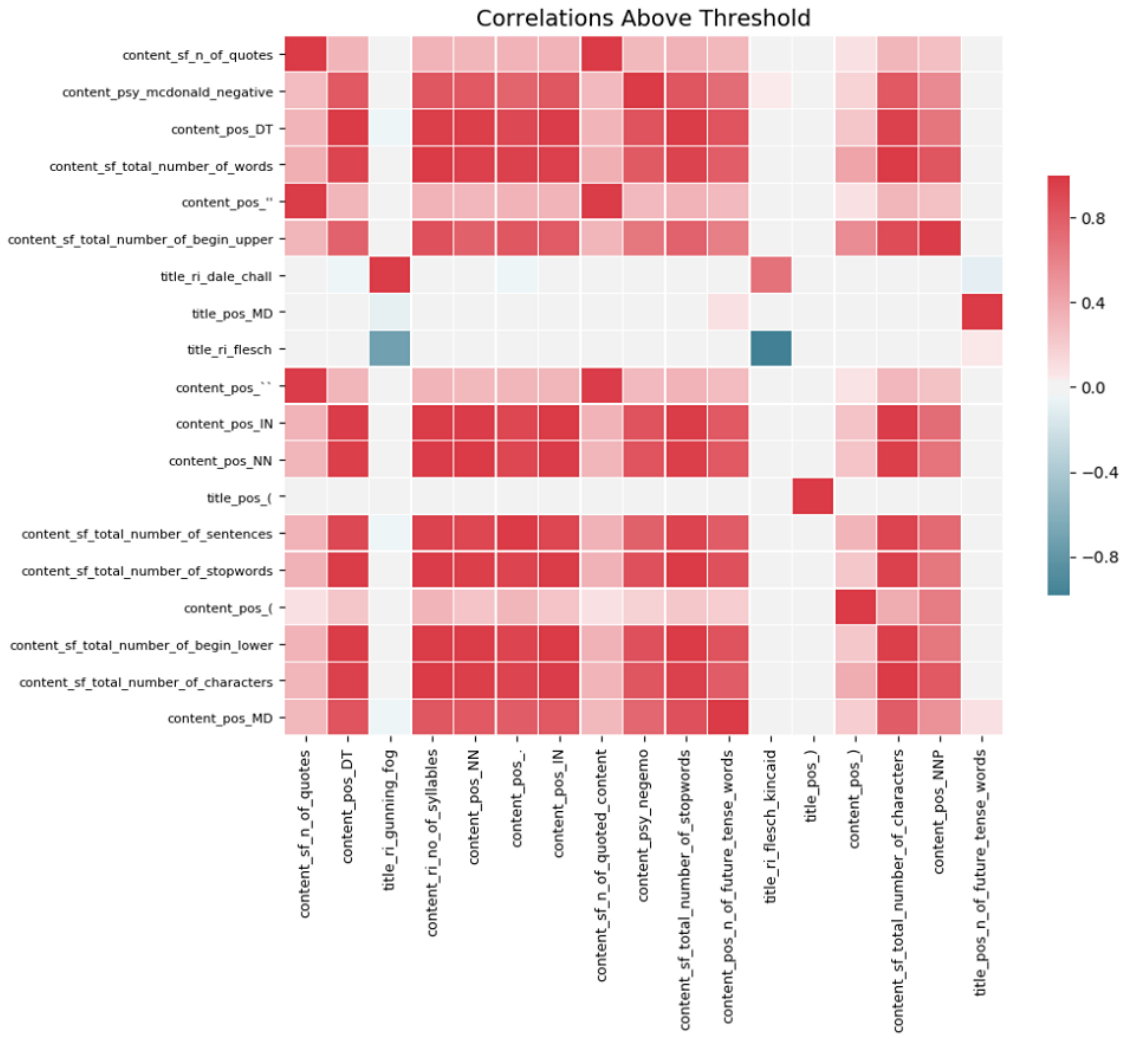


Figure 3.8: Highly correlated content features above the threshold defined as 0.975

From the initial 311 features, 149 are content features and 44 title features were observed to have zero importance using the tree-based model. After the detection of the zero importance features, the method for the detection of the low importance features was applied. The method resulted in 182 content features and 193 content features that do not contribute to a cumulative importance of 0.99. The procedure of filtering the features that do not contribute to the learning of the model and the successful classification of fake and real news, resulted to top 20 more important features Figure 3.9. Moreover, Table 3.2 shows the important score, after Feature Selection process, of each feature from the most important feature which is the total

number of lines to the twentieth most important feature which is the average number of stopwords per sentence in a title.

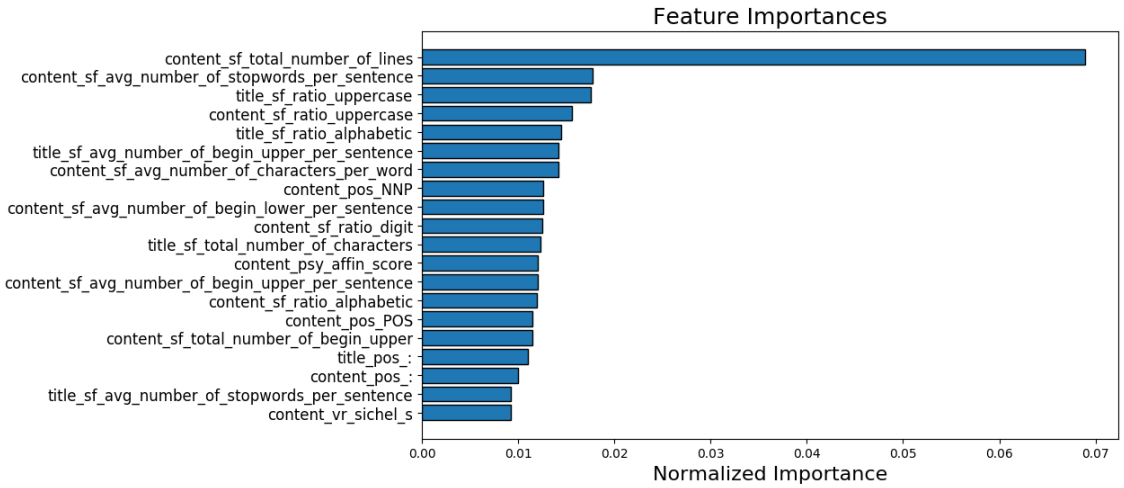


Figure 3.9: Top 20 more Important Features

No.	**Feature**	**Score**	**Type**
1	Total number of lines	0.0693	Content
2	Avg. number of stop-words per sentence	0.0185	Content
3	Ratio of uppercase letters	0.0177	Title
4	Ratio of uppercase letters	0.0152	Content
5	Avg. number of uppercase words per sentence	0.0142	Title
6	Avg. number of characters per word	0.0141	Content
7	Ratio of alphabetic letters	0.0139	Title
8	Number of proper nouns (NP)	0.0128	Content
9	Avg. number of sentences beginning with lowercase letter	0.0126	Content
10	Avg. AFINN sentiment score	0.0123	Content
11	Total number of characters	0.0122	Title
12	Ratio of digits	0.0122	Content

13	Avg. number of sentences beginning with uppercase letter	0.0122	Content
14	Ratio of alphabetic letters	0.0119	Content
15	Number of genitive markers (POS)	0.0116	Content
16	Number of colon or ellipsis	0.0116	Title
17	Total number of words beginning with uppercase letter	0.0113	Content
18	Number of colon or ellipsis	0.0102	Content
19	Avg. number of characters per word	0.0096	Title
20	Avg. number of stop-words per sentence	0.0094	Title

Table 3.2: Table with the 20 most important features as resulted from the feature selection process.

