FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Automated Fake News detection using Computational Forensic Linguistics

Ricardo Moura

Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Doctor Rui Manuel Sousa Silva

Second Supervisor: Doctor Henrique Daniel de Avelar Lopes Cardoso

July 19, 2021

Automated Fake News detection using Computational Forensic Linguistics

Ricardo Moura

Mestrado Integrado em Engenharia Informática e Computação

Approved in oral examination by the committee:

Chair: Doctor Sérgio Sobral Nunes
External Examiner: Doctor Paula Cristina Quaresma da Fonseca Carvalho
Supervisor: Doctor Rui Manuel Sousa Silva
Second Supervisor: Doctor Henrique Daniel de Avelar Lopes Cardoso

July 19, 2021

Abstract

In our society, almost everyone has access to the internet and can post anything about any topic at any time. Despite its many advantages, this possibility brought along a serious problem: fake news. Fake news can emerge very easily and rapidly spread mis- and disinformation across the world. As a result, the phenomenon has gained particular relevance over the past few decades. There is no consensual and objective vision of what fake news is, but in the context of this work, fake news is news that does not follow the journalism principles of factuality, objectivity, and neutrality. Instead, fake news try to mimic the look and feel of real news with the intent to disinform the reader.

What makes fake news a real problem is its potential negative impact on our society. Lay people are attracted to this kind of news because of their appealing titles and often give more attention to them than to truthful accounts. Although fake news have always existed, their use as a form of manipulation and control has recently gained attention, due to their fast and immediate propagation, mainly through social media, without any kind of curation or filtering. Their influence is noticeable in recent events, such as the previous election of Trump (2016) in the USA or Bolsonaro in Brazil (2018), where, according to some sources, fake news was a key determinant of the outcome of the election.

Despite the development of advanced computer systems to detect fake news, most are based on fact-checking methods. These methods are useful when facts are manipulated, but not so much when the truth in the news is distorted, exaggerated, or even placed out of context. In this work we address the fake news phenomenon by using an approach based on forensic linguistic analysis. Contrary to previous works on fake news detection, our approach builds upon methods of linguistic and stylistic analysis that have been explored in forensic linguistics. These include, but are not limited to: text statistics; spelling; *n*-grams; lexical choices; etc. A model built upon these features, which have been previously tested, e.g., to attribute authorship or detect bias in text, have a significant potential to detect fake news.

The best results reported are very promising and comparable to the state-of-the-art NLP models (e.g., BERT and GPT-2), achieving an accuracy of 97% and a macro average f1-score of 91%. However, contrary to the state-of-the-art NLP models (most based on Deep Learning), our approach has the potential to reveal several insights into how the models discriminate fake and genuine news. Although comparing the results to those in the literature review is a non-trivial task due to different datasets used, we hope that this work will spark new investigation and operate as a baseline to future Portuguese fake news detection work.

Keywords: Fake News, Forensic Linguistics, Natural Language Processing, Text Classification, Disinformation, Misinformation.

Resumo

Na nossa sociedade, o acesso à internet é generalizado e praticamente todos os utilizadores podem publicar sobre qualquer tópico a qualquer momento. Apesar das suas muitas vantagens, esta possibilidade trouxe consigo um problema sério: notícias falsas. As notícias falsas podem surgir com muita facilidade e difundir rapidamente desinformação por todo o mundo. O fenómeno ganhou particular relevância ao longo das últimas décadas. Não existe uma perspetiva consensual e objetiva do que são notícias falsas; porém, no contexto deste trabalho, considera-se notícias falsas as notícias que não seguem os princípios fundamentais do jornalismo: factualidade, objetividade e neutralidade. Em vez disso, as notícias falsas tentam imitar o aspeto de notícias reais com a intenção de desinformar o leitor.

