Fake news detection in social media

Kasseropoulos Dimitrios-Panagiotis

SID: 3308190011

SCHOOL OF SCIENCE & TECHNOLOGY

A thesis submitted for the degree of

Master of Science (MSc) in Data Science

JANUARY, 2021

THESSALONIKI - GREECE

Fake news detection in social media

Kasseropoulos Dimitrios-Panagiotis

SID: 3308190011

Supervisor:	Assoc. Prof. Christos Tjortjis
Supervising Committee Members:	Prof. Panagiotis Bozanis
	Dr. Leonidas Akritidis

SCHOOL OF SCIENCE & TECHNOLOGY

A thesis submitted for the degree of

Master of Science (MSc) in Data Science

JANUARY, 2021

THESSALONIKI - GREECE

Abstract

This dissertation was written as a part of the MSc in Data Science at the International Hellenic University.

The easy propagation and access to information in the web has the potential to become a serious issue when it comes to disinformation. The term “fake news” is describing the intentional propagation of fake news with the intent to mislead and harm the public, and has gained more attention since the U.S elections of 2016. Recent studies have used machine learning techniques to tackle it. This thesis reviews the style-based machine learning approach which relies on the textual information of news, such as the manually extraction of lexical features from the text (e.g. part of speech counts), and testing the performance of both classic and non-classic (artificial neural networks) algorithms. We have managed to find a subset of best performing linguistic features, using information-based metrics, which also tend to agree with the already existing literature. Also, we combined the Name Entity Recognition (NER) functionality of spacy’s library with the FP Growth algorithm to gain a deeper perspective of the name entities used in the two classes. Both methods reinforce the claim that fake and real news have very small differences in their content, setting limitations to style-based methods. The final results showed that convolutional neural network had the best accuracy outperforming SVM by almost 2%.

Acknowledgements

I would first like to thank my supervisor Assoc. Professor Christo Tjortji for his continuous guidance and feedback during the development of this thesis. His encouragement and non-interventionist attitude helped me to study my topic from different approaches and perspectives, gaining a spherical understanding of both the problem and the techniques used to solve it. I also have to express my gratitude to my family and friends for their unconditionally support and encouragement throughout my studies.

Kasseropoulos Dimitris Panagiotis

01/01/2021

Contents

1 Introduction

Social media have become an important part of our everyday lives, changing the way we communicate and interact with other people and the world in general. One of the aspects that could not stay unaffected is the way we receive and publish information. Easy access to high speed internet, tools that made a website deployment easier than previously and the popularity of many microblog websites, like Facebook, Instagram, Linkedin etc., made publishing and receiving news information accessible to everyone, anytime. Their adoption from society and their impact on it, can be noticed by two recent research works which showed that 62% percent of adults in U.S get news on social media whilst the other study showed that most of the young people develop their political identities via Social media platforms (Gottfried J. & Shearer E., 2017; Woolley S. C., & Philip H., 2017).

Although the large number of informative online sources increased the variety of aspects available for one topic, many of them have low quality, making filtering a necessity. There are a lot of reasons that could lead to lowering quality, such as the competition for web traffic leading to articles that are more appealing to their readers, and the lack of experience of the authors that tend to diverge from raw fact reporting. These conditions have also created a trend/danger called fake news.

Most of fake news categories, like click bait and satire, used to have an entertaining or, in the worst case, annoying character, but since 2016 and the presidential elections in the United States, fake news and their impact gained global attention (Kiesel J., Mestre M., et al., 2019). Now fake news is considered a threat to democracy and journalism (Zafarani R., et al., 2019).

Social media gave the opportunity to news to have an alternative way of reaching the public rapidly, but at the same time, they also benefit disinformation propagation. Studies have shown that fake, extremely one-side (hyperpartisan) and emotional news tend to spread far more rapidly than traditional news (Wu L., et al., 2017; Potthast M., et al., 2017)

Some factors that benefit the flourish of disinformation in the web are the difficulty of accessing trustworthy information and the lack of trust in the traditional informative means. A fake story can have serious impacts on society, if a significant volume of people believes it. Examples like the “Pizzagate” incident, a conspiracy theory that led a U.S citizen to commit an armed attack to “Comet Ping Pong” pizzeria, and the false claim of an attack on the White House which wiped out $136 billion in the stock market within two minutes, emphasizes the importance of the problem and the need for an effective solution (Shahsavari S., et al., 2020; Wu L., et al., 2017; Potthast M., et al., 2017). Finally, during the COVID – 19 era, fake news is on the rise, making the work of health professionals more difficult in an already critical situation, endangering this way public (Orso D., et al., 2020).

The rapidly propagation of information in the web makes the quick detection of fake news crucial. That is why the new technologies of machine learning and artificial intelligence have been utilized widely the last years to tackle this problem (Zafarani R., Zhou X., et al., 2019).

The scope of this thesis is to describe further: how previous studies define “fake news”, the categories of them, the proposed detection techniques, the challenges of the problem and finally, to propose our own model that classifies correctly an article as real or fake. At the last section we will discuss the results and the additional future work can be done.

2 Literature review

2.1 Fake news definition

Defining fake news is the first task of this thesis. In order to end up with a representative definition we used some recent researches on the topic.

In most of the fake news studies the authors either define or use an already existing definition for fake news. For example, in both studies of Khan, J. Y., et al., and Shu, K., et al., the authors agree on the definition "*Fake news is a news article that is intentionally and verifiably false"* (Khan J. Y., et al., 2019; Shu K., et al., 2017). Other definitions also exist in the literature (see table 1) but most of them agree upon the significance of **intention** and **validity** as a criterion for the article.

Fake news is a news article that is intentionally and verifiably false.	Shu K., et al., 2017; Hunt A. & Entzkow M., 2017
A type of yellow journalism or propaganda that consists of deliberate misinformation or hoaxes spread via traditional print and broadcast news media or online social media	Leonhardt D. & Thompson S. A., 2017
Fake news stories are stories that are known to be false and are from well-known fake news websites that are intentionally trying to spread misinformation	Horne B. D., & Adali, S., 2017
The online publication of intentionally or knowingly false statements of facts, and it has recently dominated current social media platforms.	Klein, D., & Wueller, J, 2017
Information presented as a news story that is factually incorrect and designed to deceive the consumer into believing it is true.	Golbeck J., et al., 2018
A news article or message published and propagated through media, carrying *false* information regardless the means and motives behind it.	Sharma K., et al., 2019
Misleading news stories that come from non-reputable sources	Gilda S., 2017

Table 1: Definitions of fake news in previous studies

2.2 Fake news characteristics

In their research, Cacioppo J. T., & Petty, R. E., stated the Elaboration Likelihood model of Persuasion (ELP), arguing that people are persuaded either by a central route, meaning that all the arguments are examined, or by the peripheral route, which is a more naïve, oversimplified method that focuses only in the key concepts of a claim. The main difference is

that the first method requires more energy than the second, which primarily is more frequent in social media, since studies have shown that most of the articles shared or commented are never read (Wang, L. X., Ramachandran, A., & Chaintreau, A., 2016).

Based on the ELP theorem and Petty's and Cacioppo's findings, the authors Khan, J. Y., et al., argue that fake news target the peripheral route and this is why their titles contain the most important claims about people and events.

Referring to the titles' role in fake news, they mostly serve as the main mechanism of information propagation where the body just repeats and enhances the title claims without providing any additional arguments. That is why they tend to be longer and contain simpler language, capital words, more proper nouns, many verb phrases and name entities but fewer nouns and stop words. Also titles try to compress as much information as they can in order to be more appealing to the reader, without however having significant similarity with click bait titles (Horne, B. D., & Adali S., 2017).

Regarding to the characteristics of the content, it seems that fake articles are a lot smaller in length. More specifically, they use fewer technical words, smaller words, fewer punctuation, fewer quotes and more lexical redundancy. Also, in a linguistic level, they use simpler language resulting in fewer analytic words, more personal pronouns, fewer nouns and more adverbs (Horne, B. D., & Adali, S., 2017). On the other hand, there are some studies that disagree with the length as a differentiation characteristic of the two classes (Rubin, V. L., et al., 2015).

Finally, a good indicator is the emotional response the article tries to achieve. Strong emotional words and phrases draw more attention and propagate faster than real, neutral, facts. (Ruchansky, N., Seo, S., & Liu, Y., 2017)

2.3 Fake news categories

Most of the researches split fake news into categories based on the two basic characteristics extracted from the definition of fake news: intention and quality of information. A very good optical representation of these two variables was given by Rashkin, H., et al., in their research (see figure 1.)

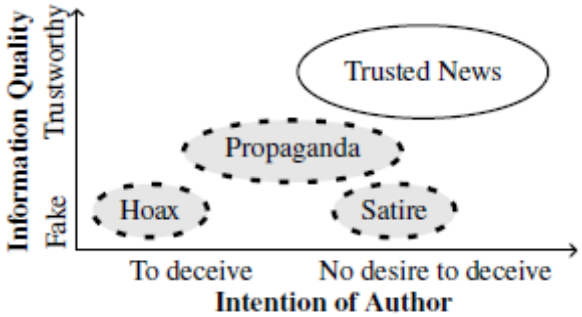

Figure 1: Fake news two factor categorization

On a first level, the author's motive to mislead or not can be used for a more generative categorization of fake news, separating them to misinformation and disinformation respectively (Sharma, K., et al., 2019). Based on the characteristics of intention and quality, the categories bellow do not belong to fake news: (1) satire news with proper context, which has no intent to mislead or deceive readers and is unlikely to be mis-perceived as factual, (2) rumors that did not originate from news events, (3) conspiracy theories, which are difficult to

verify as true or false and (4) hoaxes that are only motivated by fun or to scam targeted individuals.

On the other hand, in their research Kumar, S., West, R., & Leskovec, J., recognize the intention as an important differentiative characteristic but they do not justify the fake entity, since it still has the ability to negatively affect people's lives. Similarly, Rubin, V. L., et al., argue that all the fake entities could be harmful if they shared without context.

In the next section we describe further some of the most common fake news categories, also shown summarily in table.

Category	Citation
Propaganda 3	(Zannettou, S., et al., 2019), (Volkova, S., et al., 2017), (Rashkin, H., et al., 2017),
Conspiracy Theories 1	(Zannettou, S., et al., 2019)
Hoaxes 6	(Zannettou, S., et al., 2019), (Rubin et al., 2015), (Volkova, S., et al., 2017), (Sharma, K., et al., 2019), (Rashkin, H., et al., 2017), (Rubin, V. L., Chen, Y., & Conroy, N. K., 2015)
Biased or one-sided 1	(Zannettou, S., et al., 2019),
Rumors 3	(Zannettou, S., et al., 2019),(Rubin et al., 2015), (Sharma, K., et al., 2019),
Clickbait 3	(Zannettou, S., et al., 2019),(Volkova, S., et al., 2017), (Zhou, X., et al., 2020)
Satire News 5	(Zannettou, S., et al., 2019), (Volkova, S., et al., 2017), (Sharma, K., et al., 2019), (Rashkin, H., et al., 2017), (Rubin, V. L., Chen, Y., & Conroy, N. K., 2015)

Table 2: Fake news categories proposed by previous studies

2.3.1 Biased/one sided

Also known as "hyper partisan", these kinds of news are extremely one-side, having left-wing or right-wing standpoint. Similar with other fake entities, hyper partisan mimics the regular journalism speech favoring only the one side (Kiesel, J., et al., 2019).

Recent researches focused on finding the state of the art in automatic detection of hyper partisan. Kiesel, J., et al., developed new resources, together with a labeled dataset of 754.00 articles. Since previous researches have stated that the two opposite sides are more similar between themselves rather than with neutral news, their goal was to identify the linguistic characteristics (style, syntax, semantics and pragmatics) that differentiate neutral from one-side articles through a challenge called "SemEval task" .This act attracted 322 research teams with the best teams achieving more than 80% accuracy preparing the ground for the next step, contracting explainable classification models (Kiesel, J., et al., 2019).

2.3.2 Rumors

A rumor can be defined as "a piece of circulating information whose veracity status is yet to be verified at the time of spreading", in contrast with fake news definition, the truthfulness of

which is widely known (Zubiaga, A., et al., 2018). The authors Shu, K., et al., identified also the conspiracy theories as a subgroup of rumors, and more specifically, as long-term rumors.

Rumors were the center of studies, mainly psychological ones, since 1940 (Wu, L., et al., 2017). Today rumors flourish in social media and their detection becomes more difficult. Although most of the reliable informative sites check the credibility of their published stories, the large number of not so reliable web information sources render the web more prone to misinformation/disinformation (Liu, Y., & Xu, S., 2016).

Studies have focused on supervised, unsupervised and hybrid methods to separate rumors from real news (Alzanin, S. M., & Azmi, A. M., 2018). Other studies confirmed the point that the propagation style differs significantly from real news', and is used to classify rumors on the web (Liu, Y., & Xu, S., 2016).

2.3.3 Click bait

In order for an article to be considered clickbait it needs to have some basic characteristics: i) a short text, ii) a media attachment such as image or video and iii) the link to the publisher's article. Since one of the success metrics for a website is the crowd that visits the site, click bait articles have a direct economic profit for publishers (Potthast, M., et al., 2018). In their work, Potthast, M., et al., showed that 25% of tweets shared by the top-20 most retweeted news publishers are clickbait.

Although clickbait has the characteristics of a marketing tool, most of the publishers in social media use it, to a greater or lesser extent, to attract more readers. However, journalistic codes of ethics are opposed to these kinds of techniques since they use unethical means to misdirect the reader providing minimal, and most of the times, unreliable information (Potthast, M., et al., 2018).

To conclude, clickbait articles could serve either commercial purposes (traffic attraction), which does not fall under the fake news class, or opinion manipulation, which does (Volkova, S., et al., 2017). Click bait titles are good indicators for fake news but not a necessarily (Shu, K., et al., 2017).

2.3.4 Conspiracy theories

This genre of fake news provides explanations for stories of the news referring to famous people, historical events and other entities that exist in the center of attention, but most of the time, these explanations are based in pseudo-scientific results that look very realistic for a non-expert audience. One of the most famous recent conspiracy theory is the "Pizzagate" incident, a story about Clinton's campaign running a pedophile ring having as base a pizzeria called "Comet Ping Pong". The theory existed in many different sources on the web, making a lot of people to accept its validity and eventually leading a U.S citizen to citizen to commit an armed attack to the "operations center" of this organization (Shahsavari, S., et al., 2020).

In Washington Pizzeria Attack, Fake News Brought Real Guns

Figure 2: Excerpt from New York Times magazine referring to the "Pizzagate" incident

Conspiracy theories create a way of thinking opposite to scientific method of explanation, making the groups of people with predisposition to them more open to sharing and stand up for misinformation (Potthast, M., et al., 2017)

2.3.5 Hoaxes

Hoaxes are stories that fabricate an aspect of a real fact. Wikipedia has defined them as "a deliberately fabricated falsehood made to masquerade as truth."(Kumar, S., West, R., & Leskovec, J., 2016). This category is also known as half-truth or factoid (Zannettou, S., et al., 2019).

The "lifetime" of a hoax highly depends on the weight of their main statement. In their study, Kumar, S., West, R., & Leskovec, J., found the characteristics that differentiate hoaxes that did not propagate through web, since their statements draw too much attention (e.g. the death of Adam Sadler) that is easy to cross reference, with some hoaxes that have lasted for many years and are well cited across the web (Kumar, S., West, R., & Leskovec, J., 2016). Finally, they compared the performance between people and algorithms in the classification task, only to find out that the algorithm outperformed human notations (Kumar, S., West, R., & Leskovec, J., 2016).

Since hoaxes is a type of misinformation with only purpose to deceive the reader, they can be included in fake news category (Volkova, S., et al., 2017).

2.3.6 Propaganda

Propaganda exists as a term since the 17th century describing the propagation of Catholic faith in the New World. Later, in 1938 the Institute for Propaganda Analysis defined propaganda as "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" and classified seven techniques of it: name calling, glittering generalities, transfer, testimonial, plain folks, card stacking and bandwagon (Da San Martino, G., et al., 2020).

Nowadays, propaganda still exists as a persuasion method by digital means, known as "computational propaganda", but still there are not a lot of studies which focus on it (Bolsover, G., & Howard, P., 2017).

In their research, Woolley, S. C., and Philip H., gave a clear definition of computational propaganda: "Computational propaganda is the use of algorithms, automation, and human curation to purposefully distribute misleading information over social media networks."(Woolley, S C., & Philip Howard., 2017). They also stated that Social media platforms are at the center of the propagandist sources because it is the mean from which most of the young people develop their political identities (Woolley, S C., & Philip Howard., 2017).