3.6 Deep Neural Network

As we mentioned before the constructor of the deep learning model is not part of this research, but we have to understand how it works from an abstract point of view. Deep learning approaches belong to the broader family of machine learning techniques and there are name came up from the number of layers they used. The deep learning approach was chosen instead of the traditional machine learning approach, due to the massive attention it receives lately and the performance amplification it can achieve, not only for the detection of fake news but the general problem solving using artificial intelligence. Learning can be supervised, semi-supervised or unsupervised but in our case, we are using supervised learning. Supervised learning means that the accuracy of the model is highly correlated with the input we provide to the model. Regarding this statement, our job is crucial in the final outcome and we have to be really precise. Independent of how well the structure of the model is if the model receives amiss input the accuracy of the model will be disappointing. Before feeding the data into the DNN model, any categorical data are transformed into numerical, either via discretization or one-hot encoding, depending on the particulars

of the input. As a result, each data entry is represented as a vector of numerical features. After the pre-processing, the data is used as input to the DNN model via the model's input layer. The next layer is a Batch Normalization Layer which is responsible for the normalization of the activations of the previous layer (input layer) at each batch. Neural networks work better when the input data have zero mean and unit variance, as this enables faster learning and higher overall accuracy. A Batch Normalization Layer can achieve this by transforming and maintaining the mean and variance of its input close to zero. Next, the normalized output enters a set of fully connected layers (dense layers) that form the bottleneck. Such a bottleneck has been shown to result in automatic construction of high-level features. In our implementation, we experimented with multiple architectures, settling in a sequence of 5 layers that consist of 512, 256, 128, 64 and 32 neurons respectively. The final sequence is the one that provided the best results in our task. The units of the network are activated using the hyperbolic tangent activation function (tanh) since it is a better fit when working with standardized numerical data. Finally, in the DNN model's classification layer, one neuron per class is used with the softmax activation function to produce the probability pair of Preal and Pfake, which correspond to the probability of the article being real or fake respectively. Figure 3.10 depicts the structure of the deep neural network as its described in the previous lines.

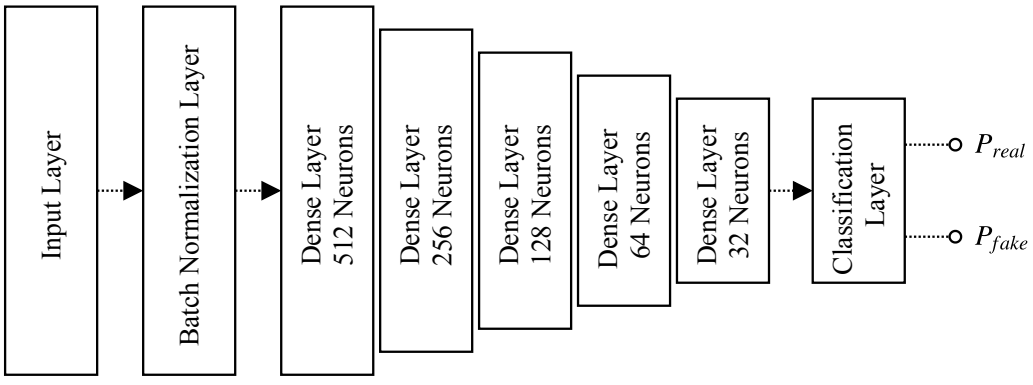


Figure 3.10: Architectural diagram for the deep neural network

Chapter 4

Experiment

Contents

4.1 Classifier Performance

The are various ways to measure the performance of a classifier but, we decide to utilize F1-score. F1-score calculated using precision and recall which are two other metrics. Before explaining those metrics, we have to clarify some essential definitions.

1. **True Positives (TP)**: The total number of accurate predictions that were "positive." In our example, this is the total number of correctly predicting an article as fake.

2. **False Positives (FP)**: The total number of inaccurate predictions that were "positive." In our example, this is the total number of wrongly predicting an article as fake.

3. **True Negative (TN)**: The total number of accurate predictions that were "negative." In our example, this is the total number of correctly predicting article as non-fake.

4. **False Negative (FN)**: The total number of inaccurate predictions that were "negative." In our example, this is the total number of incorrectly predicting article as non-fake.

Explaining those metrics leads to the explanation of Accuracy score which calculated by using equation 4.1. Explaining those metrics leads to the explanation of Accuracy score which calculated by using equation 3.1. Accuracy only works when both possible outcomes (article being fake or not) is equal. For example, if we have a dataset where 5% of their articles are fake, then we could follow a less sophisticated model and have better accuracy score. We could predict every article as non-fake and achieve a 95% accuracy score. The imbalance dataset makes accuracy, not a reliable performance metric to use. The paradox explained is refer as "Accuracy Paradox,"

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{4.1}$$

Now, we have to return to the calculation of F1-score which needs precision and recall scores. Precision refers to the evaluation of our model using positive predictions. In our case, a positive prediction is to classify a news article as fake. The precision score computed using equation 4.2.

$$Precision = \frac{TP}{TP + FP} \tag{4.2}$$

Recall refers to evaluating our data by its performance of the ground truths for positive outcomes. Meaning that we measure how well predicted positive when the results are actually positive. The recall score computed using equation 4.3.

$$Recall = \frac{TP}{TP + FN} \tag{4.3}$$

Finally, F1-score is the weighted average of Precision and Recall and takes into account both false negatives and false positives. Thus, F1-score consider as a better metric than accuracy. F1-score is computed using equation 4.4.

$$F1 - score = \frac{2 * (Precision * Recall)}{Precision + Recall} \tag{4.4}$$

We used those metrics for evaluating our model with different categories features. As we mention in the feature extraction section, we divide our feature into three main categories and those categories into 6 subcategories. We train our model with each subcategory separately and with some combinations and we calculate those metrics for each combination.