O que torna as notícias falsas um problema é que elas podem influenciar a nossa sociedade. Este tipo de notícias atrai o público em geral graças aos seus títulos apelativos e muitas vezes os leitores prestam mais atenção a este tipo de notícias do que àquelas que não são falsas. Embora as notícias falsas sempre tenham existido, a sua utilização como forma de manipulação e controlo ganhou atenção recentemente, devido à sua rápida e imediata propagação, principalmente através das redes sociais, sem qualquer tipo de filtragem. A influência deste fenómeno é percetível em eventos recentes, tais como a eleição de Trump (2016) nos EUA ou Bolsonaro no Brasil (2018), onde, de acordo com algumas fontes, as notícias falsas foram um fator determinante para o resultado das eleições.

Apesar do desenvolvimento de sistemas informáticos avançados para detetar notícias falsas, a maioria baseia-se em métodos de verificação de factos. Estes métodos são úteis quando os factos são manipulados, mas não tanto quando a verdade nas notícias é distorcida, exagerada, ou mesmo descontextualizada. O nosso objetivo é abordar o fenómeno das notícias falsas utilizando uma técnica baseada na análise de linguística forense. Ao contrário das abordagens à deteção de notícias falsas existentes, a nossa abordagem baseia-se em métodos de análise linguística e estilística que foram testados e comprovados na linguística forense. Estes incluem, mas não exclusivamente: estatísticas de texto; ortografia; *n*-gramas; escolhas lexicais; etc. Um modelo desenvolvido a partir destas características, que foram previamente testadas, por exemplo, para atribuir autoria ou detetar posicionamentos de parcialidade em texto, possuem um potencial considerável para detetar notícias falsas.

Os melhores resultados registados são muito promissores e comparáveis aos modelos de NLP mais avançados (por exemplo, BERT e GPT-2), alcançando uma precisão de 97% e uma macro average f1-score de 91%. No entanto, ao contrário dos modelos mais recentes de NLP (a maioria baseada em Deep Learning), a nossa abordagem tem potencial para revelar vários pormenores sobre a forma como os modelos distinguem notícias falsas e genuínas. Embora comparar os resultados com trabalhos publicados na mesma área seja uma tarefa não trivial devido à utilização de diferentes conjuntos de dados, esperamos que este trabalho desperte novas investigações e funcione como padrão de comparação para futuros trabalhos de deteção de notícias falsas em

português.

Keywords: Fake News, Notícias Falsas, Linguística Forense, Processamento de Linguagem Natural, Desinformação

Acknowledgements

Throughout the writing of this dissertation, I have received a great deal of support and assistance. I wish to thank various people for their contribution to this project. Without them, doing this dissertation would be far more challenging.

I would first like to thank my supervisors – Professor Rui Sousa Silva and Professor Henrique Lopes Cardoso – for their constant, valuable, and constructive guidance and availability throughout the planning and development of this research work. Most of all, I am thankful for your faith in the project and in me.

I would also like to thank my closest friends – Ana, Fábio, Miguel, Teixeira, and Xavi – for believing in me even when I didn't, for not even one time letting me think that writing this thesis was impossible, for giving the needed distractions to rest my (big) brain outside of my research, and last but not least for the ones that walked the extra mile and were (almost) a third research supervisor to this thesis.

Finally, I want to thank my family for all the continuous love, and support. My siblings who did nothing but were simply there, and my parents for waking me up every day, for giving me the opportunities and experiences that have made me who I am, for always showing how proud they are of me and for encouraging me in every single way not just during the development of this dissertation but throughout my life.

Thank you,
Ricardo Moura

"I'm a great believer in luck,
and I find the harder I work, the more I have of it."

Thomas Jefferson

"Beware of false knowledge,
it is more dangerous than ignorance."