2.3.7 Satire

Satire is a genre that mimics the writing style of journalistic reporting (Rubin, V. L., Conroy, N., Chen, Y., & Cornwell, S., 2016). According to B.D Horne, and S. Adali, fake news has more similarities in content with satire than with real news. They support their state to their experimental results where they achieved accuracy scores between 71% to 91% in separating fake and satire news from real news based on the content, but they only achieved 67% accuracy when they tried to separate satirical from fake news (Horne, B. D., & Adali, S., 2017). Similar scores were achieved when separating satire titles from real titles, having 75% cross-validation accuracy, but only 55% accuracy when separating satire titles from fake titles (Horne, B. D., & Adali, S., 2017).

Some of the common characteristics of the two genres are that both satire and fake articles use smaller, fewer technical, and fewer analytic words, as well as, fewer quotes, fewer punctuation, more adverbs, and fewer nouns than real articles. Furthermore, they use significantly more lexical redundancy than real articles (Horne, B. D., & Adali, S., 2017).

According to the authors, this means that their persuasion methods are similar and are based in heuristics and not arguments leading to the conclusion that they target audience that would not read beyond titles (Horne, B. D., & Adali, S., 2017).

Finally, most satirical news' primary goal is to entertain rather to mislead the reader (Rubin, V. L., et al., 2015). This differentiation could be enough to exclude satire from the fake news entity list but the term "satire" has also been used by many webpages that do not have any intent to entertain, but to create fake content without being accused of deception (Golbeck J., et al., 2018).

3 Different solution approaches

According to Potthast, M., et al., the detection methods of fake news can be divided in three categories which are: 1) knowledge-based 2) style-based and 3) content-based (Potthast, M., & Kiesel, J., 2017). Also, combinations between different methods have been used, utilizing most of the information that can be extracted from the articles' environment. For example, Yang, Y., et al., used a model named TI-CNN which uses both text and image information to classify entities.

3.1 Knowledge – based

Knowledge-based detection method, also known as "fact-checking", is about identifying the basic claims and statements of the article and compare them with known facts. This procedure could become either manually or automatic. (Zhou, X., & Zafarani, R., 2020)

In **manually** evaluation a person, which considers to be an expert of the specific field, or a group of people, known as crowdsourcing, are responsible to judge the validity of the article's main statements. (Zhou, X., & Zafarani, R., 2020)

Automatic denotation uses methods borrowed from information retrieval, semantic web, and Linked Open Data (LOD) field to classify an article in two stages: Fact extraction and fact checking (Potthast, M., et al., 2017).

In fact extraction the algorithm constructs a knowledge base by mining raw "facts" from the web and during the fact checking stage, it extracts the basic statements of the article and compares them with the knowledge base facts (Zhou, X., & Zafarani, R., 2020). The initial approaches of this category assumed that the credibility of a fact depends of its frequency in the web. Latter, other factors like expertise, trustworthiness, quality, and reliability of the source have been taken into consideration improving the previous method (Potthast, M., et al., 2017).

A challenge referring to the fact extraction step is the big volume of the network. Ciampaglia G. L., et al., proposed a method for finding the optimal knowledge path and evaluate the information depending on the path's length.

3.2 Style-based

"Style-based" is the most common approach in fake news detection and is relied on the findings of researches, performed by psychologists, linguistic experts and computer scientists, that studied the linguistic characteristics of deception (Burgoon, J. K., et al., 2003).

Despite the fact that deceptive writers try to mimic the writing style of journalists, there are still some characteristics that could reveal the authenticity of an article, also known as *Undeutsch hypothesis* (Tversky A. & Kahneman D., 1974)**.** Those characteristics can be split in the bellow categories:

N-grams: They use bag of words representation for each document with tf-idf values (Khan, Junaed Younus, et al, 2019).

Lexical features: Describe character and word level signals, such as total words, characters per word, frequency of large words number of unique words, of first-person pronouns, punctuation counts (Reis, J. C. S., et al., 2019; Shu, K. et al., 2017).

Language features: Syntax in sentence-level describing number of words, syllables per sentence, number of characters per sentence, complex words, long words, number of syllables, word types, and number of paragraphs (Reis, J. C. S., et al., 2019). Also, they calculate several readability metrics, including the Flesch-Kincaid, Flesch Reading Ease, Gunning Fog, and the Automatic Readability Index (ARI), that approximate the appropriate knowledge level that a reader should have in order to understand the article (Pérez-Rosas, V., et al., 2017).

Syntactic features: Include frequencies of function words, phrases and punctuations, and also Parts-Of-Speech (POS) tagging (Shu, Kai, et al., 2017).

Domain-specific linguistic features: Specifically aligned to news domain, such as quoted words, external links, number of graphs, and average length of graphs (Shu, Kai, et al., 2017).

Psycholinguistic features: Category of features based on the linguistics that have to do with the psychological aspect of words. This approach tries to identify the psychological reaction that the article tries to achieve. Psycholinguistic features have their bases on theoretical studies performed by psychologist and computer scientists who studied the linguistic choices of deceivers (Khan, J. Y., et al, 2019).

A really useful tool for these kinds of features is “Linguistic Inquiry and Word Count” (LIWC), a text mining software which correlates each word of a text with a psychological category and is one of the most common tools used for fake news detection. Empath is a similar tool that generates new lexical categories from a small set of seed terms (Khan, J. Y., et al, 2019).

Additionally other studies quantified the "toxicity" of a text, using Google’s API, the subjectivity and the sentiment of the texts in order to reinforce the classification method (Reis, J. C. S., et al., 2019).

Another less traditional approach has also utilized the writing style to identify the validity of an article. The method is called stylometry and is mainly used for extracting information about the author’s identity (like demographic background) (Potthast, M., et al., 2017). It is based on the observation that authors tend to write in relatively consistent, recognizable and unique ways. Its connection with the lying-type detection occurred because when a person tries to mimic a different writing style there are still some features that could identify the deception (Afroz, S., Brennan, M., & Greenstadt, R., 2012).

3.3 Content-based

Regarding fake news published in social media, information from the social network can be used, such as user-based information (number of followers of the publisher, if the profile is verified), post-based information (number of likes, shares etc. in a post) and network-based information (propagation of the news) to tackle effectively the problem (Shu, K., et al., 2017).

User and post based features have been utilized from some recent studies as auxiliary information, since the authors argue that style-based methods have limitations. First, the characteristics (profile/network related) that indicate the likelihood of a user to trust/distrust fake news have extracted and, in a future work, they’ve been used for classification (Shu, K., Wang, S., & Liu, H., 2018; Shu, K., et al., 2019).

Referring to the propagation methods, Wu, L., et al., proposed an approach that identifies the source of an informative element (e.g. article, video or image) to evaluate its credibility (Wu, L., & Liu, H., 2018). A similar technique which also uses the propagation path to identify the source of an article/element, but this time it checks the variety of sources that published it. An article that was published by various sources seems more likely to be real (Ko, H., et al., 2019). Another method identifies different viewpoints of a news topic and, knowing the objective truth, identifies propagation routes in microblogs that are more or less reliable than others (Jin, Z., et al., 2016).

4 Most common datasets

Previous studies had used a big variety of datasets to test their models. Some of the most common ones are referred bellow. The fact that there is no specific benchmark dataset for fake news classification is one of the challenges that have been denoted by both studies of Gravanis G., et al. and Shu K., et al.

Gravanis G., et al. ended up in a methodology of creating an **unbiased dataset,** which contains balanced distribution of articles from different categories and different sources for both fake and real news to be used as benchmark.

Since fake news detection methods use mostly linguistic features, which are connected to the individual author of each article, a dataset should provide a big variety of writing styles to be able to end up in general rules for classification. The rules for this unbiased dataset are (Gravanis G., et al., 2017)**:**

- Each fake news article should be annotated by experts
- Fake news should originate from several sources
- Real news must be published by credible journalism organizations
- Obtain articles of several news categories in order to create a pluralistic collection of real news.

Rules for constructing a reliable dataset had also been proposed by (Rubin, V. L., et al, 2016). The authors suggested that such a dataset should (1) include both fake and real news items, (2) contain text-only news items (3) have a verifiable ground-truth, (4) be homogeneous in length and (5) writing style, (6) contain news from a predefined time frame, (7) be delivered in the same manner and for the same purpose (e.g. humor, breaking news) for fake and real cases, (8) be made publicly available, and (9) should take language and cultural differences into account. (Rubin et al., 2016)

In their research, Pérez-Rosas V., et al., relied on these nine requirements to create their own datasets, called FakeNewsAMT. FakeNewsAMT combines two datasets, for the first one it uses majority voting of 10 individuals for creating a dataset of total 240 fake news and 240 true ones, and for the second set, called Gossip, they extracted articles directly from the web referring to celebrity gossip (100 true and 100 fake) (Pérez-Rosas V., et al., 2017).

Some of the most famous datasets are:

1) **Buzz Feed News**, a fake news dataset of Facebook published news from 9 well-known news agencies related to the 2016 U.S election. Each from the right and from the left side, as well as three large mainstream political news pages (Politico, CNN, ABC News) fact-checked claim-by-claim by 5 BuzzFeed journalists. (Reis, Julio CS, et al., 2019; Granik, M., & Mesyura, V., 2017; Yazdi, Kasra Majbouri, et al., 2020; Shu, Kai, et al., 2017).

2) **BS Detector**, an automatic software that labels specific webpages as fake news source and classify all its publications as fake (Yazdi, Kasra Majbouri, et al., 2020; Shu, Kai, et al., 2017)

3) **LIAR**, a dataset based on POLITIFACT.COM webpage, which consists of 12.8K human labeled short statements. Each statement belongs to one of six balanced classes (pants-fire, false, barelytrue, half-true, mostly-true, and true) according to their reliability. According to (Wang, W. Y., 2017) the previous datasets were too small (less than a thousand items) for developing a machine learning algorithm. The data gathered from political debates, TV ads, Facebook posts, tweets, interviews etc. and they include information about the statement, the source, labeling and justification of the label. The creators of this dataset stated that they created a benchmark dataset for both fake news detection and fact-checking (Wang, W. Y., 2017; Shu, Kai, et al., 2017).

4) **CREDEBANK**: CREDEBANK is a corpus of tweets labeled by 30 human annotators. It includes more than 60 million tweets in a time period of 96 weeks, grouped into 1049 real-world events (Mitra, T., & Gilbert, E., 2015).

5) **FakeNewsNet:** The authors (Shu, K., et al., 2018) argue that fake news detection is a process that requires a multidimensional solution approach and only the linguistic information is not enough.

	News Content		**Social Context**				**Spatiotemporal Information**	
	Linguistic	Visual	User	Post	Response	Network	Spatial	Temporal
BuzzFeed	✔							
LIAR	✔						✔	✔
BS Detector	✔							✔
CREDEBANK	✔		✔	✔				
FakeNewsNet	✔	✔	✔	✔	✔	✔	✔	✔

Table 3: Fake news datasets characteristics as opposed by Shu, K., et al.

For that reason they deployed FakeNewsNet (FNN), a data repository of labeled news articles providing information about the news content, social context, and spatiotemporal information. This variety of available features gives the opportunity to future studies to try different approaches and reveal different aspects of the topic, such as understanding the diffusion of fake news in social networks and identify the social media profile characteristics of provenances and persuaders (Shu K., et al., 2018).

Dataset	**Article**
Buzzfeed 11	(Reis, J. C.S., et al., 2019), (Yazdi, K. M., et al., 2020), (Granik, M., & Mesyura, V., 2017),(Shu, K., et al., 2017), (Yang, S., et al, 2019), (Potthast, M., et al., 2019), (Potthast et al., 2017), (Horne, B. D., & Adali, S., 2017), (Zhou, X., et al., 2020), (Gravanis, G., et al., 2019), (Reis, J. C., et al., 2019)
LIAR 8	(Wang, W. Y., 2017), (Yazdi, K. M., et al., 2020), (Shu, K., et al., 2017), (Yang, S., et al, 2019), (Khan, J. Y., et al., 2019) (Gravanis, G., et al., 2019), (Zhou, X., Jain, A., et al., 2020), (Rashkin, H., et al., 2017)
BS Detector 2	(Yazdi, K. M., et al., 2020), (Shu, K., et al., 2017)
CredeBank 2	(Shu K., et al., 2017), (Mitra, T., & Gilbert, E., 2015)
FakeNewsNet 1	(Shu K., et al., 2018)

Table 4: Fake news datasets used in prior studies

5 Challenges in fake news

The first challenge in fake news classification is that some topics require prior domain knowledge to test the validity of their claims. In their study Pérez-Rosas V., et al., showed that humans were better at detecting fake claims in gossip related content but not so good at

articles which contained matters from six different domains such as politics, science, geography, etc.(Pérez-Rosas, V., et al., 2017).

Another challenge in the fake news problem is that a reliable labeling of an article can only be notated by expert journalists of the field by careful analysis of all the aspects of the article, method which is very time costly, and many times, due to the large volume of articles published every day, practical impossible (Shu, K., et al., 2017). A proposed solution to this problem, which can be applied in news articles published in social media, is to use the readers' reactions in the shared article to define if it is real or fake. In that way, using an unsupervised learning technique, they overpassing the news denotation problem, but still have not reached the quality of a human-labeled method (Yang, S., et al., 2019).

In the section "Most common datasets" we have discussed that there is not yet a benchmark dataset for machine learning algorithms deployment. This leads to another obstacle in detection methods which is the inability of setting a general set of linguistic features that could be used in classification algorithms across the different domains. Since most of the datasets focus in different domains, which have different writing styles, this task seems very hard to achieve (Pérez-Rosas, V., et al., 2017).

Finally, identifying fake news isn't by itself enough to solve the problem on its root since once misperception is formed it's very hard to correct it. There are two stakeholders in this phenomenon that we should take into consideration, the reader and the publisher. Both of the sides have their own beliefs, ideas and interests.

"It's Easier to Fool People Than It Is to Convince Them That They Have Been Fooled." – Mark Twain

Referring to the reader's side, there are two psychological reasons that prevent him/her from detecting fake news. First there is the *Illusion of asymmetric insight*, which is the idea that his/her point of view is more logical than the others, and second, *Confirmation Bias*, the acceptance of information that agrees with his/her beliefs (Pronin, E., 2001; Nickerson, R. S., 1998). This creates a conflict inside the reader's interests who, from the one side, wants true information, but from the other side, wants information that agrees with his/hers prior beliefs.

Similar behaviors are also evident in social media. Users end up surrounding themselves with people that have the same opinions as theirs. In that way, they repeatedly receive only one side aspects of news, which also happens to be their own aspect. This phenomenon is known as **Echo Chamber** effect. The result of is to create a very fertile environment for fake news to flourish in digital communities. That is because we tend to accept an information as true if our environment accepts it too (Availability cascade theory) and if we come across it very often (Validity effect) (Kuran, T., & Sunstein, C. R., 1998; Boehm, L. E., 1994). The echo chamber effect covers both of these aspects meaning that if the close digital society of a user accept a fake information as true, the user himself/herself is very possible to accept it too (Jamieson, K. H., & Cappella, J. N., 2008).

Publishers also deal with their dilemmas. The writers want to gain popularity and status, so they might change pieces of information trying to make their article more appealing and interesting to the majority of the readers, since it is more possible for users to share a piece of news that agrees with them. On the other hand they also want to be considered trustworthy (Zhou, X., et al., 2019).

6 Describing the Classifiers

In the next sections we are going to describe some of the most common classic (not artificial neural networks) algorithms that have been used to tackle fake news detection problem. Table 3 represents some of them and the corresponding researches that've been used in.