4.2 Comparisons

4.2.1 Experiment Setup

One of our main contributions is to compare different kinds of natural language processing characteristics and define which of them are helping the most at fake news classification. Therefore our model is trained with each kind individually and then we joined a few categories to investigate if they have a greater impact on the outcome. All of the experiments run in stratified 10-fold cross-validation to guarantee the validity of our results. Cross-validation shuffles the dataset randomly and splits it into k groups (10 in our case). Then for each group take one as test data and the remaining groups as a training data set. The model is trained using the train set and evaluated using the test set. Retain the evaluation score and discard the model and summarize the skill of the model using the sample of model evaluation scores. Cross-validation technique ensures that our final score is valid and does not come up because of the dataset split. During training, categorical cross-entropy has been used as a loss function and Adam as the optimization function. Îǎo prevents the model from over-fitting. Early stopping is responsible for interrupting the training if the validation loss does not drop for 10consecutive epochs. Moreover, we have to mention that every experiment take place at my personal computer which is composed of 4-cores and 8-threads CPU clocked in 4.2 GHz, 16 Gb RAM, and Windows 10 operating system. As from the software perspective, we used Jupyter Notebook which helps us to write Python programming language. Tensorflow library used for creating the deep neural network and evaluating with each feature category.

4.2.2 Results

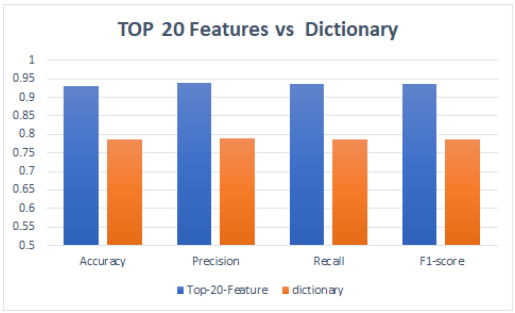
This section presents the results of our study and provides a summary table 4.1 which contains the accuracy, precision, recall, and F1-score for each feature combination we tried. Furthermore, we create a visual comparison of each combination with our Top 20 Features model to give a better understanding of the feature's impact on the outcome. The natural language processing feature that stands out is the POS tags, which as we explained in the feature extraction section, is the tagging of each word with a label. POS tags reach F1-score 0.873, which is the higher of every combination we made but still, is much less than our top 20 features model which owns 0.93 F1-score. As we explained in the feature selection section, we applied five different feature selection methods which lead to the top 20 best features for our dataset.

Category	**Accuracy**	**Precision**	**Recall**	**F1-score**
readability_ index	0.803	0.805	0.803	0.803
vocabulary_ richness	0.769	0.771	0.769	0.769
surface	0.806	0.808	0.806	0.806
pos_tags	**0.873**	**0.875**	**0.873**	**0.873**
psychological	0.798	0.801	0.798	0.798
sentiment	0.768	0.770	0.768	0.767
Stylistic (structural)	0.792	0.794	0.792	0.791
complexity	0.789	0.791	0.789	0.788
dictionary	0.787	0.790	0.787	0.786
Stylistic (structural)l-dictionary	0.802	0.804	0.802	0.801

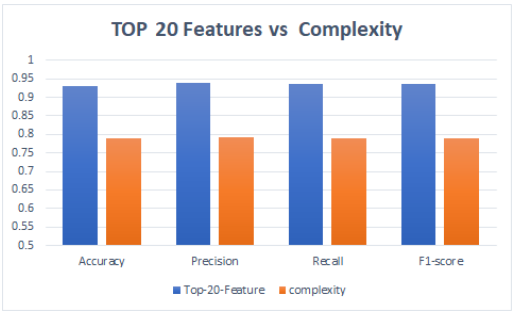
Stylistic (structural)-complexity	0.815	0.817	0.815	0.814
dictionary-complexity	0.816	0.818	0.816	0.816
Stylistic (structural)-dictionary-complexity	0.826	0.828	0.826	0.825
surface-readbility_index	0.833	0.835	0.833	0.832
posTags-readbility_index	0.837	0.839	0.837	0.836
posTags-vocabulary_richness	0.839	0.841	0.839	0.839
posTags-psychological	0.842	0.844	0.842	0.842
Top 20 Features	**0.930**	**0.940**	**0.937**	**0.937**

Table 4.1: Comparison of metrics for each combination

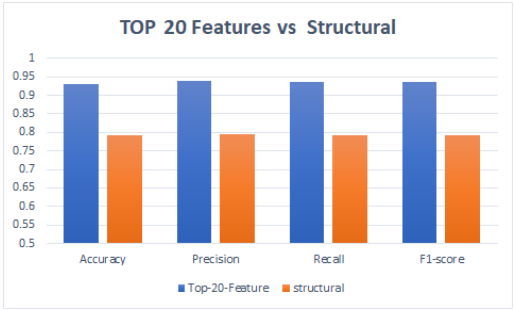
The feature's category with the lowest F1-score is the sentiment, with 0.767. The explanation of this fact maybe is the kind of articles our dataset contains. Most of them are news articles which have a similar tone if they are fake and does not provide any certain emotions. Furthermore, due to our performance limitations, we can not use an expert sentiment analysis tool and that may affect the sentiment score. The following graphs present the visual comparison of every combination with the top 20 features.



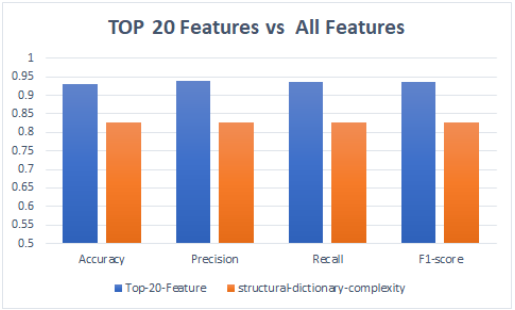
(a)



(b)



(c)



(d)