George Bernard Shaw

Contents

List of Figures

List of Tables

Abbreviations

AI	Artificial intelligence
AUC	Area Under the ROC curve
BERT	Bidirectional Encoder Representations from Transformers
BoW	Bag of Words
DT	Decision Tree
DL	Deep Learning
FN	False Negatives
FP	False Positives
LR	Logistic regression
LRL	Low Resourced Languages
LSVM	Linear Support Vector Machine
ML	Machine Learning
NB	Naive Bayes
NER	Named-entity recognition
NLP	Natural Language Processing
POS	Part of Speech
RF	Radom Forest
ROC	Receiver Operating Characteristic
SGD	Stochastic Gradient Descent
SVM	Support Vector Machine
TF-IDF	Term Frequency-Inverse Document Frequency
TN	True Negatives
TP	True Positives

Chapter 1

Introduction

1.1 Context

Technology has improved a lot in recent years, and its development and adoption have become increasingly fast and easy. For instance, the Internet as we know it only came into existence in 1990, when computer scientist Tim Berners-Lee created the fundamental technologies that remain the foundation of today's world wide web [19]. This invention completely changed the world, and with it, a new set of opportunities for new technologies emerged. One of the technologies that came to define and influence the next generations was new computer-mediated communication channels, such as social media, messaging services, blogs, and so on. These channels made it possible for anyone to communicate and share anything about any topic at any time, instantly and effortlessly.

Nowadays, people are more connected to social media than ever. Companies are aware of this phenomenon and attempt to use it for their advantage. In addition to posting the news on their website, news companies now share the news on social media. In fact, studies report that people are shifting away from traditional news sources to social media and messaging services to find news of their interest [59]. Even though these platforms have many advantages, they bring along a serious problem: fake news. Because those platforms give all users the freedom to share anything they want at any time, fake news can emerge very easily and rapidly spread mis- and disinformation. Misinformation concerns false or misleading information that is inadvertently created or spread without the intention of deceiving the reader, while with disinfomation false information is deliberately spread to deceive the reader.

The fake news phenomenon can be defined in several different ways and acquire multiple forms, from satire to fabrication [42]. Some of them, such as satire, are even socially acceptable. The definition of fake news has mutated throughout the years and began to be applied in the wrong circumstances [55]. Therefore, deciding on a precise and exact definition for fake news is an important effort that will certainly help build a solid foundation for this project. In the context of this dissertation, fake news is news that does not follow the journalism principles of factuality, objectivity, and neutrality [6, 25]. Instead, fake news try to mimic the look and feel of

real news [56] with the intent to disinform the reader. That is, the information provided is not necessarily factually untrue, but it is presented in a way that aims to desinform the reader.

1.2 Motivation

Any information posted on the web can reach millions of people within minutes or even less. Even though this speed and accessibility of content delivery have numerous advantages, they also present some challenges. They give the reader more responsibility to decide whether the content is trustworthy or not, making the readers more vulnerable to unreliable information. This decision becomes increasingly difficult to make with the growing number of unreliable sources of information to which people are exposed [3]. As a consequence, what makes fake news a real problem is how this information can influence our society.

Although untruthful news accounts have always existed, their use as a way of manipulation and control has recently gained more attention, due to their fast and immediate propagation through social media, without any kind of curation or filtering. Moreover, lay people are attracted to this kind of news because of their alluring headlines (which lead to *clickbait*) and often give more attention to them than to truthful accounts [9].

The influence of this phenomenon has been particularly noticeable in recent events, especially in a political context, such as the previous election of Trump (2016) in the USA or Bolsonaro in Brazil (2018). In both cases, according to some sources [4, 7, 44], the fake news phenomenon was a key determinant of the election outcome.

Another example of this influence is the 2020 USA presidential elections. BBC News wrote an article [57] regarding how the now former president Donald Trump and his team challenged the result of the election by spreading fake news. One of the many fake news pieces mentioned in this article was a claim shared by Trump (Figure 1.1) and seen by a large number of people, which implied that in some states, more votes than registered voters were recorded, an outcome that is known as "overvote". This idea started a rumor and was supported by a document (Figure 1.2), shared by a former Republican congress candidate, that supposedly listed some precincts of Michigan where overvote had happened. This document was proven to be completely false by BBC News. Not only does it show the wrong State (these precincts are all in the State of Minnesota), but also the information is wrong (in reality, overvote did not happen on any of the Minnesotan precincts listed).