6.1 Classic algorithms

Algorithm	Papers
Naïve Bayes 7	(Granik, M., & Mesyura, V., 2017), (Yazdi, K. M., et al., 2020), (Gravanis, G., et al., 2019), (Granik, M., & Mesyura, V., 2017), (Khan, J. Y., et al., 2019), (Rashkin, H., et al., 2017), (Reis, J. C., et al., 2019),
SVM 10	(Reis, Julio CS, et al., 2019), (Horne, B. D., & Adali, S., 2017), (Gravanis, G., et al., 2019), (Pérez-Rosas, V., et al., 2017), (Wang, W. Y., 2017), (Khan, J. Y., Khondaker, M., et al., 2019), (Horne, B. D., & Adali, S., 2017), (Yazdi, K. M., et al., 2020), (Reis, J. C., et al., 2019), (Gilda, S., 2017)
Logistic Regression 4	(Wang, W. Y., 2017), (Khan, J. Y., Khondaker, M., et al., 2019), (Volkova, S., Shaffer, K., et al., 2017), (Gilda, S., 2017)
kNN 4	(Reis, Julio CS, et al., 2019), (Gravanis, G., et al., 2019), (Khan, J. Y., Khondaker, M., et al., 2019), (Reis, J. C., Correia, A., et al., 2019)
Decision Tree 3	(Yazdi, K. M., et al., 2020), (Gravanis, G., et al., 2019), (Khan, J. Y., et al., 2019)
Adaboost 2	(Gravanis, G., et al., 2019), (Khan, J. Y., et al., 2019)
Bagging 1	(Gravanis, G., et al., 2019)
Random Forest 4	(Reis, Julio CS, et al., 2019), (Zhou, X., et al., 2020), (Reis, J. C., et al., 2019), (Gilda, S., 2017)
Gradient Boosting 1	(Gilda, S., 2017)
XGBoost 3	(Reis, Julio CS, et al., 2019), (Zhou, X., et al., 2020), (Reis, J. C., et al., 2019),

Table 5: Classic algorithms used for fake news detection

6.1.1 Support Vector Machine (SVM)

Support Vector Machine was the first attempt to approximate binary classifiers (Suykens, J. A., & Vandewalle, J., 1999). It is one of the most famous and usable machine learning algorithms for both classification and regression tasks. The basic strategy of SVM is that it projects data points into higher dimensions where are linearly separable, and as a result, easier to separate using a hyperplane.

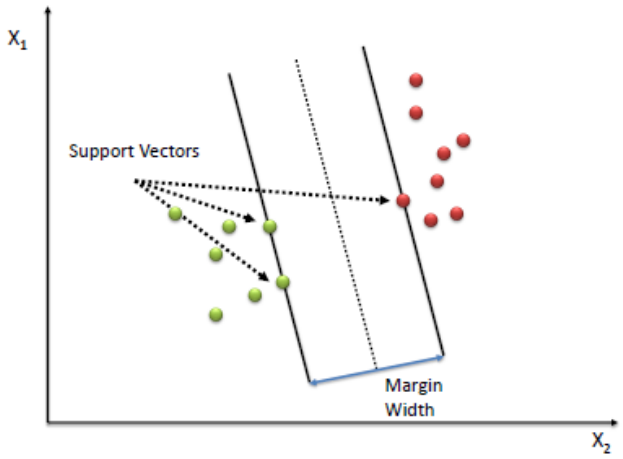

Figure 3: Support Vector machine in 2-D space

In order to find the optimal hyperplane, SVM tries to maximize the margin. Margin is the distance between the points of the different classes, and maximal margin means less probability of misclassification (Suykens, J. A., & Vandewalle, J., 1999).

One of SVM's biggest advantages is that it can separate also non-linearly classes by transforming the data into higher space. This is achievable by applying a Φ function to the data, projecting them into higher space z = Φ(x). Luckily, SVM doesn't need to calculate the function itself but only the product $\Phi*\Phi^T$ which is called Kernel. This is known as the "Kernel trick" and saves a lot of computational space. The most common Kernels used for SVM are: Gaussian (or RBF), exponential, Sigmoid and Linear. From them RBF is the most powerful one because it can project data into infinite dimensions (Hofmann, M., 2006).

Although Support Vector Machines are very useful models they are not very efficient when handling large volume of data and also the user needs to choose the values of some hyper parameters meaning that there isn't a guarantee that will end up in the optimal version of the model (Suykens, J. A., & Vandewalle, J., 1999).

6.1.2 Naïve Bayes

Naïve Bayes uses a simplified approach to classify data points. The classifier relies on two important simple (or naïve) assumptions. In particular, it assumes that the features are statistically independent, and it assumes that no hidden or latent attributes influence the prediction process. Even most of the times the assumptions are violated, still Naïve Bayes tend to have good results (Raschka, S., 2014).

Naïve Bayes is relied on the "Bayes Theorem" which utilizes prior probability to calculate the posterior probability of a sample to belong to a class. More specific, for a test sample x, its probability to belong to the class C is given by the formula:

$$P(C = c|\, X = x) = \frac{p(C = c) * p(X = x\,|C = c)}{p(X = x)}$$

Where p(C=c) is the probability for a random sample to belong to class C, p(X=x) is the probability for random sample to have the value X, and p(X=x | C=c) is the probability of a random sample x to have the value X given that belongs to the class c. The algorithm calculates

for each sample its probability to belong to each class and assigns the sample to the class with the highest probability (John, G. H., & Langley, P., 2013).

Naïve Bayes is very popular in text classification problems too. The first step in this process is to transform the text into a numerical form, also known as word embedding, for the algorithm to process, some of the most common word embedding techniques will be analyzed in next section. A document is analyzed as part of a document selection where each unique word of the selection form the corpus.

$$P(text\ |class = c_1) = \prod P(word_i \mid c_1)^b\ (1 - P(word_i\ |c_1))^{1-b}\ \ b \in (0,1)$$

When each word of the corpus has a vector representation, Naïve Bayes algorithm calculates the probability of a document to belong in a class given the words used in the text. To do that it calculates separately for each word ($word_i$) the probability of the text to belong to this class and finally adds all the words' probabilities. The formula above parses all the words of the corpus and the parameter "b" is equal to 1 when the $word_i$ appears in the specific document and 0 if it doesn't (Raschka, S., 2014).

6.1.3 Decision Tree

Decision Tree is also one of the most common algorithms for classification and regression in machine learning due to its simplicity and efficiency. In contrast with other machine learning models, decision trees don't need a lot of data preprocessing, like handling missing values or scaling (Rokach, L., & Maimon, O. Z., 2008)

A Decision tree consists of i) decision/internal nodes and ii) leaf nodes. A decision node splits data into subsets based on a feature's condition. If the new subset is homogeneous then no additional splits accruing, if not, then the algorithms splits furtherer. The splitting stops at the leaf nodes in the following cases; all the data in a node belong to the same class, there are no more attributes to split or there are no samples left (Quinlan, J. R., 1987). In order to decide if the sample is homogeneous some statistical measures could be used like Information Gain, Gini index and Gain Ratio (Tzirakis, P., & Tjortjis, C., 2017)

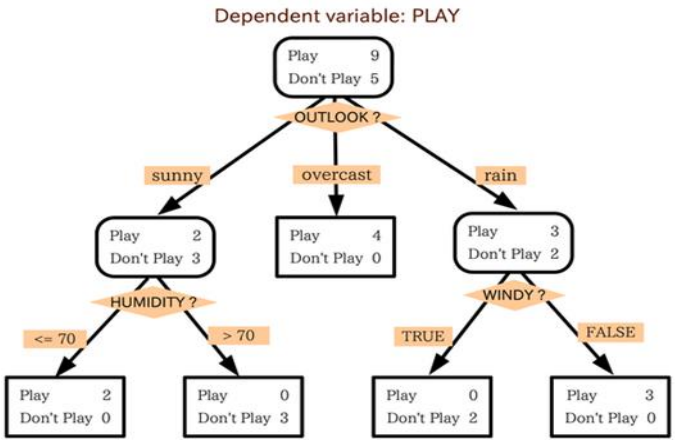

Figure 4: Decision Tree's structure

Information Gain (ID3) is the reduction of entropy in the dataset after a split. The entropy of a dataset is defined by the bellow formula:

$$\mathrm{Info}(\mathrm{D}) = -\sum_{\mathrm{i}}^{\mathrm{C}} \mathrm{p_i} * \log_2 \mathrm{p_i} \quad (1)$$

Where pi is the probability that a random sample D belongs to class i.

$$\mathrm{Info}\ (\mathrm{D_A}) = \sum_{\mathrm{j}=1}^{\mathrm{v}} \frac{\mathrm{D_j}}{\mathrm{D}} * \mathrm{Info}(\mathrm{D_j}) \quad (2)$$

The Information needed to classify D after we split the dataset in A.

$$\mathrm{Gain}(\mathrm{A}) = \mathrm{Info}(\mathrm{D}) - \mathrm{Info}(\mathrm{D_A}) \quad (3)$$

Gain(A) is the information gained by the splitting in A.

So we are looking for the split that maximizes the decrease in entropy.

The problem with information gain is that it favors the big attributes with large number of values. The processor of ID3 is called *Gain Ratio* (C4.5) and uses a normalized version of Information Gain (Salzberg, S. L., 1994).

$$\mathrm{GainRatio} = \frac{\mathrm{Gain}(\mathrm{A})}{\mathrm{SplitInfo}(\mathrm{A})} \quad (4)$$

$$\mathrm{SplitInfo}(\mathrm{A}) = -\sum_{\mathrm{j}=1}^{\mathrm{v}} \frac{\mathrm{D_j}}{\mathrm{D}} \log_2 \frac{\mathrm{D_j}}{\mathrm{D}} \quad (5)$$

The attribute with the max gain ratio is picked for the split.

Finally another metric, called *Gini Index* (CART), has been used in a binary decision tree algorithm called CART. Gini index calculates the probability of a particular variable to be classified wrong when picked randomly (Apté, C., & Weiss, S., 1997)

$$\mathrm{Gini}(\mathrm{D}) = 1 - \sum_{\mathrm{j}=1}^{\mathrm{n}} \mathrm{p_j} \quad (6)$$

$$\mathrm{gini_A}(\mathrm{D}) = \sum_{\mathrm{j}=1}^{\mathrm{n}} \frac{\mathrm{D_j}}{\mathrm{D}} \mathrm{gini}(\mathrm{D_j}) \quad (7)$$

$$\Delta\mathrm{gini}(\mathrm{A}) = \mathrm{gini}(\mathrm{D}) - \mathrm{gini_A}(\mathrm{D}) \quad (8)$$

A decision tree splits the feature space into mutually exclusive sections where each section belongs to a class. These sections are called 'disjucts' and consist of a list of attribute-value conditions in the form of "IF condition=True THEN". To minimize the errors, each disjuct is assigned to the class where the majority of the points belongs. The algorithm might split the instance space in a large number of one class disjucts leading to overfit (Apté, C., & Weiss, S., 1997).

To mitigate overfitting, the learning procedure goes through a second phase, called pruning. There are two kinds of pruning according to when it takes place i) pre-pruning, which sets a goodness threshold in every split to avoid overfitting (but it's difficult to choose the threshold) and ii) post-pruning that removes branches from a fully grown tree (Apté, C., & Weiss, S., 1997). Ensemble methods could also help mitigating overfitting problems.

6.2 Ensemble methods

Ensemble is called the technique where a number of models, ideally weak models, are joined to increase the performance of the combined model. Some of the most popular ensemble methods are i) Boosting (weighted vote between n classifiers) and ii) Bagging (averaging the prediction of n classifiers) (Breiman, L., 1996; Schapire, R. E., 1990).

6.2.1 Boosting

Boosting method is based on the logic of dividing the dataset into smaller ones and test/train the base algorithm to the least accurate classified parts.

In more detail, the algorithm divides the dataset into smaller subsets with all having the same probability to be chosen for training and testing. Then, depending on the performance of the algorithm, it increases the probability of the false classified subset(s) to be picked. This procedure repeats many times.

The final model would have a weighted voting about all the subsets outcomes, with weights being function of each model's accuracy in the specific subset, meaning that the model with the best performance in a subset would contribute more in the final outcome of the specific subset (Breiman, L., 1996).

6.2.2 Gradient Boosting

Gradient boosting uses decision trees as base algorithms. At the first step, the algorithm starts only with one leaf node and makes the initial predictions based on that. Then it calculates pseudo residuals by differencing the predicted values with the real ones. In the second step, a decision tree is used trying to predict the residuals. The new prediction will be equal to the initial prediction plus the weighted prediction of the first tree for the residual. Using the new predictions, new residuals are calculated, and based on them, a second tree is being used. Repeating these steps many times, the final prediction would look like this:

$$y_{pred} = y_{pred1} + a * y_{tree1} + a * y_{tree2} + \cdots + a * y_{tree_n} \quad (9)$$

Where n is the number of the decision trees that were used. The constant α is the learning rate which takes values from 0 to 1 and tackles overfitting by making small steps towards the right direction each time (Friedman, 2002).

6.2.2 Bagging

Bagging or Bootstrap Aggregation method divides the training set into n subsets and then it constructs m bootstrap samples (one for each estimator) that contains random subsets with replacement. The base algorithm is trained in m bootstrap samples and the final output is calculated by averaging the output (for regression) or by the majority voting (for classification) of each classifier (Breiman, L., 2001)

6.2.4 Random Forest

Random forest is using Bagging method based on the decision trees (Breiman, L., 2001)

At each step the estimator trains a decision tree using the bootstrap data and tests using the out of the Bootstrap (OOB) set. The final prediction is made by aggregating all the decisions of the m-classifiers, known as majority voting (Liaw, A., & Wiener, M., 2002). The Random Forest algorithm is able to mitigate overfitting problem of decision trees by combining a lot of them and also can handle very efficiently missing values (Ullah, I., Raza, B., et al., 2019). Also it provides very useful insights about the variable importance (Breiman, L., 1996).

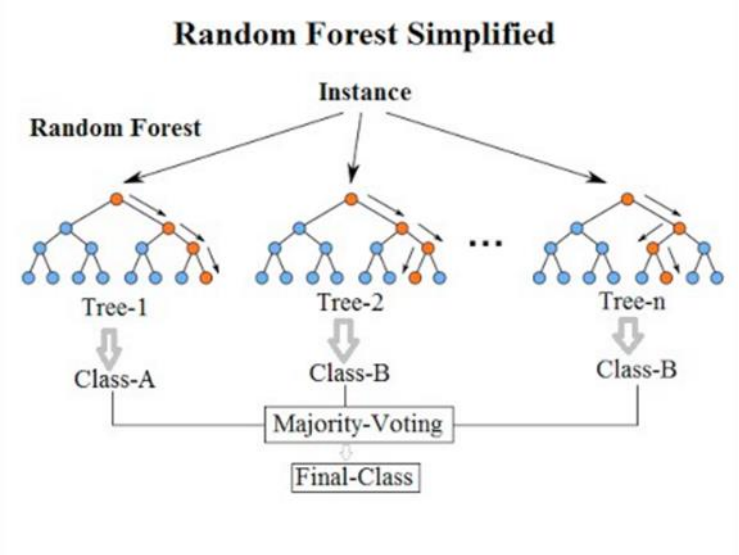

Figure 5: Random Forest structure

6.3 Artificial Neural Networks

Artificial Neural Networks have produced some of the best performances in fake news detection and many text classification tasks in general (Zhang, Y., et al., 2015). In the next section we will describe the functionality of some neural networks, starting from the simplest, the Perceptron, to the more complicated Multilayer Perceptron (MLP), Convolutional Neural Network (CNN) and Long-Short Term Memory neural network (LSTM). The table below represents recent papers that used one (or more) of the above neural networks to tackle fake news problem.

Algorithm	Article
CNN 5	(Khan, J. Y., et al, 2019)(Wang, W. Y., 2017), (Wang, W. Y., 2017), (Volkova, S., Shaffer, K., et al., 2017), (Yang, Y., et al., 2018)
LSTM 5	(Khan, J. Y., et al, 2019) (Rashkin, H., et al., 2017), (Wang, W. Y., 2017), (Volkova, S., et al., 2017), (Rashkin, H., et al., 2017)
MLP 1	(Khan, J. Y., et al, 2019)

Table 6: References of CNN, LSTM and MLP in the literature

6.3.1 Perceptron

Perceptron is a supervised learning algorithm used for binary linearly separated classification problems. The basic components of a Perceptron is a) the input layer (X), b) the synaptic weights of each input and c) the activation function (Noriega, L., 2005).