Figure 4.1: Top 20 Features vs Three Main Categories

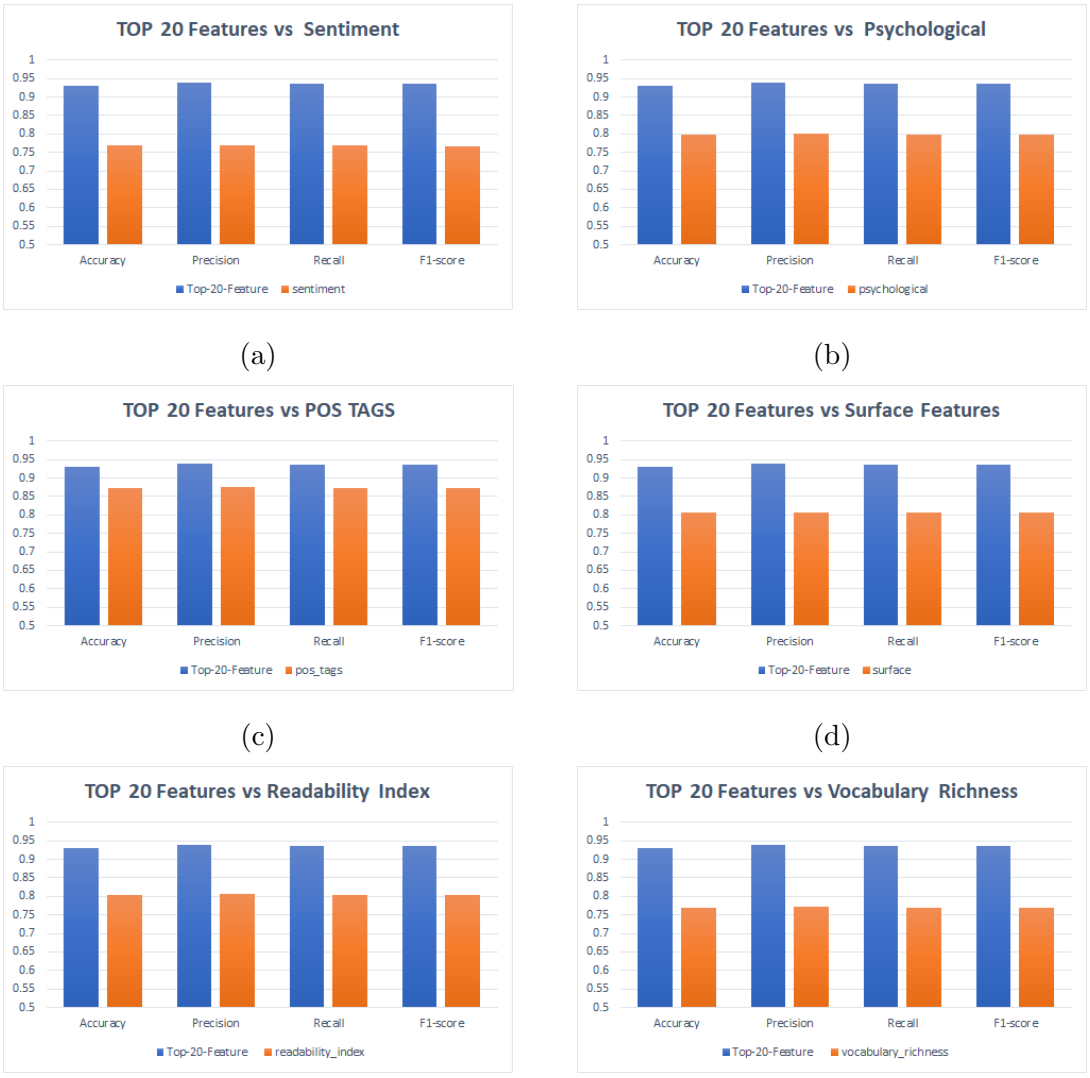
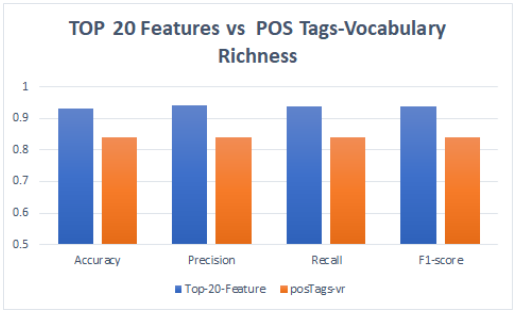
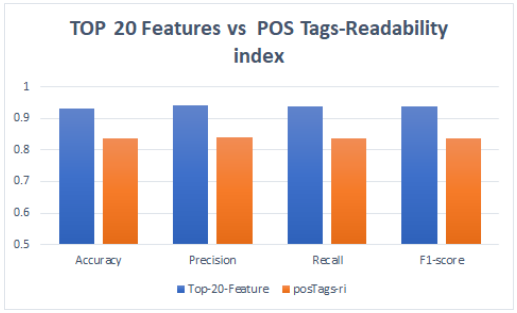


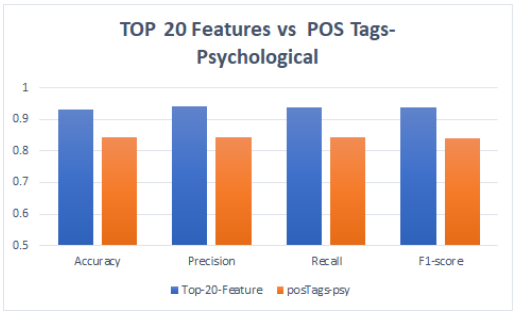
Figure 4.2: Top 20 Features vs 6 main sub-categoreis



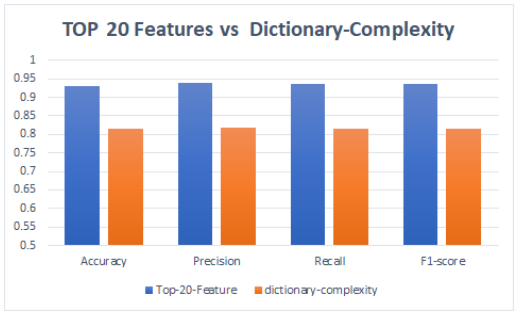
(a)



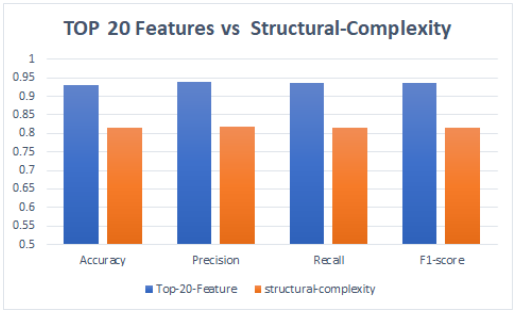
(b)



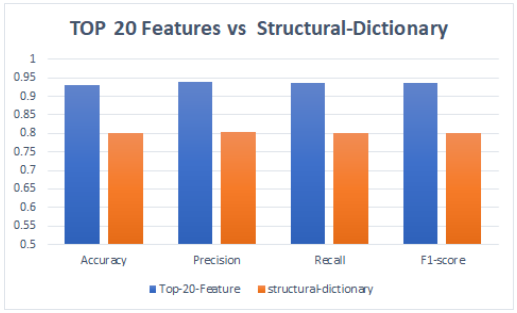
(c)



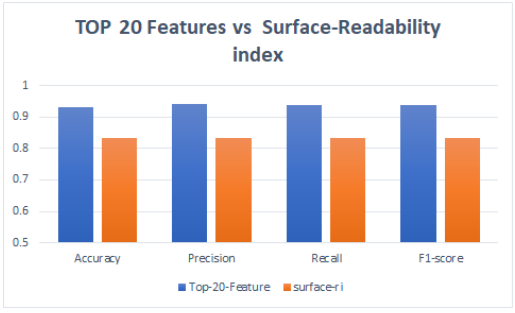
(d)



(e)



(f)



(g)

Figure 4.3: Top 20 Features vs Features Combinations

4.3 TOP 20 Features Optimization

As we mention in the previous section, the top 20 features achieve the highest score of every combination. Our top most prominent features reach **93% F1-score**, which is absolutely an impressive percentage regarding our limitations. Figure 4.4 presents the confusion matrix for those features. The confusion matrix defines specifically our true positives, false positives, true negatives, and false negatives. We evaluate our model using 10000 articles from our dataset (5000 Reliable and 5000 Fake Articles).