A case reported by Polígrafo[1] also reveals the influence caused by fake news on our society where a piece of fake news had a significant impact [40]. A snippet of a news post (Figure 1.3) was shared on WhatsApp about Cristiano Ronaldo's hotels, that was later proven false. The news' snippet suggested that Cristiano Ronaldo would transform all the hotels he owns into hospitals to receive and treat patients infected with Covid-19, completely free of charge. The post was shared so often and had such a huge impact that it made the headlines to the well-known Spanish sports

[1] Polígrafo is an online journalistic project that aims to analyze and expose the veracity of the most recent and viral news articles. It is considered the first fact-checking newspaper developed in Portugal.

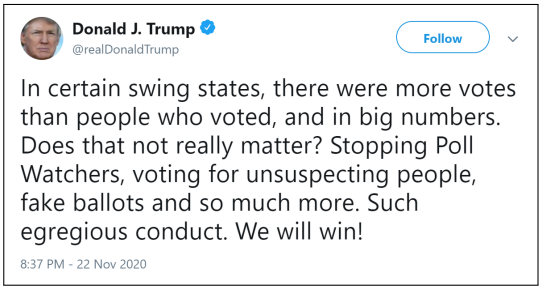
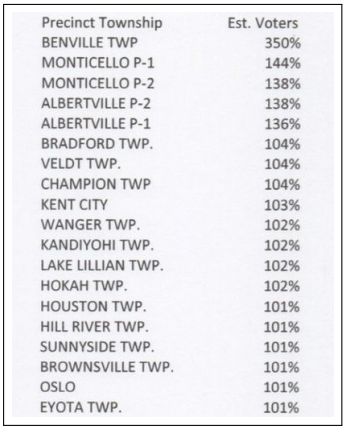


Figure 1.1: Trump tweeted about how in some states more votes than registered voters were recorded.
Retrieved from Trump's Twitter account.



Precinct Township	Est. Voters
BENVILLE TWP	350%
MONTICELLO P-1	144%
MONTICELLO P-2	138%
ALBERTVILLE P-2	138%
ALBERTVILLE P-1	136%
BRADFORD TWP.	104%
VELDT TWP.	104%
CHAMPION TWP	104%
KENT CITY	103%
WANGER TWP.	102%
KANDIYOHI TWP.	102%
LAKE LILLIAN TWP.	102%
HOKAH TWP.	102%
HOUSTON TWP.	101%
HILL RIVER TWP.	101%
SUNNYSIDE TWP.	101%
BROWNSVILLE TWP.	101%
OSLO	101%
EYOTA TWP.	101%

Figure 1.2: Document that supported the claims done by Trump.
Retrieved from BBC News article [57].

newspaper *Marca* (Figure 1.4).

Figure 1.3: Image shared by many in WhatsApp that appears to be a piece of real news.
Retrieved from [40].

Figure 1.4: Sports newspaper *Marca*, a well-known newspaper, posted about the situation.
Retrieved from Polígrafo.

These are two good examples that show the ability of fake news to spread and influence society.

1.3 Objectives

Verifying the veracity of online information is a difficult but critical challenge. Currently, there are two widely used methods for detecting fake news: a manual alternative with human intervention, or automatically with Machine Learning methods [64].

In the first alternative, the responsibility relies entirely on humans to assess the news' veracity and accuracy and then flag it depending on their judgment. However, the findings from a study about deception judgment [8] show that (non-experts) humans are not skilled enough to detect

lies in text. Based on more than 200 experiments, the study indicates that humans are just 4% better than chance. Moreover, manually checking for fake news is not feasible due to its limited scalability.

The second alternative concerns the use of sophisticated computer systems to detect fake news. However, most existing systems are based on fact-checking methods, which fall short of the desired – and required – effectiveness. Normally, those systems are affiliated with a larger media outlet or are entrepreneurial [53], and we want these systems ideally to be as least biased as possible. Furthermore, despite several attempts, these systems still lack the robustness to perform a reliable verification of which information is false [64]. Additionally, detecting fake news goes beyond identifying false information. Fact-checking methods are useful when facts are manipulated, but less so when the truth in the news is distorted, exaggerated, or even placed out of context.