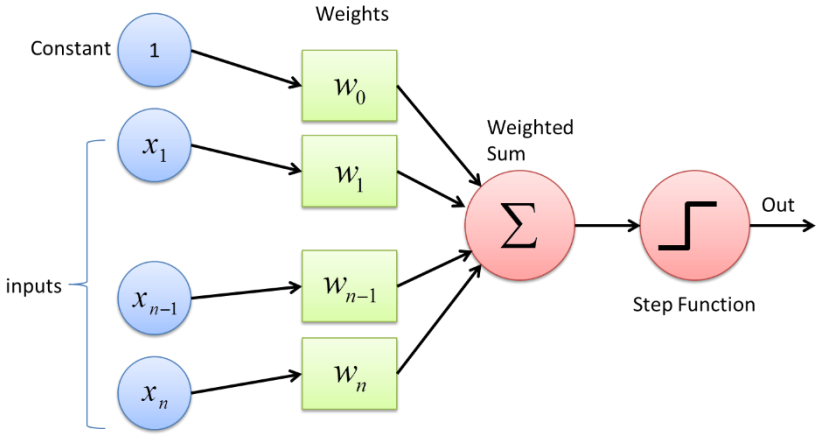

Figure 6: Perceptron's basic structure

The input data X = $[1,x_1,x_2,..x_n]$ entering perceptron one pattern at a time. When all the patterns have entered the model is the end of one epoch. The input values multiplied with each neuron's weight W = $[w_0,w_1,w_2,..w_n]$ and added up.

$$U = \Sigma w_i * x_i - \theta \text{ (10a)}$$

Θ is called threshold of the neuron and it's equal to w_0. An activation function f is applied to the result of the addition (U) and the output depends on which activation function is more appropriate for the problem.

$$\Upsilon = f(U) \text{ (10b)}$$

Most common activation function, since the problems needed to be solved are classification ones, is a step function (0/1 or -1/1).

$$\text{Out} = \begin{cases} 0 \; if \; y < 0 \\ 1 \; if \; y \geq 0 \end{cases} \text{ (10c)}$$

At the end of each epoch, the weights are updating depending on a cost function. The algorithm terminates when cost function has converge to a minimum values.

One basic perceptron's limitation is that if the classes aren't linear separable (e.g. Exclusive OR or XOR problem) the algorithm never converges. This limitation is overpassed by using more layers between the input and the output layer, forming that way a Multi-layer perceptron (Noriega, L., 2005).

6.3.2 Multi-layer Perceptron (MLP)

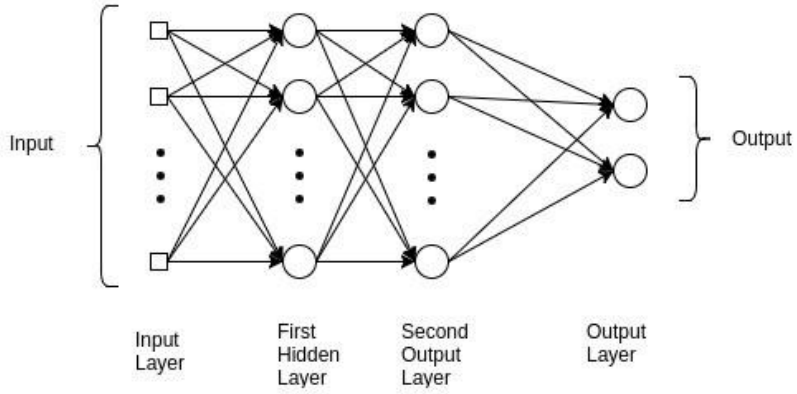

Figure 7: Multi-layer Perceptron's structure

Multi-layer Perceptron uses additional hidden layers between the input and the output layers, which contain one or more neurons. The data are inserted from the input layer and they are forwarded to the next hidden layer. Each neuron receives inputs from all the neurons of the previous layer and forwards its output to all the neurons of the next layer creating a fully connected network. This procedure continues until the final layer, where the output is calculated. The connection between two neurons have weights that are changing during the training phase (Noriega, L., 2005). At the end of each epoch, MLP computes the divergence of target-output value using a cost function (e.g. mean squared error), and by using Back – Propagation, readjust the weights of the neurons connections due to minimize the cost (Jain, A. K., Mao, J., & Mohiuddin, K. M., 1996).

A significant difference that makes MLP more effective than Perceptron is the different activation functions that the hidden layer neurons use. In MLP, except from the step function, a sigmoid or hyperbolic tangent can be used. In that way, MLP can approximate not only discrete values but continues (Noriega, L., 2005).

Multi-layer perceptron is considered a universal approximator, meaning that given enough hidden units and data, it can approximate any function (Csáji, B. C., 2001).

One of its defects is that back-propagation is a gradient method which uses the derivative of the activation functions. Sigmoid and hyperbolic tangent could get very big or zero values for specific inputs leading to either exploding or vanishing gradient. In these cases the weight update will fail and the model will stack in a specific state. This problem can be resolved with Deep Belief Networks but it is out of the scope of this thesis (Kolbusz, J., Rozycki, P., & Wilamowski, B. M., 2017).

6.3.3 Convolutional Neural Network (CNN)

Convolutional Neural Networks are categories of multilayer networks inspired by the virtual cortex. They are supervised deep learning techniques most often used for image/video processing tasks, but they have also applied for recommendation systems, time series and natural language processing tasks.

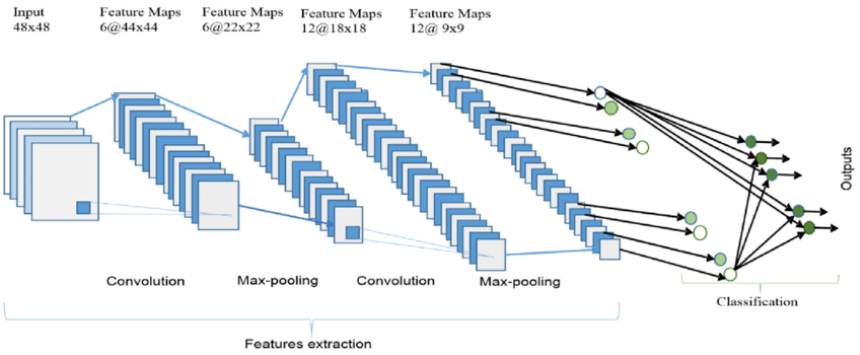

Figure 8: Convolutional Neural Network's structure

The key-concept in CNN's architecture is "convolution". In simple words, convolution is a filter, or mask or kernel, applied in an NxN matrix, detects some patterns in the matrix depending on the task we want to perform, and return a smaller nxn matrix (feature map) that contains the biggest amount of information. In that way, each neuron in CNN receives inputs only from the inputs that match with the filter of the previous layer creating this way a deep local network. The filter area is called receptive field and it's applied step by step to the input layer according to the *stribe* length, which defines the distance between adjacent receptive fields along x-y directions. CNN uses these layers, called convolutional layers, with an activation function (e.g. ReLU) to compute the layer's output (Le, Q. V., 2015).

Convolution layers reduce significantly the weights needed to be trained. Also, by using a technique called "weight sharing" some weights are constraint to be equal with others, reducing further the number of trainable parameters. The training of the model becomes faster and since it has less neuron connections, meaning less weights to train, it mitigates overfitting and the need of a big training set (Wu, J., 2017).

After a convolutional layer, it is very common to use a pooling layer. Pooling layers summary their input providing a smaller output that captures, fortunately, most of the information. Most common techniques are max pooling and average pooling, which return the maximum and average value of the receptive field respectively. Another very important property of pooling layer is that returns fixed output, which is required for classification tasks (Le, Q. V., 2015).

6.3.4 Long Short Term Memory (LSTM)

Long-Short Term Memory network (LSTMs) belongs in the category of recurrent neural networks that take into consideration their previous states to compute the new output, acting as they have some kind of memory. LSTM is one of the most powerful model in this category because it has arbitrary long memory. The key characteristics that differentiate it from the other recurrent networks are the two gates it uses, g_{in} and g_{out}, and the forgetting factor f (Staudemeyer, R. C., & Morris, E. R., 2019).

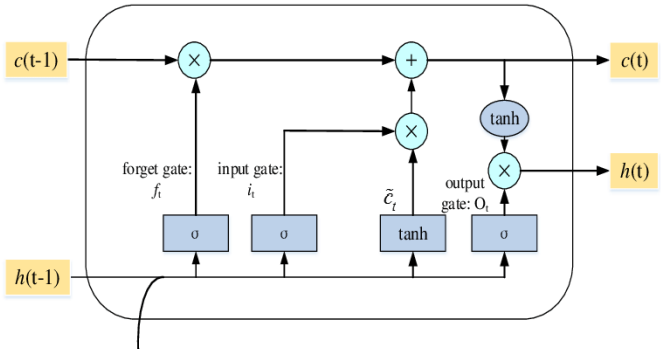

Figure 9: Long – Short term memory neural network's structure

Very important term also is the "state" of the system. LSTM is a stateful/dynamic

system which means that the output is not only depend by the input value but also from the previous outputs. These previous outputs describe the "state" of the system.

Each time an input is inserted in the network, the g_{in} decides how much of the current input should affect the output. The g_{out} gate decides if the current state should be propagate further and how much.

The forgetting factor f takes values from 0 to 1 and affects how much to remember from the previous state c(t-1), with 0 means none and 1 means all of it. Because f is a trainable variable, the algorithm eventually decides how much of the previous states to keep and that's what affect the "Long" – "Short" term memory of the model. Also, g_{in} and g_{out} are trainable parameters giving the system the sense of adaptability.

LSTM has been used for many natural language processing tasks like speech recognition and machine translation (Staudemeyer, R. C., & Morris, E. R., 2019).

6.3.5 Word Embeddings

Another important contribution of the neural networks in text classification problems were the semantic word embedding.

The problem that needed to overpass was that machine learning models can only understand numerical inputs, so words needed to be represented as vectors. The first step of defining a word vector is first to define its surrounding environment, meaning the corpus of words, which also affects the vector's dimensions. For example, in an article all the unique words represent the corpus within each word can be represented (Aizawa, A., 2003).

The most simple word representation is one hot encoding vector were value 1 indicates the existence of this word in this dimension, and 0 indicates the absence of it.

In the sentence "The cat sat on the mat" the one hot vector representation of the word "the" would be [0,1,0,0,0,1,0] and for the word "cat" would be [1,0,0,0,0,0].

	The	Cat	Sat	On	The	Mat
The	**1**	**0**	**0**	**0**	**1**	**0**
Cat	**0**	**1**	**0**	**0**	**1**	**0**
Sat	0	0	1	0	0	0
On	0	0	0	1	0	0
Mat	0	0	0	0	0	1

Table 7: One hot encoding vector example

Other very common techniques based on the frequency of the words broadly used for creating word embeddings are; the word count, term frequency – inverse document frequency (tf-idf) score and co-occurance vector, but we would not describe further as its out of the scope of this thesis (Aizawa, A., 2003). The revolution in word embeddings came in 2013 with Google's word2vec algorithm.

Word2vec was the first neural word embedding algorithm. The difference that brought in the field of natural language processing was that the vector representation of a word could also capture its semantics, meaning that words with similar meaning would also have similar

(close) vectors. The two basic neural networks that can be used to produce those vectors are skip-gram and CBOW (continues bag of words) (Ling, W., et al., 2015).

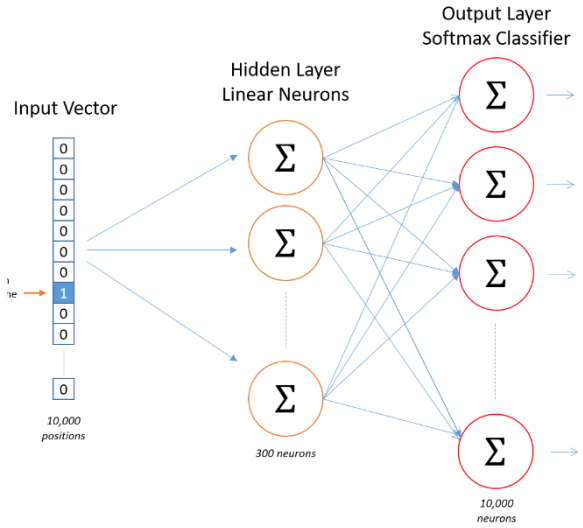

Figure 10: Example of Skip gram's architecture

The architecture of those neural networks consists of an input layer, one hidden layer and one output layer with softmax classifier. The algorithms are trained by a large corpus of texts, like Wikipedia articles or google news. The text is split to sentences and each sentence to its words. For each input sentence the algorithm try to predict either one word using their surroundings (skip gram) or the surroundings using one word (CBOW). When the model is trained, then the neuron's weights from hidden layer correspond to the word embeddings (Levy, O., & Goldberg, Y., 2014).

Word embeddings create very useful feature sets that could be used separately or combined with other feature sets. Also there are a lot of open source libraries that provide pre-trained word vectors, like spacy's library, meaning that a user does not have to train its own model.

7 Experimental Part

In this section we will test several models for separating fake from real news by utilizing information from the text of the articles.

In the first part we will describe the dataset used in the experiment, the preprocessing method, the functions used, the features extracted and the feature selection process. In the next part we will test the different performances of the feature sets using classical (non-neural network) algorithms. Then we will test the performance of three neural networks, MLP, CNN and LSTM, first by finding the optimal hyper parameters using grid search and nested for loops, and then trying different architectures. Finally we will comment on the results and findings.

7.1 Datasets description

Fake News Detection Challenge KDD 2020

The dataset used for this thesis was the one provided in the "Second International TrueFact Workshop: Making a Credible Web for Tomorrow in conjunction with SIGKDD 2020", created by Kai Shu and contained news for famous people of the timeliness. The reason why we chose this dataset over some of the most common ones was because we had the opportunity to test in real time our results with other competitors, having this way a performance perspective.

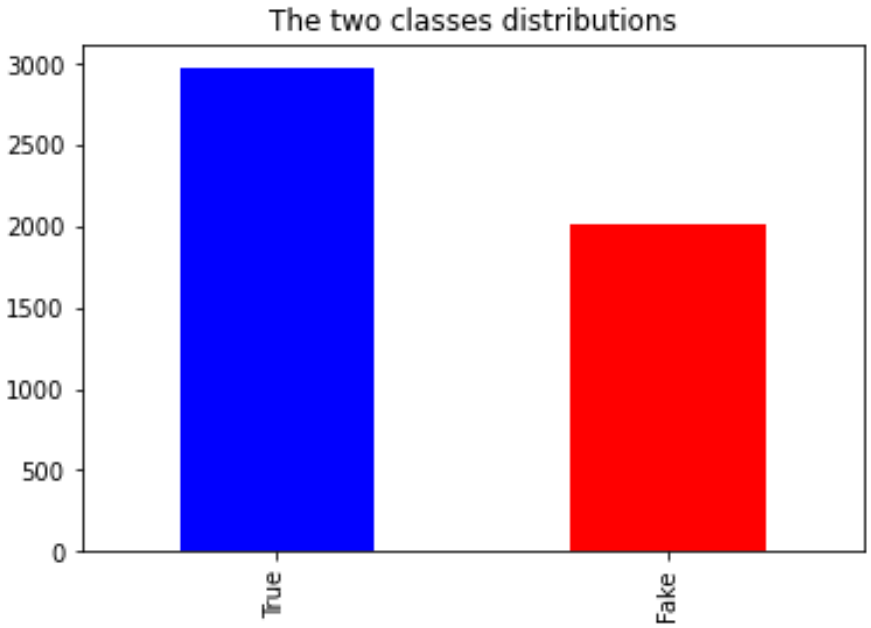

Figure 11: Distribution between the two classes

The dataset consists of 2972 real and 2014 fake news, meaning that we're dealing with a balanced dataset.

7.2 Preprocessing

7.2.1 Text preprocessing

The methods that we used for text preprocessing were some of the most common methods in natural language processing tasks.

First of all, in sentence level, we applied some techniques to clean the text from the unnecessary elements such as stop words, numbers and links. These elements do not contribute significant in the overall information of the text and just increase the noise.

In word level, we tested three different representations of the text: i) by removing stop words and keep the rest of the text words as they were, ii) by applying stemming and iii) applying lemmatization. The raw text was used to count POS frequencies and for Name Entity features. All the three representations were used for word embeddings creation and tested before we ended up in the best representation.

7.2.2 Principal Component Analysis (PCA)

Another method that was used during the preprocessing phase was principal component analysis, also known as PCA.

Primarily, PCA was utilized to reduce the number of dimensions by identifying the more important features i.e. the principal components. The number of principal components is less than or equal to the smaller of the number of original variables. The first principal component has the largest possible variance and each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to the preceding components.

PCA was primarily used for decreasing the word embeddings dimensions and not in the lexical feature set. The reason for that is that PCA technique does not just drop dimensions from the feature set but it compresses them, meaning that we would not be able to separate the best performing features after applying it. Other methods were used for that cause, like computing the mutual information between independent and dependent variable and keeping the most important subset, which will be discussed in further sections.