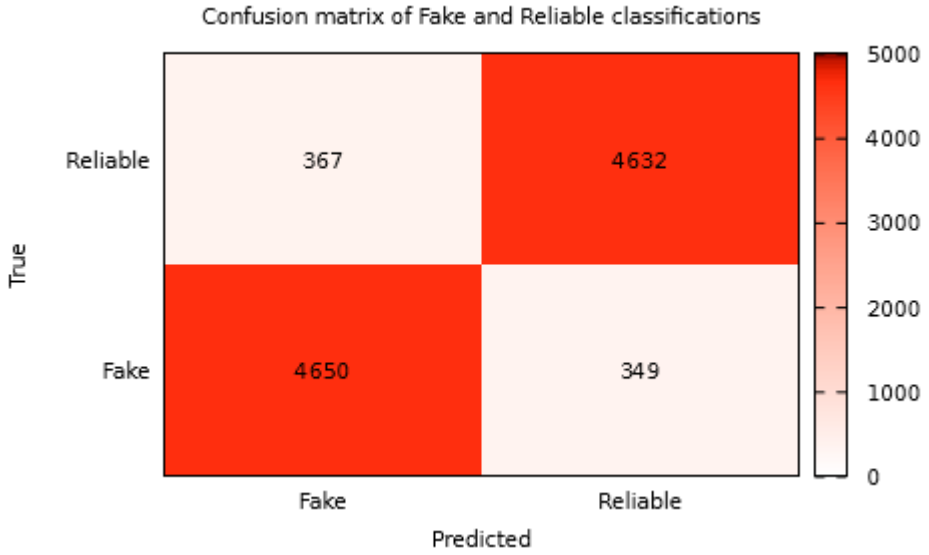


Figure 4.4: Confusion matrix of default classifications

Figure 4.4 shows us that from 5000 fake articles we correctly define the 4650 as fake. However, we define 349 news articles as reliable wrongly. Therefore, our false negative percentage is 7%, and our false positive rate is 93%. Regarding reliable news articles, we define 4632 as reliable and 367 as unreliable. Therefore, our true negative percentage is 7%, and our true positive rate is 93%. Those outcomes extracted with a default threshold value which is 50%. That means that if the model outputs a percentage of 0.51 for an article to be reliable and 0.49 for an article to be fake, it means that we set the article as reliable.

As we mention in the Introduction, our thesis will contribute a largest scale project call Check It. Check It has a very significant requirement, which is the elimina-

tion of TN. Accordingly, we do not desire to classify news as fake when is reliable. Therefore, we evaluate our model in the same dataset, but this time, we had increased the threshold gradually to examine the impact on TN. We conclude with a threshold of 99% in order to vanish TN. Figure 4.5 presents in one confusion matrix the metrics for both thresholds. One the one hand, as we can observe from the ma-

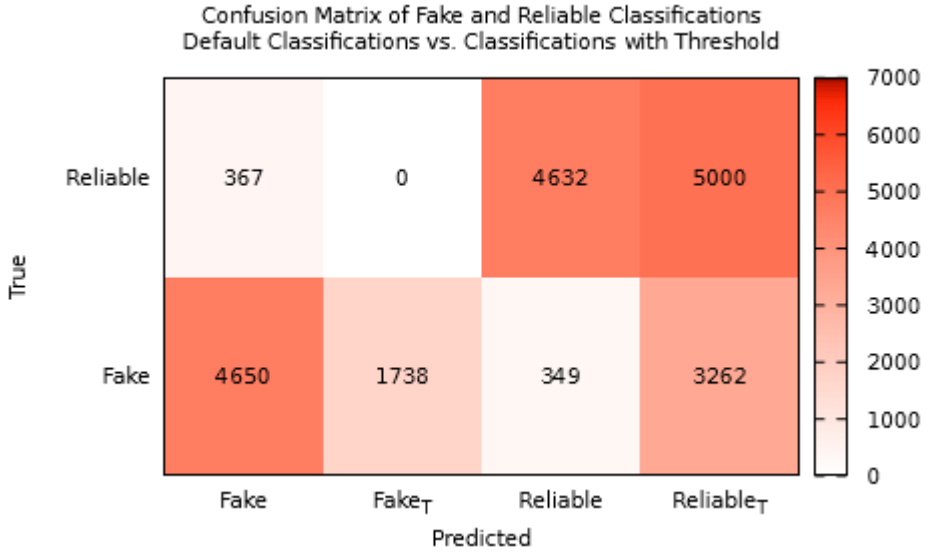


Figure 4.5: Confusion matrix of default classifications andclassifications with threshold

trix, we achieved to vanish any true negatives, because of the level of our threshold. We classify a news article as fake only if we are 99% sure about it. On the other hand, increasing our threshold affects the number of FP and the following graph 4.6 represents the number of FP and TN as a function of our threshold, starting from 0.50 to 0.99 with a step of 0.01. As we observe from the graph, the more we increase the threshold, the more we decrease the number of true negatives and the more we increase the number of false positives. However, the decreasing of true negatives is linear and the increasing of FP is exponential. We have to deal with the trade-off, because of our requirements. In order to evaluate the generalization of our model and the performance with the modified threshold, an additional evaluation was made on several authoritative news articles from sources including The Guardian, New York Times, CNN and BBC. Specifically, 1158 news articles used as input to the DNN model with the adjusted threshold, from which only a single

article was miss-classify as fake.