This project presents a system that, contrary to fact-checking, does not depend on the veracity of the facts. Instead, we look at how the author communicates and how the news is written. In light of this, we address the fake news phenomenon using an approach based on forensic linguistic analysis. Our approach builds upon linguistic and stylistic analysis methods tried and tested in forensic contexts. These include, but are not limited to: text statistics (e.g., average text, paragraph, sentence and word length, and n-gram sequences); spelling; and lexical choices (e.g., Part-of-Speech tags used). We claim that the approach described, which has been previously used, for instance, to attribute authorship or detect bias in texts [54], has a significant potential to detect fake news.

1.4 Research Questions

In this research, we formulate the following research questions that serve as a guideline for the development of this project:

- Can an approach based on forensic linguistic analysis yield good results at detecting fake news?
- Which are the most relevant features to detect fake news in a forensic linguistics-based system?
- How do systems based on forensic linguistic analysis compare to a modern Deep Learning approach?

The first question is the most important to answer, as it establishes the success of the project. It focuses on the potential of a forensic linguistics approach to achieve our goal of identifying fake news. The second question is related to the findings and conclusions that we can draw about what features a text should exhibit to be labeled as fake news. The last question concerns comparing our approach to state-of-the-art NLP tools such as BERT or GPT-2. We aim to compare factors such as performance metrics and feature engineering efforts, among others.

1.5 Document Structure

The dissertation proposal is organized as follows:

This initial chapter (Chapter 1) introduces the project's context, presenting the impact that fake news has on our society while also exposing its issues. Finally it presents the main objective of this dissertation and the research questions that will serve as a guideline for the deployment of this project.

In the second chapter (Chapter 2) we will present the Literature Review. This chapter will be divided into three sections: the linguistic approach, the fact-checking approach, and the hybrid approach. In each section, we will address how each approach detects fake news and review previous papers that can be helpful to the work in this dissertation. For each paper, we will review their main objectives, outlining how the author did it by describing what features and models were tested and used. We will also report what the results were and what conclusions were drawn.

The third chapter (Chapter 3) presents the background of Natural Language Processing relevant for this work, so as to have the reader acquainted with a common ground of knowledge necessary to understand the main contribution of this thesis. We will detail what steps are usually involved in an NLP pipeline, including Preprocessing, Feature Extraction, and Data Representation. We will also describe the most commonly used algorithms and performance metrics in NLP classification problems. Moreover, we will clarify why working with low-resourced languages (in this case, Portuguese) can affect the outcomes.

The fourth chapter (Chapter 4) will describe the methodology used in this project. We will explain in more detail all of the resources used, from the collected corpus and the dataset generated to the choices related to NLP tools. Additionally, we will describe and discuss some characteristics of the dataset.

The fifth chapter (Chapter 5) shares, evaluates, and discusses the results obtained in this project for all the approaches proposed. Furthermore, we do feature analysis, exploring the most defining characteristics in a text that lead the models created into considering them fake.

The sixth and last chapter (Chapter 6) will draw some conclusions and give a perspective into the stage of the project. Finally, we will address possible threats to the project's validity, give an overview of the results obtained, and discuss possible improvements or experiences for future work.

Chapter 2

Literature Review

In this chapter we will present the Literature Review on fake news detection systems. This chapter is organized into three sections: the linguistics-based approaches, the fact-checking approach, and the hybrid approach. In each section, we will address how each approach detects fake news and review previous papers potentially relevant to this dissertation. For each work, we will review its main objectives, outlining the approach by describing what features and models were tested and used. We will also report what the results were and what conclusions were drawn.