7.2.3 K – Fold cross validation

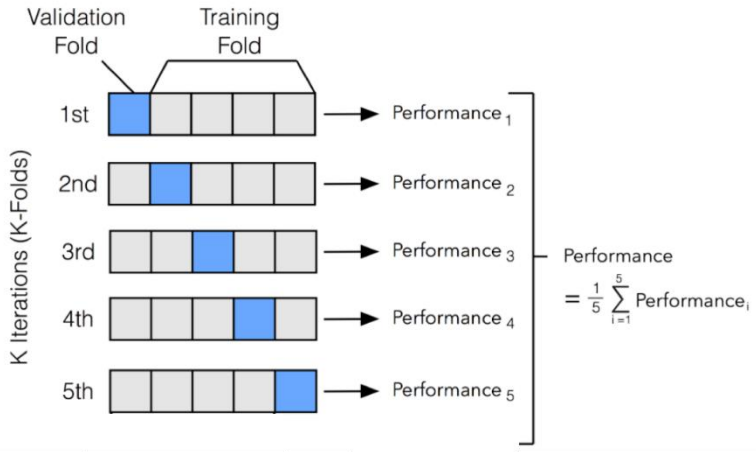

Figure 12: 5-fold cross validation example

Cross-Validation is a technique that separates a dataset into two segments: one used to train the model and the other used to test the model. In our case, a k-fold cross validation for 10 folds was used, in which the data is first divided into k equally (or nearly equally) sized parts or folds. Subsequently k iterations of training and validation are performed such that within each iteration a different fold of the data is held-out for validation while the remaining k-1 folds are used for learning. By utilizing k - fold cross validation in my experiments I was able to avoid overfitting and extract more representative results in the model testing step.

7.3 Experimental steps

As mentioned in the literature review, there are some key differences between fake and real news in linguistic level. Our goal was to utilize this lexical information from the articles for the classification problem. In the next section all the functions used for preprocessing and feature extraction tasks are presented together with a small description.

7.3.1 Functions used

Function	Explanation
clean_text	Returns a clean text by removing URLs, mentions (e.g @Name), numbers and punctuations. I didn't perform lemmatization or stemming because I wanted to count syllables of words and other metrics in the next steps.
stem_text	Applies stemming to the words of the text. It was used in the clean text representation of the articles.
lemm_text	Applies Lemmatization to the words of the text. It was used in the clean text representation of the articles.
extract_sentiment	A function that uses TextBlob library to perform sentiment analysis in the raw/initial text.
weighted_sentiment	Weighted sentiment multiplies the polarity of the word with its significance based on its tf-idf score. In that way an important emotional word will have a higher impact on the whole sentiment of the text.
count_sentimental_words	Counts the positive and negative words in the text.
subjectivity	This function utilizes the "subjectivity" method of TextBlob library to calculate the subjectivity of a text.
count_outlier_sentences	I defined as big/small sentences those that are one standard deviation above/below the total average length of the sentences of the dataset. Then I count how many of them existed in each article.
syllables_count	Returns the total number of syllables of the raw text.
semantic_redundancy	It uses the vector representation of each sentence in the text and measures their similarity. If the similarity is higher than 95% I assume that there is redundancy in the meaning of two sentences.
pos_find	Extracting what part of speech is a word and the tag (additional information about POS etc. pos=verb tag= past verb) of it.
noun_phrase	Returns how many noun phrases exist in the text and their average length.
verb_phrase	Returns the number of verb phrases in the text.
passive_voice	Gets as input a list of sentences and returns the percentage of passive voice verbs by counting the nominal subjects and auxiliary verbs.
common_score_ratio	Takes a clean text and returns how many common words exist, excluding stop words. In order to define common words I used a corpus of common words found in Wiktionary's word frequency lists, which is compiled by statistically analyzing a sample of 29 million words used in English TV and movie scripts.
stop_words_ratio	The percentage of stop words in the whole text.

Table 8: Functions used to extract features from the text

7.3.2 Style-based features

Attribute type	Feature	Explanation
Quantity	#characters	Number of characters in the text
	#words	Number of words in text
	#small_words	Number of words with two or less syllables
	#big_words	Number of words having more than three syllables
	#sentences	Number of sentences
	#big_sentences	Number of big sentences
	#small_sentences	Number of small sentences
	#noun_phrases	Number of noun phrases
	#function_words	Number of function words
	#capitalwords	Number of capital words
	#mentions	Number of mentions in the article (@user)
	%punctuations	Punctuation characters
Complexity	avg_character_per_word	Average character per word
	avg_word_per_sent	Average word per sentence
	avg_punct_per_sentence	Average punctuation per sentence
	big_words_ratio	Ratio of big words over total words
	small_words_ratio	Ratio of small words over total words
	avg_word_syllables	Average syllables per word
	big_sentence_ratio	Ration of big sentences over number of senteces
	small_sentence_ratio	Number of small sentences over total sentences
	avg_len_noun_ phrase	Average length of noun phrase
	pausality	Is defined as #punctuations/#sentences
Uncertainty	%mverbs_freq	How many model verbs are being used.
	#certainty_words	Words that indicate centrainty (always, never, etc.)
	%non_factive_verbs	Percentage of factive verbs ('think','believe','suppose','expect', 'seem','appear','figure') Over all the verbs used
Subjectivity	subjectivity	As computed by the Textblob library.
	#sencory_words	Words that indicate sensation like feel, smell, taste
	#subjectivity_verbs	Words that indicate subjectivity like believe and feel
Non-immediacy	%first_person_singular	Percentage of first person singular references over total words
	%first_person_plural	Percentage of first person plural references over total words

	%second_person	Percentage of second person references over total words
	%third_person_singular	Percentage of third person singular references over total words
	%third_person_plural	Percentage of third person plural references over total words
	%passive_voice	Count how many of the used verbs are in passive indicating that the subject is being acted upon rather than doing something.
	#quotes	Number of quotes
Sentiment	sentiment	Text's sentiment as computed by Texblob library (takes values -1 for negative, 0 for neutral and 1 for positive)
	%neg_words	Percentage of negative words
	%positive_words	Percentage of positive words
	#exclamation_marks	Number of exclamation marks
	weighted_sentiment	Combination of words tf-idf score and its polarity
	emotiveness	Is defined as : (total#of adjectives +total#of adverbs)/(total#of nouns + total#of verbs)
Diversity	lexcl_diversity	The number of different words divided by the total number of words in the text
	noun_diversity	Number of unique nouns used
	adverb_diversity	Number of unique adverb used
	verb_diversity	Number of unique verb used
	adjective_diversity	Number of unique adjective used
	function_words_diversity	Diversity of function words
	redundancy	Defined as #function_words/#sentences
	semantic_redundancy	Number of sentences inside article that have more than 90% cosine similarity
Readability	ARI_score	Readability measure. Computes a grade level reading score. A higher score means a document takes a higher education level to read.
	Flesch_score	
	Gunfog_score	
	%common_score	Common words used in the text.
	%stopwords	Stop words in text.
Name Entity Count	%person	Known person reference
	%organization	Names of organizations like Google, Facebook etc.
	%countries	Country references.
	%location	Location references.
	%nationalities	Nationalities.
	%events	Named hurricanes, battles, wars, sports events, etc

	%dates	Absolute or relative dates or periods.
	%time	Times smaller than a day.
	%money	Words that indicate currency.
	%art	Titles of books, songs, etc.
	#links	Number of links
	%quantity	Quantitative references
	%events	Event references (hurricanes, elections etc.)
	%ordinal	Ordinal references (first, second, etc.)
	numerical	Numerical entities
	%percent	Percentages references
	%ordinal	Words that indicate order (first, second etc)
Part of speech counts	%nouns	The percentage of nouns in the text
	%verbs	The percentage of verbs In the text.
	%adverbs	The percentage of adverbs
	%adjectives	Percentage of adjectives
	%present_verbs	The percentage of present verbs
	%past_verbs	The percentage of past verbs
	%comperative_adv	The percentage of comparative adverbs
	%supperative_adv	The percentage of superlative adverbs
	%gerund_participle	Percentage of gerund particles
	%possesive_prn	Pronouns that indicate possession.
	%wh_possesive_prn	Wh-pronouns that indicate possession (which, whom, whose etc.)
	%clauses	% clauses used in text
	%modifiers	Percentage of modifiers over total words
	%proper_nouns	Percentage of proper nouns over

Table 9: Style-based features

The three readability metrics compute the grade level a reader should have in order to understand the text. The greater the level, the most complex the text.

Gunning fog Formula

$$\text{Grade level} = 0.4 * (\text{average sentence lenght} + \text{percentage of hard words}) \quad (11a)$$

$$\text{Percentage of hard words} = \frac{\text{words with more than syllables}}{\text{total words}} \quad (11b)$$

$$\text{Average sentence length} = \frac{\text{number of words}}{\text{number of sentences}} \quad (11c)$$

Flesch Formula

Another way to determine the text's difficulty level.

$$\text{Reading ease} = 206.8 - (1.015 * \text{AVG sentence length}) - (84.6 * \text{AVG word lenght}) \quad (12)$$

ARI Score

ARI score takes as consideration both word and sentence complexity and is given by the formula:

$$\text{ARI score} = 4.71 * \frac{\text{number of letters}}{\text{number of words}} + 0.7 * \frac{\text{number of words}}{\text{number of sentences}} \quad (13)$$

7.3.3 Creating word embeddings

Three different word embedding libraries were tested i) The pre-trained vectors of Google, trained on part of Google News dataset (about 100 billion words), a model that contains 300-dimensional vectors for 3 million words and phrases (public available on: https://code.google.com/archive/p/word2vec/) ii) spacy's word embeddings that includes 1 million different 300-dimensional vectors designed by using the GloVe algorithm, and iii) our own vectors trained using word2vec algorithm in the given dataset/corpus.

Testing all three of them, using 10-fold cross validation to a linear SVM classifier, we ended up to the spacy's representations that had the best accuracy results.

Google's word vectors	Spacy's word vectors	Trained word vectors
73.8%	**74.5%**	71.5%

Table 10: Testing different word embeddings

7.4 Association rules and parts of speech frequencies

Two important functionalities provided by spacy library is Name Entity Recognition (NER) and Part Of Speech Recognition (POSR).

Counting the frequencies of part of speech (POS) in a text has been used by many fake news detection studies. The information that can extract from a sentence referring to the parts of speech used and the dependencies between them.

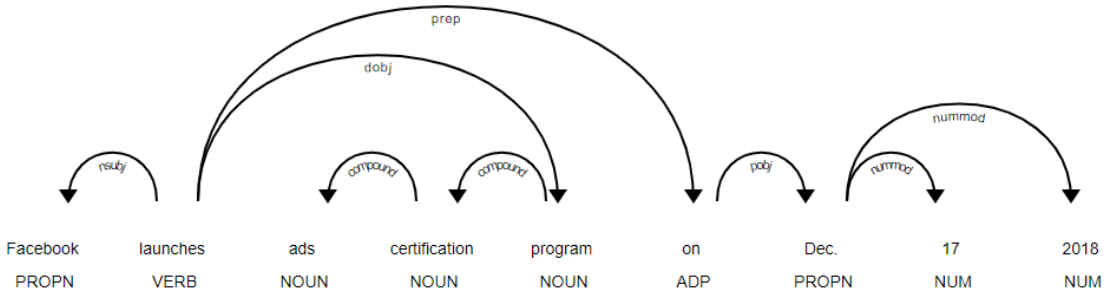

Figure 13: Part of speech and dependency tagging using spacy library

In that way, style based features like %verbs, %adverbs, noun diversity, were created and boosted the performance of the classification algorithms.

NER is the task where the words of the text are being classified in one of the categories below;

PERSON: People, including fictional. NORP: Nationalities or religious or political groups. FAC: Buildings, airports, highways, bridges, etc. ORG: Companies, agencies, institutions, etc. GPE:

Countries, cities, states. LOC: Non-GPE locations, mountain ranges, bodies of water. PRODUCT: Objects, vehicles, foods, etc. (Not services.) EVENT: Named hurricanes, battles, wars, sports events, etc. WORK_OF_ART: Titles of books, songs, etc. LAW: Named documents made into laws. LANGUAGE: Any named language. DATE: Absolute or relative dates or periods. TIME: Times smaller than a day. PERCENT: Percentage, including"%". MONEY: Monetary values, including unit. QUANTITY: Measurements, as of weight or distance. ORDINAL: "first", "second", etc. CARDINAL: Numerals that do not fall under another type.

For example, for the sentence "Facebook launches ads certification program on Dec. 17 2018" spacy was able to identify some name entities.

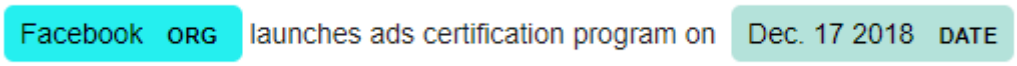

Figure 14: Name Entity Recognition (NER) with spacy library

Additional features, like date and person references, were used as features in the classification step.

Except from feature extraction, NER functionality was used to extracted all the name entities of the articles in a sentence level and, using Frequent Pattern Growth algorithm (FP Growth) to find entities and entities' association rules that are most common in fake and real news, trying this way to identify some differentiate characteristics in the content of the two categories.

Frequent Pattern Growth algorithm is an improvement of the Appriori algorithm that constructs association rules from a database by using a frequent pattern tree and not by creating pairs of candidate items comparing their frequencies together. This method provides a lot of scalability in large datasets with a significant change in the execution time opposed to Appriori.

The results are presented in the table below.

True news association rules	**False news association rules**
'PERSON'	'PERSON'
'DATE'	'DATE'
'ORG'	'ORG'
'CARDINAL'	'GPE'
'GPE'	'CARDINAL'
'WORK_OF_ART'	'DATE', 'PERSON'
'DATE', 'PERSON'	'ORG', 'PERSON'
'ORG', 'PERSON'	'DATE', 'ORG'
'DATE', 'ORG'	'DATE', 'ORG', 'PERSON'
'DATE', 'ORG', 'PERSON'	'GPE', 'PERSON'
'CARDINAL', 'PERSON'	'DATE', 'GPE'
'DATE', 'CARDINAL'	'ORG', 'GPE'
'ORG', 'CARDINAL'	'DATE', 'GPE', 'PERSON'
'DATE', 'CARDINAL', 'PERSON'	'ORG', 'GPE', 'PERSON'

'ORG', 'CARDINAL', 'PERSON'	'DATE', 'ORG', 'GPE'
'DATE', 'ORG', 'CARDINAL'	'DATE', 'ORG', 'GPE', 'PERSON'
'DATE', 'ORG', 'CARDINAL', 'PERSON'	'CARDINAL', 'PERSON'
'GPE', 'PERSON'	'DATE', 'CARDINAL'
'DATE', 'GPE'	'ORG', 'CARDINAL'
'ORG', 'GPE'	'DATE', 'CARDINAL', 'PERSON'
'DATE', 'GPE', 'PERSON'	'ORG', 'CARDINAL', 'PERSON'
'ORG', 'GPE', 'PERSON'	'DATE', 'ORG', 'CARDINAL'
'DATE', 'ORG', 'GPE'	'DATE', 'ORG', 'CARDINAL', 'PERSON'
'DATE', 'ORG', 'GPE', 'PERSON'	
'WORK_OF_ART', 'PERSON'	
'DATE', 'WORK_OF_ART'	
'DATE', 'WORK_OF_ART', 'PERSON'	

Table 11: Association rules extracted by the entities using FP Growth

The results indicate that, fake and real news haven't significant differences in the entities that use in their articles, for the specific dataset at least, confirming this way the state that fake news tend to have big similarity with real ones, limiting the linguistic approaches. The only entity that differentiates real news is the "work of art" item referring to titles of books, songs, etc.

7.5 Testing the different feature sets

7.5.1 Testing all the lexical features extracted

The first experimental results have been produced by using all the lexical features extracted from the texts. Because Support Vector Machine uses distances to classify its samples, scaling had a significant boost on its performance. Tree based algorithms on the other hand don't affect by scaling dissimilarities by their nature, so I overtook the scaling step in their testing phase.

	SVM	**Decision Tree**	**Random Forest**	**Gradient Boosting**
Accuracy	59.8%	61.5%	71.9%	70.8%
Accuracy (after scaling the values)	**72.1%**	-	-	-

Table 12: Performance using all the lexical features.

7.5.2 Finding the best lexical features

Mutual information between variables

The first method that I tested in order to end up in the optimal feature set, was the mutual information method. Mutual information between two variables describes the amount of information gained for the one variable by observing the other.