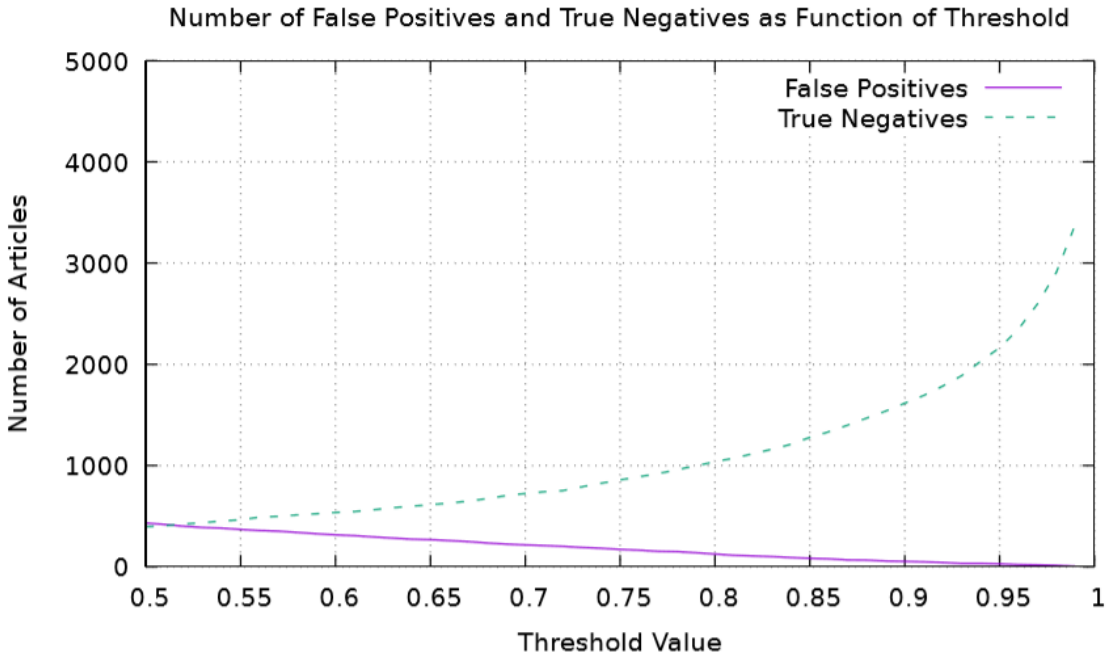


Figure 4.6: Number of False Positives and True Negatives asFunction of Threshold

Chapter 5

Conclusion

Contents

5.1 Conclusion

Taking everything into consideration, the goal of this study is to examine the impact of different natural language processing features and contributes to the fake news classification challenge. We strongly believe that we achieve this objective and we offer some valuable information which will help researches to address this very taff problem. The variety of features we used and the amount of our dataset gives a state-of-the-art outcome. The experiments we cover suggest the use of feature selection methods to extract the best features of your dataset and achieve a higher possible score. Moreover, we figure out that POS tags have a significant impact on our classification and we proofed that low processing features can also achieve state-of-the-art outcomes. The number of features we extracted and the categorization of them into three main domains and six sub-domains aims to organize natural language processing features and help the researches decide a suitable category in their case. We are convinced that natural language processing is one of the most critical keys to overcome this challenging issue called Fake News.

5.2 Future Work

This study approaches the fake news classification challenge with low processing natural language features. However, there is a variety of more complex features such as n-grams and term frequency features which seems to have a significant impact. Those features with the combination of our features will maybe give a higher F1-score.

Moreover, a great future work will be the examination of the evolution of our features during the years. As we all know the defense against fake news are improved, however, the amount of fault news does not seem to decrease and the main reason for that is the need of some people to bias the social opinion. Therefore, fake news articles evolved and try to avoid previous common mistakes which help automated fact-checking systems to detect them. The collection of old and recent datasets and the extraction of their feature will help us to identify the differences. The outcome of research in the evolution of natural language features will be extremely valuable for the community and will help us understand further how can we strike Fake News.

Bibliography

[1] J. Thorne and A. Vlachos, "Automated Fact Checking: Task formulations, methods and future directions", Jun. 2018. arXiv: 1806.07687. [Online]. Available: http://arxiv.org/abs/1806.07687.

[2] H. Allcott and M. Gentzkow, "Social Media and Fake News in the 2016 Election", *Journal of Economic Perspectives*, vol. 31, no. 2, pp. 211–236, May 2017, ISSN: 0895-3309. DOI: 10.1257/jep.31.2.211. [Online]. Available: http://pubs.aeaweb.org/doi/10.1257/jep.31.2.211.

[3] B. D. Horne and S. Adali, "This Just In: Fake News Packs a Lot in Title, Uses Simpler, Repetitive Content in Text Body, More Similar to Satire than Real News", 2017. arXiv: 1703.09398. [Online]. Available: http://arxiv.org/abs/1703.09398.

[4] L. Wang, Y. Wang, G. De Melo, and G. Weikum, "Five shades of untruth: Finer-grained classification of fake news", *Proceedings of the 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2018*, pp. 593–594, 2018. DOI: 10.1109/ASONAM.2018.8508256.

[5] Y. Goldberg, "A primer on neural network models for natural language processing", *Journal of Artificial Intelligence Research*, vol. 57, pp. 345–420, 2016, ISSN: 10769757. arXiv: arXiv:1510.00726v1.

[6] D. Weiss, C. Alberti, M. Collins, and S. Petrov, "Structured Training for Neural Network Transition-Based Parsing", no. 2012, pp. 323–333, 2015. arXiv: 1506.06158. [Online]. Available: http://arxiv.org/abs/1506.06158.

[7] Y. Chen, N. J. Conroy, and V. L. Rubin, "Misleading Online Content", in *Proceedings of the 2015 ACM on Workshop on Multimodal Deception Detection*

- *WMDD '15*, New York, New York, USA: ACM Press, 2015, pp. 15–19, ISBN: 9781450339872. DOI: 10.1145/2823465.2823467. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2823465.2823467.

[8] G. Durrett and D. Klein, "Neural CRF Parsing", pp. 302–312, 2015. arXiv: 1507.03641. [Online]. Available: http://arxiv.org/abs/1507.03641.

[9] W. Pei, T. Ge, and B. Chang, "An Effective Neural Network Model for Graph-based Dependency Parsing", pp. 313–322, 2015. DOI: 10.3115/v1/p15-1031.

[10] R. Johnson and T. Zhang, "Effective Use of Word Order for Text Categorization with Convolutional Neural Networks", no. 2011, 2014. arXiv: 1412.1058. [Online]. Available: http://arxiv.org/abs/1412.1058.

[11] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A Convolutional Neural Network for Modelling Sentences", pp. 655–665, 2014. arXiv: 1404.2188. [Online]. Available: http://arxiv.org/abs/1404.2188.

[12] L. Wang, Y. Wang, G. De Melo, and G. Weikum, "Five shades of untruth: Finer-grained classification of fake news", *Proceedings of the 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2018*, pp. 593–594, 2018. DOI: 10.1109/ASONAM.2018.8508256.

[13] C. D. Santos and B. Zadrozny, "Learning Character-level Representations for Part-of-Speech Tagging", *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, vol. 32, no. 2011, pp. 1818–1826, 2014. [Online]. Available: http://jmlr.org/proceedings/papers/v32/santos14.pdf.

[14] C. G. A. Kuchler, "Learning Task-Dependent Distributed Representations by Backpropagation Through Structure", 1996.

[15] J. L. Elman, "Finding structure in time", *Cognitive Science*, vol. 14, no. 2, pp. 179–211, 1990, ISSN: 03640213. DOI: 10.1016/0364-0213(90)90002-E.

[16] Richard Socher John Bauer Christopher D. Manning Andrew Y. Ng, "Parsing with Compositional Vector Grammars", *Journal of the American Chemical Society*, vol. 80, no. 19, pp. 5080–5083, 2013, ISSN: 15205126. DOI: 10.1021/ja01552a021.

[17] H. Ahmed, I. Traore, and S. Saad, "Detection of Online Fake News Using N-Gram Analysis and Machine Learning Techniques", in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10618 LNCS, Springer, Cham, Oct. 2017, pp. 127–138, ISBN: 9783319691541. DOI: 10.1007/978-3-319-69155-8_9. [Online]. Available: http://link.springer.com/10.1007/978-3-319-69155-8%7B%5C_%7D9.

[18] V. Pérez-Rosas, B. Kleinberg, A. Lefevre, and R. Mihalcea, "Automatic Detection of Fake News", Aug. 2017. arXiv: 1708.07104. [Online]. Available: http://arxiv.org/abs/1708.07104.

[19] N. J. Conroy, V. L. Rubin, and Y. Chen, "Automatic deception detection: Methods for finding fake news", *Proceedings of the Association for Information Science and Technology*, vol. 52, no. 1, pp. 1–4, 2015, ISSN: 23739231. DOI: 10.1002/pra2.2015.145052010082. [Online]. Available: http://doi.wiley.com/10.1002/pra2.2015.145052010082.

[20] V. Rubin, N. Conroy, Y. Chen, and S. Cornwell, "Fake News or Truth? Using Satirical Cues to Detect Potentially Misleading News", *Proceedings of the Second Workshop on Computational Approaches to Deception Detection*, 2016. [Online]. Available: http://www.academia.edu/24790089/Fake%7B%5C_%7DNews%7B%5C_%7Dor%7B%5C_%7DTruth%7B%5C_%7DUsing%7B%5C_%7DSatirical%7B%5C_%7DCues%7B%5C_%7Dto%7B%5C_%7DDetect%7B%5C_%7DPotentially%7B%5C_%7DMisleading%7B%5C_%7DNews.

[21] H. Rashkin, E. Choi, J. Y. Jang, S. Volkova, and Y. Choi, "Truth of Varying Shades: Analyzing Language in Fake News and Political Fact-Checking", in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2017, pp. 2931–2937. DOI: 10.18653/v1/D17-1317. [Online]. Available: http://aclweb.org/anthology/D17-1317.

[22] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, "Fake News Detection on Social Media", *ACM SIGKDD Explorations Newsletter*, vol. 19, no. 1, pp. 22–

36, Sep. 2017, ISSN: 19310145. DOI: 10.1145/3137597.3137600. [Online]. Available: http://dl.acm.org/citation.cfm?doid=3137597.3137600.

Appendices

Appendix A

A-1 Dictionary Features

Feature	Definition	Examples
Loughran Mcdonald Dictionary		
LM_NEGATIVE	Loughran Mcdonald's words show negative tone	abducates, burden, careless
LM_POSITIVE	Loughran Mcdonald's words show positive tone	advancement, dream, innovator
LM_UNCERTAINTY	Loughran Mcdonald's words show uncertainty	approximate, doubted, speculate
LM_LITIGIOUS	Loughran Mcdonald's words show litigious tone	absolved, crime, executory
LM_CONSTRAINING	Loughran Mcdonald's words show constraining tone	confines, forbids, unavailability
LM_SUPERFLUOUS	Loughran Mcdonald unnecessary words	assimilate, theses, whilst
LM_INTERESTING	Loughran Mcdonald interesting words	extraordinary, rabbi, toxic
LM_MODAL WORDS STRONG	Loughran Mcdonald's words show strong modal	always, must, never
LM_INTERESTING	Loughran Mcdonald interesting words	extraordinary, rabbi, toxic

Laver Garry Dictionary		
LG_CULTURE-HIGH	Laver Garry's words show high culture	artistic, music, theatre
LG_CULTURE-POPULAR	Laver Garry's words show popular culture	media
LG_CULTURE-SPORT	Laver Garry's words show sport culture	angler, civil war, people
LG_ECONOMY	Laver Garry's words related with economy	accounting, earn, loan
LG_ENVIRONMENT	Laver Garry's words related with environment	green, planet, recycle
LG_GROUPS_ETHNIC	Laver Garry's words related with ethnic groups	Asian, race, ethnic
LG_GROUPS_WOMEN	Laver Garry's words related with women	girls, woman, women
LG_INSTITUTIONS_CONSERVATIVE	Laver Garry's words related with conservative institutions	authority, inspect, rule
LG_INSTITUTIONS_NEUTRAL	Laver Garry's words related with neutral institutions	chair, scheme, voting
LG_LAW_and_ORDER	Laver Garry's words related with law and order	police, punish, victim
LG_RUDAL	Laver Garry's words related with countryside	farm, forest, village
LG_VALUES_CONSERVATIVE	Laver Garry's words with conservative values	glories, past, proud
LG_VALUES_LIBERAL	Laver Garry's words with liberal values	cruel, rights, sex
RID Primary Needs		

RID_ORALITY	RID's words show orality	belly, cook, eat
RID_ANALITY	RID's words show anality	anal, dirt, fart
RID_SEX	RID's words related with sex	lover, kiss, naked
RID Primary Sensation		
RID_TOUCH	RID's words related with touching	contact, sting, touch
RID_TASTE	RID's words related with tasting	flavor, savor, spicy
RID_ODOR	RID's words related with smelling	aroma, nose, sniff
RID_GEN_SENSATION	RID's words related with general sensation	awareness, charm, fair
RID_SOUND	RID's words related with sounds	bell, ear, music
RID_VISION	RID's words related with vision	bright, gray, spy
RID_COLD	RID's words related with cold	Alaska, ice, polar
RID_HARD	RID's words related with feels hard in touching	crispy, metal, rock
RID_SOFT	RID's words related with feels soft in touching	feather, lace, velvet
RID Primary DEFENSIVE_SYMBOL		
RID_PASSIVITY	RID's words related with passivity	bed, dead, safe
RID_VOYAGE	RID's words related with trips	journey, nomad, travel
RID_RANDOM MOVEMENT	RID's words related with random movements	jerk, spin, wave

RID_DIFFUSION	RID's words related with diffusion	fog, mist, shadow
RID_CHAOS	RID's words related with chaos	char, discord, random
RID_CHAOS	RID's words related with chaos	char, discord, random
RID Primary Regressive Cognition		
RID_UNKNOW	RID's words shows unknown feelings	secret, strange, unknown
RID_TIMELESSNES	RID's words related with infinity time	eternal, forever, immortal
RID_COUNSCIOUS	RID's words shows consciousness alteration	dream, sleep, wake
RID_BRINK-PASSAGE	RID's words shows brink passage	road, wall, door
RID_NARCISSISM	RID's words shows narcissism	eye, heart, hand
RID_CONCRETENESS	RID's words shows something specific	here, tip, wide
RID Primary Icarian Imagery		
RID_ASCEND	RID's words shows that something ascend	climb, fly, wing
RID_HEIGHT	RID's words related with height	bird, hill, sky
RID_DESCENT	RID's words shows that something descent	dig, drop, fall
RID_DEPTH	RID's words related with depth	cave, hole, tunnel
RID_FIRE	RID's words related with fire	solar, coal, warm