In this part, the mutual information between the dependent variable y and the independent variables X was calculated using sklearn's library "mutual_info_classif". The steps I followed were the below:

1) Use the full feature set
2) Sort the features based on mutual information score
3) Measure the performance of the linear Support Vector Classifier.
4) Drop the bottom two features
5) Reapet 3-4 until I have only one or none features

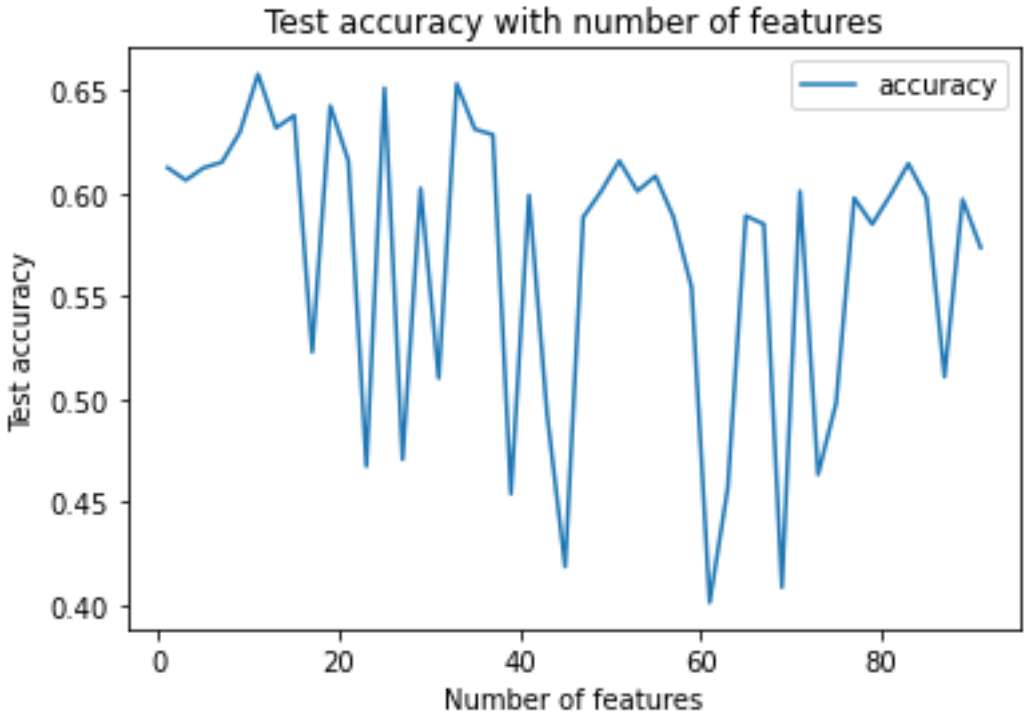

Figure 15: Finding optimal lexical subset using mutual information.

The maximum performance was equal to 65.7% and it corresponded to ten features shown below:

Best Lexical Features	
%present_verbs	weighted_sentiment
%mverbs_freq	noun_diversity
%punctuations	%dates
%adverbs	Gunfog_score
%clauses	%common_score

Table 13: Ten best performing features

Gini impurity criterion

Another method I tried in order to define an optimal subset was the Decision Tree's feature importance method based on the Gini impurity criterion. More specifically, Decision Tree calculates the total reduction of impurity for each feature and, the higher the decrease the more important the feature. The methodology I followed was again similar with the previous:

1) Use the full feature set

2) Measure the performance of the linear Support Vector.

3) Sort the features based on Decision Tree's feature importance.

4) Drop the bottom two features.

5) Repeat steps 2-4 until I don't have only one or none features are left.

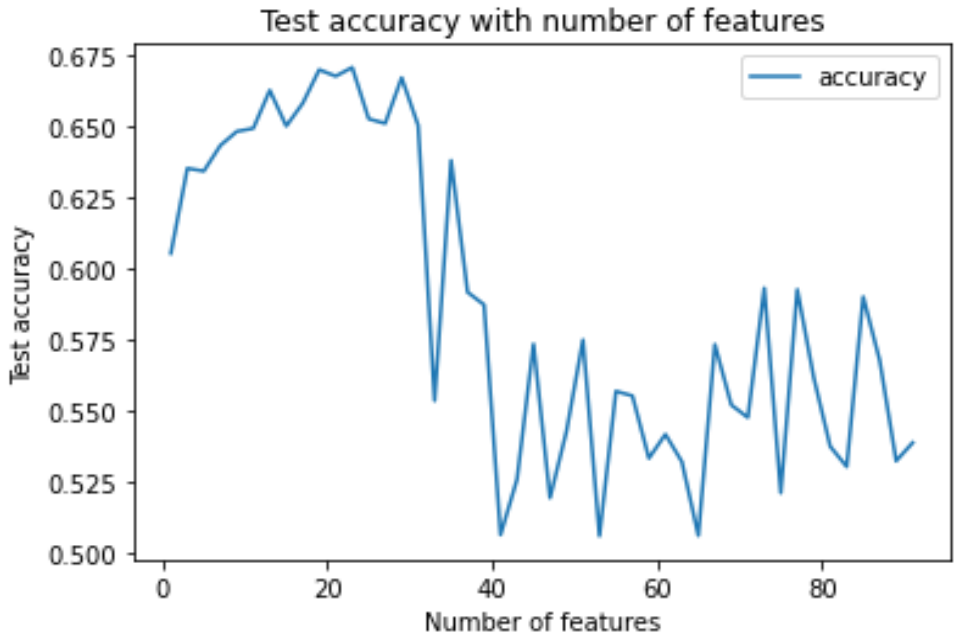

Figure 16: Finding optimal lexical subset using Gini impurity

The best lexical feature was consist of 23 variables and it corresponded to accuracy 67.02%

Best Lexical features	
%mverbs_freq	%third_person_singular
%first_person_singular	%art
%present_verbs	%third_person_plural
%possesive_prn	avg_punct_per_sentence
%adverbs	%gerund_participle
Redundancy	noun_diversity
%stopwords	verb_diversity
weighted_sentiment	subjectivity
%punctuations	big_words_ratio
%clauses	function_words_diversity
semantic_redundancy	avg_len_noun_phrase
%modifiers	

Table 14: The 23 best lexical features

Entropy criterion

Again using a linear Support Vector machine as base algorithm and the Decision Tree's feature importance, based on the entropy reduction this time, I've ended up in a different subset of optimal features. The results are shown below:

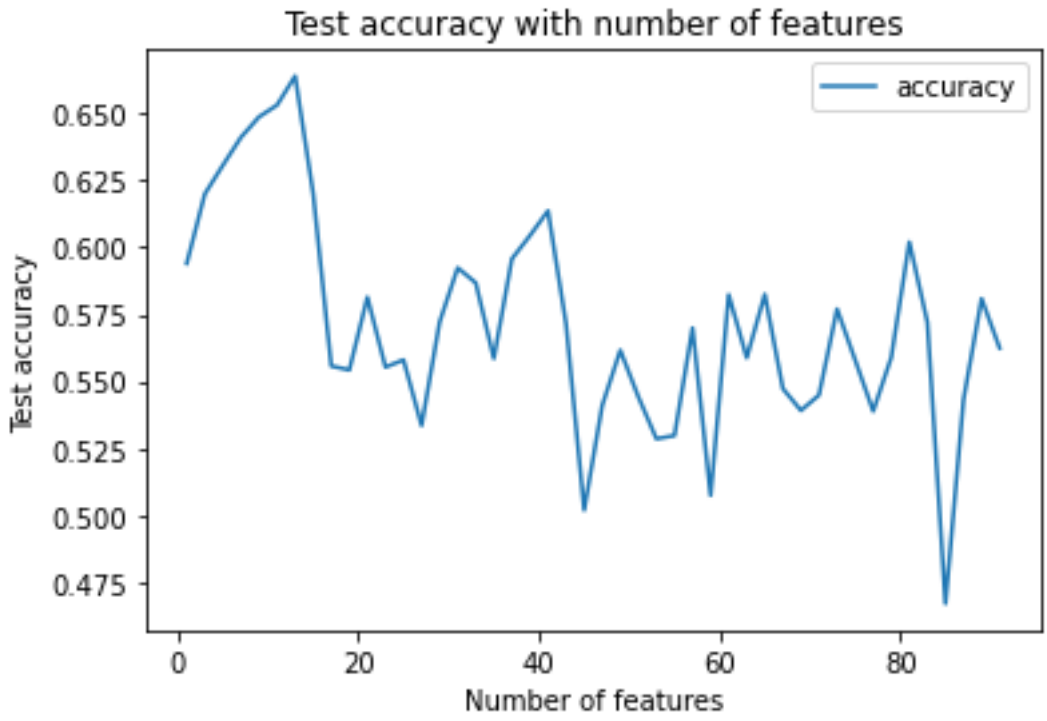

Figure 17: Results of the entropy criterion

The best performance was achieved with 12 features and it was 66.4%. The optimal subset was:

Best Lexical features	
Redundancy	%verbs
%mverbs_freq	%person
pos_polarity	entities
%possesive_prn	noun_diversity
Subjectivity	%first_person_singular
%present_verbs	%organization

Table 15: The best 12 features of the method

Results

The Gini impurity criterion gave the best performance for the base estimator so its resulted 23 - feature subset will going to be used in the next experiments as “best lexical features”.

7.5.3 Testing the best lexical features

	SVM	Decision Tree	Random Forest	Gradient Boosting
Accuracy	58.9%	62.0%	**72.3%**	70.4%
Accuracy (after scaling the values)	71.7%	-		-

Table 16: Accuracy results using the best features

7.5.4 Testing word embeddings

First I needed to choose the best preprocessing method for text which returns the best accuracy results. The base algorithm used for testing was a linear SVM.

Text preprocessing method	Accuracy score
Raw text representation	71.3%
Clean Text representation	**74.7%**
Apply Lemmatization to clean text	74.2%
Apply Stemming to clean text	73.6%

Table 17: Testing different text representations for word embeddings

The next step was to use PCA to reduce the dimensions of the 300-dimensional embeddings trying to improve the performance. The best results were achieved for 180 dimensions using the clean text word representation.

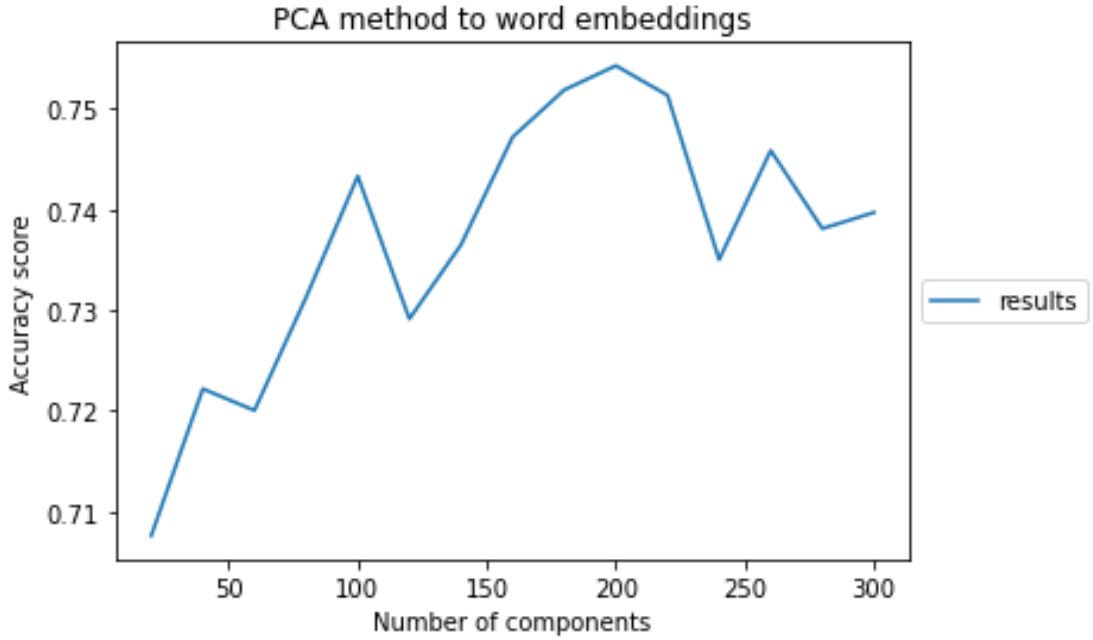

Figure 18: Reducing the dimensions of word embeddings using PCA

	SVM	Decision Tree	Random Forest	Gradient Boosting
Accuracy	74.7%	63.8%	75.3%	75.3%
Accuracy (reduced dimensions)	**77.2%**	65.5%	73.5%	74.1%

Table 18: Accuracy results using the word embeddings

7.5.6 Combine lexical features with word embeddings

	SVM	Decision Tree	Random Forest	Gradient Boosting
Best lexical features with word embeddings	**76.2%**	65.8%	75.8%	**76.2%**
Best lexical features with PCA word embeddings	75.8%	65.2**%**	75.5%	75.9%

Table 19: Accuracy results by combining best lexical features with word embeddings.

7.5.7 Testing Naïve Bayes using the three different word representations

For Naïve Bayes algorithm I had to find the best text representation and then I also tested two basic ensemble learning techniques; Boosting and Bagging.

Naïve Bayes	
Using clean corpus	73.29%
Lemmatized corpus	72.95%
Stemmed corpus	72.56%
Ensemble method: Boosting	
Using clean corpus	**74.2%**
Ensemble method: Bagging	
Using clean corpus	73.7%

Table 20: Naïve Bayes results

7.6 Testing Neural Networks

One of the biggest challenges referring to the neural networks training was the hyper parameter tuning. There was a big number of parameters and combinations that needed to be tested before I ended up in an optimal version of the model. To do that I applied the GridSearchCV function of sklearn library, in order to define the best parameters for multilayer perceptron. For the convolutional neural network (CNN) and the long-short term memory (LSTM) I needed to use nested "for" loops to test a big number of activation functions and optimizers. Also, for the embedding layer I used a technique called "transfer learning" for which I used the pre-trained word embeddings as weights in the embedding layer.

7.6.1 Multilayer Perceptron (MLP)

The features and the values tested for multilayer perceptron via grid search were the below from which the best combination of parameters are shown in bold:

$$hidden\ layer\ sizes = \begin{cases} (50,50,50) \\ (50,100,50) \\ (\mathbf{100},\mathbf{50},\mathbf{10}) \end{cases}$$

$$solver = \begin{cases} sgd \\ \boldsymbol{adam} \\ adamax \end{cases} \qquad activation = \begin{cases} \boldsymbol{tanh} \\ relu \end{cases}$$

$$aplha = \begin{cases} 0.0001 \\ 0.05 \end{cases} \qquad learning\ rate = \begin{cases} constant \\ \boldsymbol{adaptive} \end{cases}$$

The best score was 62.97% using all the lexical features non-scaled.

	Testing all the lexical features extracted	**Testing the best lexical features**
Non-scaled values	60.3%	65.6%
Scaled values	68.1%	69.1%
	Testing word embeddings	**Combine best lexical features with word embeddings**
Whole feature set	73.4%	73.66%
Reduced dimensions	75.4%	**76.5**%

Table 21: Accuracy score of MLP in different feature sets

7.6.2 Convolutional Neural Network (CNN)

Finding the best activation functions

In order to find the best hyper parameters I used a simple shallow neural network constructed by four layers. First, there was the embedding layer which consists of three hundred neurons, then there was the Convolutional layer, a max pooling layer and finally, a dense layer with the activation function.

Layer	**Output dimension**
Embedding Layer	300
Convolutional Layer	128
Max pooling	128
Dense Layer	1

For choosing the best combination of activation functions and optimizer I performed a nested for loop testing all the combinations of:

$$CNN\ activation\ functions = \begin{cases} \boldsymbol{relu} \\ sigmoid \\ softmax \\ softsign \\ tanh \\ exponential \end{cases}$$

$$Dense\ layer\ activation\ functions = \begin{cases} relu \\ \boldsymbol{sigmoid} \\ softmax \\ softsign \\ tanh \\ exponential \end{cases}$$

$$Optimizers\ = \begin{cases} \boldsymbol{adam} \\ adadelta \\ adamax \\ SGD \end{cases}$$

Fifteen epochs were used for each of the combinations and the best performance was given with relu activation function, sigmoid activation function for the dense layer, and adam optimizer, with performance 75.6%

In the next step a couple of different architectures were tested to find the optimal one.

Testing architectures

1. Architecture

First architecture was the pre mentioned shallow network.

Layer	Output dimension
Embedding Layer	300
Convolutional Layer	128
Max pooling	128
Dense Layer	1

The performance of it was 75.6%.

2. Architecture

For the second architecture I added more convolutional layers creating a reverse pyramid schema inside the hidden layers. The accuracy didn't change (75.62%).

Layer	Output dimension
Embedding Layer	300
Convolutional Layer	128
Convolutional Layer	70
Convolutional Layer	20
Max pooling	128
Dense Layer	1

3. Architecture

In the third architecture I added two additional convolutional layers at the top with 20 and 70 neurons creating the schema of rhombus.