RID_WATER	RID's words related with water	ocean, sea, pool
RID Secondary feeling		
RID_ABSTRACT_ TOUGHT	RID's words related with abstraction	know, may, thought
RID_SOCIAL_ BEHAVIOR	RID's words related with social behavior	ask, tell, call
RID_INSTRU_ BEHAVIOR	RID's words related with instrumental behavior	make, find, work
RID_RESTRAINT	RID's words related with restraint behavior	must, stop, bind
RID_ORDER	RID's words related with order(form)	measure, array, system
RID_TEMPORAL_ REPERE	RID's words related with temporal references	when, now, then
RID_MORAL_ IMPERATIVE	RID's words related with moral imperatives	should, right, virtue
RID Emotions		
RID_POSITIVE_ AFFECT	RID's words related with positive emotions	cheerful, enjoy, fun
RID_ANXIETY	RID's words related with anxiety emotions	avoid, horror, shy
RID_SADNESS	RID's words related with sad emotions	hopeless, pain, tragic
RID_AFFECTION	RID's words related with affection	bride, like, mercy
RID_EXPRESSIVE_ BEH	RID's words related with expressive behavior	dance, sing, art
RID_GLORY	RID's words related with glory	elite, kingdom, royal

RID_GLORY	RID's words related with glory	elite, kingdom, royal
Other Dictionaries		
Uncertainty_words	Words that shows uncertainty	assume, could, maybe
Bad words		bastards, tits, porn
Common words	Words that commonly used	the, of, come
Emotional tone words		angry, happy, tolerant
Emotional words		bored, helpless, hurt
Negative words		abandon, abuse, concern
Positive words		boost, easy, enjoys
Hu_Liu Negative words	Hu Liu negative words	abrade, bankrupt, cataclysm
Hu_Liu Positive words	Hu Liu positive words	accurate, brighten, fascination
Litigious words		appeal, dockets, indict
Strong modal words		best, never, will
Weak modal words		could, depend, may
Slang words	Slang words are very informal language	hello, 2mr, 4give
AFINN Dictionary		
AFINN score	The AFINN lexicon is a list of English terms manually rated for valence with an integer between -5 (negative) and +5 (positive) by Finn Årup Nielsen	abuses: -3, amazing: 4, avoid: -1

Table 1: Dictionary Features

A-2 Complexity Features

Feature	Definition
Readability Index	
Flesch reading ease	$206.835 - 1.015\left(\frac{\text{total words}}{\text{total sentences}}\right) - 84.6\left(\frac{\text{total syllables}}{\text{total words}}\right)$
Flesch–Kincaid	$0.39\left(\frac{\text{total words}}{\text{total sentences}}\right) + 11.8\left(\frac{\text{total syllables}}{\text{total words}}\right) - 15.59$
SMOG	$\text{grade} = 1.0430\sqrt{\text{number of polysyllables} \times \frac{30}{\text{number of sentences}}} + 3.1291$
Automated readability index	$0.39\left(\frac{\text{total words}}{\text{total sentences}}\right) + 11.8\left(\frac{\text{total syllables}}{\text{total words}}\right) - 15.59$
Dale-Chall	$0.1579\left(\frac{\text{difficult words}}{\text{words}} \times 100\right) + 0.0496\left(\frac{\text{words}}{\text{sentences}}\right)$
Coleman–Liau	$CLI = 0.0588L - 0.296S - 15.8$ L = Letters / Words * 100 = 639 / 119 * 100 = 537 S = Sentences / Words * 100 = 5 / 119 * 100 = 4.20
Gunning fog	$0.4\left[\left(\frac{\text{words}}{\text{sentences}}\right) + 100\left(\frac{\text{complex words}}{\text{words}}\right)\right]$
Vocabulary Richness	
Yule K	The larger the number the less rich is the vocabulary of the text
TTR	The larger the number the richer is the vocabulary of the text
Brunets	The larger the number the richer is the vocabulary of the text
Sichel	The larger the number the richer is the vocabulary of the text

Table 2: Complexity Features

A-3 Stylistic Features

Feature	**Meaning**
Part Of Speech Tags	
CC	Coordinating conjunction
CD	Cardinal digit
DT	Determiner
EX	Existential there (like: "there is" ... think of it like "there exists")
FW	Foreign word
IN	preposition/subordinating conjunction
JJ	adjective 'big'
JJR	adjective, comparative 'bigger'
JJS	adjective, superlative 'biggest'
LS	list marker 1)
MD	modal could, will
NN	noun, singular 'desk'
NNS	noun plural 'desks'
NNP	proper noun, singular 'Harrison'
NNPS	proper noun, plural 'Americans'
PDT	predeterminer 'all the kids
POS	possessive ending parent's
PRP	personal pronoun I, he, she
PRP$	possessive pronoun my, his, hers
MD	modal could, will
RB	adverb very, silently
RBR	adverb, comparative better
RBS	adverb, superlative best
RP	particle give up
TO,	to go 'to' the store.
UH	interjection, errrrrrrrm

VB	verb, base form take
VBD	verb, past tense took
VBG	verb, gerund/present participle taking
VBN	verb, past participle taken
VBP	verb, sing. present, non-3d take
VBZ	verb, 3rd person sing. present takes
WDT	wh-determiner which
WP	wh-pronoun who, what
WP$	possessive wh-pronoun whose
WRB	wh-abverb where, when

Table 3: Part Of Speech Features

Feature	**Meaning**
Structural	
total_number_of_sentences	
total_number_of_words	
total_number_of_characters	
total_number_of_begin_upper	Words with first capital letter
total_number_of_begin_lower	Words with first lower-case letter
total_number_of_all_caps	Word with all capital letters
total_number_of_stopwords	
total_number_of_lines	
number_of_I_pronouns	
number_of_we_pronouns	
number_of_you_pronouns	
number_of_he_she_pronouns	
number_of_exclamation_marks	

number_of_quotes	
number_of_happax_legomena	
number_of_happax_dislegomena	
has_quoted_content	
ratio_alphabetic	
ratio_uppercase	
ratio_digit	
avg_number_of_characters_per_word	
avg_number_of_words_per_sentence	
avg_number_of_characters_per_sentence	
avg_number_of_begin_upper_per_sentence	
avg_number_of_all_caps_per_sentence	
avg_number_of_begin_lower_per_sentence	
avg_number_of_stopwords_per_sentence	

Table 4: Structural Features