Layer	Output dimension
Embedding Layer	300
Convolutional Layer	20
Convolutional Layer	70
Convolutional Layer	128
Convolutional Layer	70
Convolutional Layer	20
Max pooling	128
Dense Layer	1

The accuracy of this architecture dropped to 72.4%

4. Architecture

The next architecture consisted of three convolutional layers with the same number of neurons scoring accuracy equal to 72.5%.

Layer	Output dimension
Embedding Layer	300
Convolutional Layer	128
Convolutional Layer	128
Convolutional Layer	128
Max pooling	128
Dense Layer	1

5. Architecture

Eventually the initial architecture combined with one aditional dense layer brought promising results 76.6%

Layer	Output dimension
Embedding Layer	300
Convolutional Layer	128
Max pooling	128
Dense Layer	28
Dense Layer	1

6. Architecture

Finally, the most successful architecture that achieved performance equal to 79.2%.

Layer	Output dimension
Embedding Layer	300

Convolutional Layer	300
Convolutional Layer	128
Max pooling	128
Dense Layer	28
Dense Layer	1

7.6.3 Long-Short Term Memory Network (LSTM)

Finding the best activation functions

Similar with the methodology for CNN, the first step was to find the best combination of activation function and optimizer. The tests were conducted in a pretty simple architecture of LSTM neural network and the tested parameters were;

$$Dense\ layer = \begin{cases} relu \\ sigmoid \\ softmax \\ softsign \\ tanh \\ exponential \end{cases} \qquad Optimizer = \begin{cases} relu \\ sigmoid \\ softmax \\ softsign \\ tanh \\ exponential \end{cases}$$

And the architecture was:

Layer	Output layer
Embedding Layer	300
LSTM Layer	300
Dropout	300
Flatten	300
Dense Layer	1

The best combination was the sigmoid activation function and the adamax optimizer.

Testing architectures

1. Architecture

The first architecture was the pre mentioned one with only one LST layer, a dropout layer, a flatten layer and the dense layer that calculates the output. The accuracy score was equal to 71.5%.

Layer	Output dimension
Embedding Layer	300
LSTM Layer	300
Dropout	300
Flatten	300

Dense	1

2. Architecture

In the second architecture I added one additional dense layer with 30 neurons before the output layer. This schema achieved the best accuracy equal to 75.6%

Layer	Output dimension
Embedding Layer	300
LSTM Layer	300
Dropout	300
Flatten	300
Dense Layer	30
Dense Layer	1

3. Architecture

The third architecture used the previous one, with the two dense layers, and added one more LSTM layer with 128 neurons. The accuracy dropped to 73.8%

Layer	Output dimension
Embedding Layer	300
LSTM Layer	300
LSTM Layer	300
Dropout	300
Flatten	300
Dense Layer	30
Dense Layer	1

4. Architecture

Similar results were also achieved by using three identical LSTM layers. The accuracy was equal to 73.5%

Layer	Output dimension
Embedding Layer	300
LSTM Layer	300
LSTM Layer	300
LSTM Layer	300
Dropout	300
Flatten	300
Dense Layer	30
Dense Layer	1

8 Discussion

In the experimental processing we were able to extract 89 style-based features and achieving 72.1% performance using the Decision Tree algorithm.

By using the mutual information, entropy and Gini impurity metrics, we were able to end up in an optimal subset of 23 linguistic features, increasing the performance of SVM to 72.5%. We are going to discuss further those features.

Percentage of modal verbs: In general, modal verbs indicate uncertainty (would, could, might etc.) and most often been used by deceivers. This happens because the deceivers are not sure about the information they propagate, and many times, they know it's not true so they tend to make hypotheses and imply correlations about events, people and facts that do not have clear connection between them (Zhou, L., 2003).

Statistic	Real news	Fake news
Mean	3.93 %	4.54 %
Median	3.86 %	4.46 %
Standard deviation	1.85 %	1.85 %

Table 22: Statistics of "% modal verbs" for the two classes

Also used in: (Khan, J. Y., et al., 2019; Horne, B. D., & Adali, S., 2017; Zhou, X., et al., 2020; Gravanis, G., et al., 2019).

Third person singular / Third person plural/**First Person Singular**: The use of first person singular is more often in real articles and not in fake ones. This is due to the fact that writers who try to deceive readers tend to separate themselves from the information they propagate (Zhou, L., 2003).

Previous studies that also used this feature: (Horne, B. D., & Adali, S., 2017; Zhou, X., et al., 2020; Gravanis, G., et al., 2019).

Statistic	3rd person singular		3rd person plural		1st Person Singular	
	Real news	Fake news	Real news	Fake news	Real news	Fake news
Mean	1.957%	2.318%	0.322%	0.47%	0.883%	0.667%
Median	1.829%	2.185%	0.155%	0.292%	0.394%	0.167%
Standard deviation	1.428%	1.538%	0.524%	0.606%	1.86	1.47

Table 23: Statistics of the features for the two classes

Percentage of present verbs: Syntactic features, such as counting parts of speech, is a technique broadly used in fake news studies, so the specific feature isn't correlated direct with deceiving or not behaviors (Shu, K., et al., 2017).

Used by the studies: (Horne, B. D., & Adali, S., 2017; Zhou, X., et al., 2020; Gravanis, G., et al., 2019).

Percentage of possessive pronouns: The possessive pronouns are often used to emphasize on the separate perspectives and ideas between two sides, often the author's and a third person's or party's (Horne, B. D., & Adali, S., 2017; Zhou, X., et al., 2020).

Adverbs: The number of adverbs was stated as a differentiative characteristic by the study of Horne, B. D., & Adali, S., since fake news tend to use more adverbs that real ones. Our experiment agrees with this state.

	adverbs		% possessive pronouns		% present verbs	
Statistic	**Real news**	**Fake news**	**Real news**	**Fake news**	**Real news**	**Fake news**
Mean	4.006 %	4.448 %	1.747 %	1.8 %	8.623 %	9.665 %
Median	3.922 %	4.335 %	1.652 %	1.699 %	8.333 %	9.528 %
Standard deviation	1.958 %	2.294 %	1.164 %	1.169 %	3.403 %	3.62 %

Table 24: Statistics of the features for the two classes

Weighted sentiment: This metric is based on the state that fake news writers tend to use more emotional vocabulary in their texts trying to persuade the readers (Ruchansky, N., Seo, S., & Liu, Y., 2017). In order to enhance this metric I also added a weight on each word based on its importance in the text using tf-idf score. The feature is calculated by the formula:

$$\text{weighted sentiment} = |\text{sentiment polarity}| * (1 + \text{tfidf score}) \quad (13)$$

In that way, the metric captures the sentiment of the word and adds the additional importance. The text's final score is computed by adding all the words' values. Values near zero indicate neutrality were more positive values indicate emotional vocabulary.

This exact feature hasn't used, at least, in the studies I included in my literature review, however sentiment based features are common in fake news detection tasks (Khan, J. Y., et al., 2019).

Statistic	**Real news**	**Fake news**
Mean	239.898	300.477
Median	46.769	38.679
Standard deviation	946.311	1647.084

Table 25: Weighted sentiments statistics for the two classes

What needs to be noted here is the big difference between the mean and the median value which indicates outliers in the values. This can be also seen in the graph bellow.

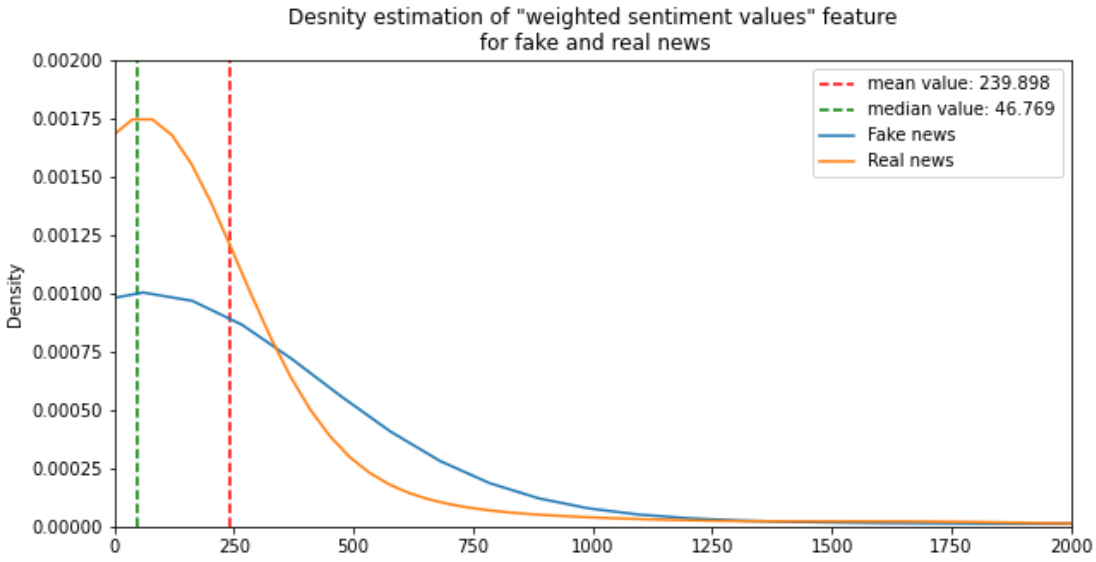

Figure 19

Punctuation-based metrics: Punctuations, which include characters such as periods, commas, dashes, question marks and exclamation marks, were notated also in previous researches as useful deception indicators (Rubin, V. L., et al., 2015; Horne, B. D., & Adali, S., 2017; Zhou, X., et al., 2020). In my experiment I used two punctuation based features, which were also important classification features.

i) *Percentage of punctuations*

ii) *Average punctuation per sentence*

	Percentage of punctuations		**Average punctuation per sentence**	
Statistic	**Real news**	**Fake news**	**Real news**	**Fake news**
Mean	13.68 %	13.542 %	3.9	4.14
Median	13.683 %	13.481 %	3.50	3.59
Standard deviation	3.467 %	3.256 %	4.71	4.05

Table 26: Statistic results for the features

The percentage of punctuations was expected to be lower in fake news, as denoted in the study of Horne, B. D., & Adali, S., but the higher number of the average punctuation per sentence is an issue that needs to be addressed.

Redundancy: The "redundancy" metric was defined as the number of function words divided by the number of sentences, and it's a metric of diversity. This implies that writers who tend to repeat their content don't have enough information to support their claims, which is very often in fake news (Horne, B. D., & Adali, S., 2017). My experimental results agree with this state.

Semantic redundancy: This metric is also based on the state that fake news tend to repeat their basic claims in the main body of the article without adding content or evidence about their claim. In order to measure the redundancy in a text I used the vector representation of

each sentence in an article and count the similarities between each one of them. I assume that two sentences with similarity higher than 95% contain the same information.

	Redundancy		Semantic redundancy	
Statistic	**Real news**	**Fake news**	**Real news**	**Fake news**
Mean	0.775	0.884	1.955 %	3.721 %
Median	0.705	0.75	0.483 %	1.053 %
Standard deviation	0.591	0.741	4.791 %	8.909 %

Table 27: Statistic results for Redundancy and Semantic redundancy

Subjectivity: The subjectivity metric was expected to be higher in the fake news since it indicates that the author relies on his/hers point of view and not in the objective facts. The feature, as defined by the TextBlob library, had the same mean value but had higher median for fake news. However, the difference doesn't seem quite important.

Statistic	**Real news**	**Fake news**
Mean	0.447	0.447
Median	0.454	0.456
Standard deviation	0.118	0.114

Table 28: Statistic results for subjectivity

Noun/Verb/Function words diversity: The diversity of these three parts of speech is a good indicator of the linguistic level of the article, and therefore, of the writer. A more diverse vocabulary indicates deeper knowledge of the subject. As confirmed by the study of (Zhou, L., et al., 2003) deceivers tend to use less lexical diversity. Lexical diversity also used by the studies of (Horne, B. D., & Adali, S., 2017; Zhou, X., et al., 2020).

In the specific experiment, noun diversity agrees with the theoretical studies, were verb and function diversity didn't. Although the differences are quite small, this is an issue that needs further investigation in future work.

	Noun diversity		Verb diversity		Function words diversity	
Statistic	**Real news**	**Fake news**	**Real news**	**Fake news**	**Real news**	**Fake news**
Mean	0.772	0.762	0.836	0.838	0.689	0.694
Median	0.782	0.775	0.857	0.857	0.692	0.699
Standard deviation	0.143	0.145	0.137	0.134	0.227	0.215

Table 29: Statistic results for noun, verb and function words diversity

Percentage of stopwords: Stop words are words used very often (the, that, he, have etc.) and don't contribute significant in the meaning of the text, that's why they are ignored in most of the text processing methods. The increased use stopwords, once more indicate lack of content, mainly observed in fake news (Horne, B. D., & Adali, S., 2017).

Big words ratio: Big words ratio was calculated by divide the number of big words with the total words used in the article. The use of bigger words is a metric of the vocabulary's complexity, since bigger words also tend to be more complicate showing a deeper understanding of the topic less often used by deceivers (Burgoon, J. K., et al., 2003). This feature was also important indicator in the studies of (Pérez-Rosas, V., et al., 2017; Gravanis, G., et al., 2019)

Average length of noun phrase: The average length of noun phrase is also a metric of complexity which should be higher for real articles. This state was proved true for this dataset also. This metric also captures the complexity in sentence level since more words per phrase means the average sentence structure complexity is high (Horne, B. D., & Adali, S., 2017; Gravanis, G., et al., 2019).

	Percentage of stop words		Big words ratio		Average length of noun phrase	
Statistic	**Real news**	**Fake news**	**Real news**	**Fake news**	**Real news**	**Fake news**
Mean	0.362 %	0.375 %	0.064 %	0.067 %	1.977	1.915
Median	0.367 %	0.378 %	0.06 %	0.062 %	1.952	1.887
Standard deviation	0.061 %	0.056 %	0.035 %	0.038 %	0.297	0.289

Table 30: Statistics for percentage of stopwords, big words ratio and average length of noun phrase

The experimental results about stop words and average length of noun phrase agree with the theoretical background. Big words ratio on the other hand didn't have the expected behavior and needs further investigation.

Percentage of art references: This was one characteristic that didn't run into other studies findings. The use reference to works of art must be a characteristic of the specific dataset.

Entities: The "entities" feature was counting the references of person, organization, countries, location, nationalities, events and work of art. The high importance of this feature seems to be a characteristic of the dataset and also the fact that it also contains the "art" feature which was important differentiate characteristic of the two classes as it turned out from both feature importance and association rules mining.

	Percentage of art references		Entities	
Statistic	**Real news**	**Fake news**	**Real news**	**Fake news**
Mean	0.052%	0.036 %	0.664 %	0.711 %
Median	0.027 %	0.018 %	0.683 %	0.727 %
Standard deviation	0.089 %	0.065 %	0.181 %	0.166 %

Table 31: Percentage of art references and entities

A strange outcome was that none of the readability measures was among the 23 best lexical features, but if we take as consideration that these metrics use other more fundamental features which appear in the subset, it's more logical.

By observing the density distributions of the two most important lexical features (% modal verbs frequency, % first person singular) we observe that they don't have significant differences, there is a high amount of overlap instead making them difficult to separate. This is also the reason why, by themselves, lexical feature sets had 72% accuracy score, where

when combined with the word embeddings information increased to 77%.

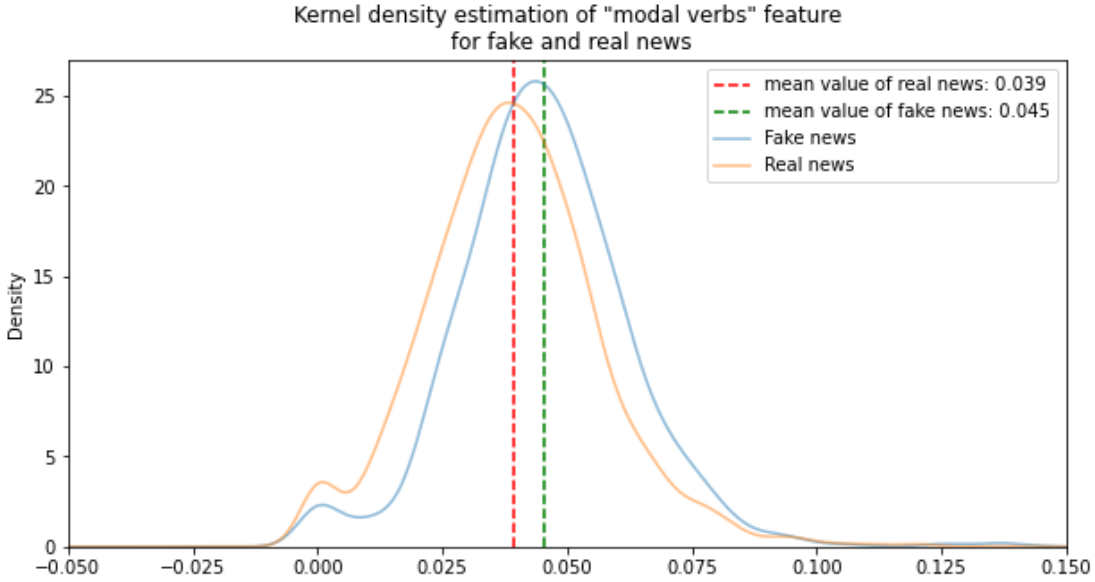

Figure 20

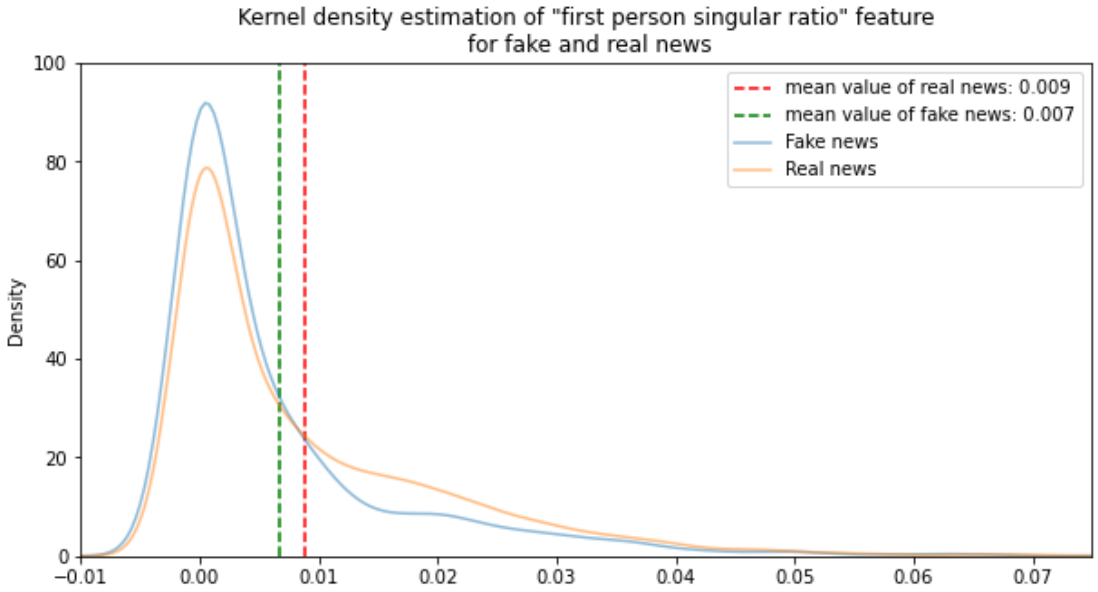

Figure 21

Although, style-based features provide interesting explainable results, their performance was overpassed by the embedding representation of the texts. More specifically, SVM achieved 77.2% accuracy using the embeddings with 180 dimensions, and 76.2% by combining the full 300-dimensional embeddings with the 23 best lexical features.

Continuing, the combination of specie's Name Entity Recognition functionality with the Frequent Pattern Growth algorithm on the extraction of association rules of name entities in real and fake news, was able to provide results which, once again, support the claim that fake and real news have quite similar structures.

Finally, the best performance was achieved by convolutional neural network, agreeing with the state of the art trend to utilize ANNs in text classification tasks (Zhang, Y., et al., 2015).

Layer	Output dimension
Embedding Layer	300
Convolutional Layer	300
Convolutional Layer	128
Max pooling	128
Dense Layer	28
Dense Layer	1
Performance	**79.2%**

Table 32: Architecture of the best performing CNN

For LSTM neural network, which is quite promising for text classification tasks, the maximum performance was equal to 75.2%. However, both of the neural networks have a lot of potential since there are still a lot of combinations and different architectures that could be tested.

Layer	Output dimension
Embedding Layer	300
LSTM Layer	300
Dropout	300
Flatten	300
Dense Layer	30
Dense Layer	1
Performance	**75.3%**

Table 33: Architecture of the best performing LSTM

9 Conclusions and future work

The scope of this thesis was to review the state of the art on fake news detection methods, denote the most important stakeholders and challenges, and find an efficient way to perform text classification by utilizing the linguistic information of the content. We used both classic and non-classic (ANNs) algorithms with the second ones outperform the first. Since ANNs brought promising results, for future work, we would focus on the research that has been done in the field of text classification using ANNs. Also, the neural networks that we utilized do not consider quite deep, since the maximum number of layers was four. This leaves space for additional searching and experimentations with deep neural networks.

On the part of style based features, most of the psychological studies in which those features are based contacted in real time communication, where fake articles belong in a different category. Although they bring quite satisfying explainable results, the base line theory does not fit so well since there are a lot of differences in the process of writing a fake text and telling lies in real time and first person. Also there are some results that do not agree with previous studies that need further investigation. In general, the statistical differences between the most important features of the two classes were quite small. An additional statistical analysis

could be applied, like t-statistic test to normalized data, to check if the means of two sample/classes distributions are significantly different from each other.

Finally, it would be very interesting to test the style based method in some of the most common fake news datasets, for which sufficient literature already exists, and compare our results.

References

Afroz, S., Brennan, M., & Greenstadt, R. (2012, May). Detecting hoaxes, frauds, and deception in writing style online. In 2012 IEEE Symposium on Security and Privacy (pp. 461-475). IEEE.

Aizawa, A. (2003). An information-theoretic perspective of tf–idf measures. Information Processing & Management, 39(1), 45-65.

Alzanin, S. M., & Azmi, A. M. (2018). Detecting rumors in social media: A survey. Procedia computer science, 142, 294-300.

Apté, C., & Weiss, S. (1997). Data mining with decision trees and decision rules. Future generation computer systems, 13(2-3), 197-210.

Boehm, L. E. (1994). The validity effect: A search for mediating variables. *Personality and Social Psychology Bulletin*, *20*(3), 285-293.

Breiman, L. (1996). Bagging predictors. Machine learning, 24(2), 123-140.

Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.

Burgoon, J. K., Blair, J. P., Qin, T., & Nunamaker, J. F. (2003, June). Detecting deception through linguistic analysis. In International Conference on Intelligence and Security Informatics (pp. 91-101). Springer, Berlin, Heidelberg.

Ciampaglia, G. L., Shiralkar, P., Rocha, L. M., Bollen, J., Menczer, F., & Flammini, A. (2015). Computational fact checking from knowledge networks. PloS one, 10(6), e0128193.

Csáji, B. C. (2001). Approximation with artificial neural networks. Faculty of Sciences, Etvs Lornd University, Hungary, 24(48), 7.

Freund, Y., Schapire, R.E. (1999) Large Margin Classification Using the Perceptron Algorithm. Machine Learning 37, 277–296 (1999). https://doi.org/10.1023/A:1007662407062

Gottfried, J., & Shearer, E. (2017). Americans' online news use is closing in on TV news use. Pew Research Center, 7.

Hofmann, M. (2006). Support vector machines-kernels and the kernel trick. Notes, 26(3).

Howard, P. N., Bolsover, G., Kollanyi, B., Bradshaw, S., & Neudert, L. M. (2017). Junk news and bots during the US election: What were Michigan voters sharing over Twitter. CompProp, OII, Data Memo.

Jain, A. K., Mao, J., & Mohiuddin, K. M. (1996). Artificial neural networks: A tutorial. Computer, 29(3), 31-44.

Jamieson, K. H., & Cappella, J. N. (2008). Echo chamber: Rush Limbaugh and the conservative media establishment. Oxford University Press.

Jin, Z., Cao, J., Zhang, Y., & Luo, J. (2016, March). News verification by exploiting conflicting social viewpoints in microblogs. In Thirtieth AAAI conference on artificial intelligence.

Kiesel, J., Mestre, M., Shukla, R., Vincent, E., Adineh, P., Corney, D., Stein, B. and Potthast, M., (2019 June). Semeval-2019 task 4: Hyperpartisan news detection. In Proceedings of the 13th International Workshop on Semantic Evaluation (pp. 829-839).

Kolbusz, J., Rozycki, P., & Wilamowski, B. M. (2017, June). The study of architecture MLP with linear neurons in order to eliminate the "vanishing gradient" problem. In International Conference on Artificial Intelligence and Soft Computing (pp. 97-106). Springer, Cham.

Kumar, S., West, R., & Leskovec, J. (2016, April). Disinformation on the web: Impact, characteristics, and detection of wikipedia hoaxes. In Proceedings of the 25th international conference on World Wide Web (pp. 591-602).

Kuran, T., & Sunstein, C. R. (1998). Availability cascades and risk regulation. *Stan. L. Rev.*, *51*, 683.

Le, Q. V. (2015). A tutorial on deep learning part 2: Autoencoders, convolutional neural networks and recurrent neural networks. Google Brain, 1-20.

Ling, W., Dyer, C., Black, A. W., & Trancoso, I. (2015). Two/too simple adaptations of word2vec for syntax problems. In Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (pp. 1299-1304).

Liu, Y., & Xu, S. (2016). Detecting rumors through modeling information propagation networks in a social media environment. IEEE Transactions on computational social systems, 3(2), 46-62.

Nickerson, R. S. (1998). Confirmation bias: A ubiquitous phenomenon in many guises. Review of general psychology, 2(2), 175-220.

Noriega, L. (2005). Multilayer perceptron tutorial. School of Computing. Staffordshire University.

Orso, D., Federici, N., Copetti, R., Vetrugno, L., & Bove, T. (2020). Infodemic and the spread of fake news in the COVID-19-era. European Journal of Emergency Medicine.

Potthast, M., Gollub, T., Komlossy, K., Schuster, S., Wiegmann, M., Fernandez, E. P. G., ... & Stein, B. (2018, August). Crowdsourcing a large corpus of clickbait on twitter. In Proceedings of the 27th international conference on computational linguistics (pp. 1498-1507).

Potthast, M., Kiesel, J., Reinartz, K., Bevendorff, J., & Stein, B. (2017). A stylometric inquiry into hyperpartisan and fake news. *arXiv preprint arXiv:1702.05638*.

Pronin, E., Kruger, J., Savtisky, K., & Ross, L. (2001). You don't know me, but I know you: The illusion of asymmetric insight. Journal of Personality and Social Psychology, 81(4), 639.

Quinlan, J. R. (1987, August). Generating production rules from decision trees. In ijcai (Vol. 87, pp. 304-307).

Raschka, S. (2014). Naive bayes and text classification i-introduction and theory. arXiv preprint arXiv:1410.5329.

Reis, J. C., Correia, A., Murai, F., Veloso, A., & Benevenuto, F. (2019). Supervised learning for fake news detection. *IEEE Intelligent Systems*, *34*(2), 76-81.

Rokach, L., & Maimon, O. Z. (2008). Data mining with decision trees: theory and applications (Vol. 69). World scientific.

Rubin, V. L., Conroy, N., Chen, Y., & Cornwell, S. (2016, June). Fake news or truth? using satirical cues to detect potentially misleading news. In Proceedings of the second workshop on computational approaches to deception detection (pp. 7-17).

Salzberg, S. L. (1994). C4. 5: Programs for machine learning by j. ross quinlan. morgan kaufmann publishers, inc., 1993.

Schapire, R. E. (1990). The strength of weak learnability. Machine learning, 5(2), 197-227.

Shahsavari, S., Holur, P., Tangherlini, T. R., & Roychowdhury, V. (2020). Conspiracy in the time of corona: Automatic detection of covid-19 conspiracy theories in social media and the news. arXiv preprint arXiv:2004.13783.

Shu, K., Mahudeswaran, D., Wang, S., Lee, D., & Liu, H. (2018). Fakenewsnet: A data repository with news content, social context and dynamic information for studying fake news on social media. arXiv preprint arXiv:1809.01286, 8.

Shu, K., Wang, S., & Liu, H. (2018, April). Understanding user profiles on social media for fake news detection. In 2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR) (pp. 430-435). IEEE.

Shu, K., Zhou, X., Wang, S., Zafarani, R., & Liu, H. (2019, August). The role of user profiles for fake news detection. In Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (pp. 436-439).

Staudemeyer, R. C., & Morris, E. R. (2019). Understanding LSTM--a tutorial into Long Short-Term Memory Recurrent Neural Networks. arXiv preprint arXiv:1909.09586.

Suykens, J. A., & Vandewalle, J. (1999). Least squares support vector machine classifiers. Neural processing letters, 9(3), 293-300.

Tzirakis, P., & Tjortjis, C. (2017). T3C: improving a decision tree classification algorithm's interval splits on continuous attributes. Advances in Data Analysis and Classification, 11(2), 353-370.

Ullah, I., Raza, B., Malik, A. K., Imran, M., Islam, S. U., & Kim, S. W. (2019). A churn prediction model using random forest: analysis of machine learning techniques for churn prediction and factor identification in telecom sector. IEEE Access, 7, 60134-60149.4

Wang, L. X., Ramachandran, A., & Chaintreau, A. (2016, April). Measuring click and share dynamics on social media: a reproducible and validated approach. In *Tenth International AAAI Conference on Web and Social Media*.

Wu, J. (2017). Introduction to convolutional neural networks. National Key Lab for Novel Software Technology. Nanjing University. China, 5, 23.

Wu, L., & Liu, H. (2018, February). Tracing fake-news footprints: Characterizing social media messages by how they propagate. In Proceedings of the eleventh ACM international conference on Web Search and Data Mining (pp. 637-645).

Wu, L., Li, J., Hu, X., & Liu, H. (2017, June). Gleaning wisdom from the past: Early detection of emerging rumors in social media. In Proceedings of the 2017 SIAM international conference on data mining (pp. 99-107). Society for Industrial and Applied Mathematics.

Zannettou, S., Caulfield, T., De Cristofaro, E., Sirivianos, M., Stringhini, G., & Blackburn, J. (2019, May). Disinformation warfare: Understanding state-sponsored trolls on Twitter and their influence on the web. In Companion proceedings of the 2019 world wide web conference (pp. 218-226).

Zannettou, S., Sirivianos, M., Blackburn, J., & Kourtellis, N. (2019). The web of false information: Rumors, fake news, hoaxes, clickbait, and various other shenanigans. Journal of Data and Information Quality (JDIQ), 11(3), 1-37.

Zhang, Y., & Wallace, B. (2015). A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. arXiv preprint arXiv:1510.03820.

Zhou, L., Twitchell, D. P., Qin, T., Burgoon, J. K., & Nunamaker, J. F. (2003, January). An exploratory study into deception detection in text-based computer-mediated communication. In 36th Annual Hawaii International Conference on System Sciences, 2003. Proceedings of the (pp. 10-pp). IEEE.

Zubiaga, A., Aker, A., Bontcheva, K., Liakata, M., & Procter, R. (2018). Detection and resolution of rumours in social media: A survey. *ACM Computing Surveys (CSUR)*, *51*(2), 1-36.

Appendix

Semantic redundancy function

```
import spacy
import pandas as pd
import numpy as np
from sklearn.metrics.pairwise import cosine_similarity
nlp = spacy.load("en_core_web_lg")

def semantic_redundancy(corpus):
    ''' Sentence to Vector '''
    w2v = pd.DataFrame([nlp(txt).vector for txt in corpus
                        if nlp(txt).vector.mean() != 0
                        ])
    vector_similarity = cosine_similarity(w2v,w2v)
    np.fill_diagonal(vector_similarity, np.nan)

    lenght = len(vector_similarity)
    num_of_similar=0
    total=0.0001 + lenght*(lenght-1)/2

    for i in range(lenght):
        for j in range(i+1,lenght):
            if i!=j:
                if vector_similarity[i,j]>0.95:
                    num_of_similar+=1

    return num_of_similar/total*100
```

Weighted sentiment function

```
import numpy as np
from textblob import TextBlob

from sklearn.feature_extraction.text import TfidfVectorizer


def weighted_sentiment(sentences):

    vect = TfidfVectorizer()
    tfidf = vect.fit_transform(sentences)
    arr = tfidf.toarray()

    column_name = vect.get_feature_names()
    column_sentiment = []

    for word in column_name:
        analysis = TextBlob(word)
        sent = abs(analysis.sentiment.polarity)
        column_sentiment.append(sent)

    column_sentiment = np.array(column_sentiment)
    sentiment_tfidf_weight = column_sentiment*(1+arr)

    weighted_text_sentiment = np.sum(sentiment_tfidf_weight)

    return weighted_text_sentiment
```