2.1 Linguistics-based Approaches

This approach is the one that is most aligned with the methods of this dissertation. This approach builds upon the assumption that when someone writes a lie, they strategically use their language to avoid being caught. However, it is not easy to control language, and most of the times, not all traces and patterns can be hidden. A study from 2007 [24] investigated changes in both the liar's and the conversational partner's linguistic style. The study analyzed more than 200 transcripts and confirmed the existence of these types of patterns. It revealed that liars produced more words, more sense-based words, avoided causal terms, and used more other-oriented pronouns, among other findings. So the goal in this approach is to detect when someone is trying to spread a lie by finding these mistakes or patterns in the text. Unlike fact-checking, this approach is somewhat understudied. Nevertheless, some examples are the as follows.

Ahmed et al. (2017) [1] propose an approach to detect fake news using *n*-gram analysis. The authors begin by analyzing two different feature extraction techniques and six different machine classification methods to get the best results. The paper compares Term Frequency (TF) and Term Frequency-Inverted Document Frequency (TF-IDF) for feature extraction. To establish the best classification method, the authors compared the following classification methods: Support

Vector Machine (SVM), Linear Support Vector Machine (LSVM), K-Nearest Neighbors (k-NN), Decision Tree (DT) and Logistic Regression (LR).

They concluded that linear-based classifiers (e.g., LSVM and LR) perform better than the non-linear ones. They tested the impact that the n-grams had on the performance, ranging from $n = 1$ to $n = 4$ and alternating the maximum number of n-grams collected between 1000, 5000, 10000, and 50000. The best results were achieved when using a set of 50000 uni-grams ($n = 1$). Moreover, they reached the best performance using Term Frequency-Inverse Document Frequency (TF-IDF) as a feature extraction technique and Linear Support Vector Machine (LSVM) as a classifier, with an accuracy of 92%. The accuracy yielded in this work was better than the results obtained in Horne and Adali (2017) [27] where the dataset was initially collected. Although this study achieved high accuracy, this can owe to a Population Bias or Representation Bias [36] since the authors direct their attention to n-gram analysis. As we see in Cruz et al. (2019) [15], reliance only on n-grams could present a problem because this feature extraction technique may vary depending on media attention throughout the years.

Perez et al. (2017) [47] can be divided into 3 contributions. Firstly, the paper introduces and describes the collection, annotation, and validation process of two novel datasets for the task of fake news detection: one collected via crowd-sourcing covering six news domains, and the other scraped from the web and covers celebrity fake news. Secondly, using the datasets mentioned above, the authors made a set of experiments and exploratory analyses to identify linguistic properties predominantly present in fake content. The classification method used was a linear support vector machine (LSVM) algorithm with five-fold cross-validation. In contrast with the first paper [1], which focuses more on discovering the best combination of feature extraction and machine classification technique and less on the features itself (using n-gram derived features), this work focuses on finding the best combination of features. In order to do so, the authors also conducted several experiments with different combinations of features, including:

- ***N*-grams:** unigrams and bigrams were extracted. These features are encoded as TF-IDF values;
- **Punctuation Characters:** this includes the count of periods, commas, dashes, and question and exclamation marks;
- **Psycholinguistic Features:** this includes features such as: summary categories (e.g., analytical thinking, emotional tone), linguistic processes (e.g., function words, pronouns), and psychological processes (e.g., affective processes, social processes);
- **Readability:** features that indicate text comprehensibility, based on elements such as the number of characters, complex words, long words, number of syllables, word types, and paragraphs, among other content features;
- **Syntax:** a set of features encoded as TF-IDF values derived production rules based on context-free grammar (CFG) trees.

Relying on this information, they built a fake news detector that achieved the best performance when all features were used with a 78% accuracy. To conclude, the paper shares some insights and reflections into the performed experiments. The results suggest important differences in fake news as compared to legitimate news contents. Some of these differences are that fake news contents use more social and positive words, express more certainty, focus on the present and future actions, and exhibit more adverbs, verbs, and punctuation characters.

Cruz et al. (2019) [15] propose the creation of a linguistically-guided model for hyperpartisan news detection from a minimal set of interpretable features. This system intends to test the use of linguistic features, which were successfully used in authorship attribution, to detect hyperpartisan news. Although, in this paper, the object of study is hyperpartisan news[1], understanding the methods applied is very important since they are very similar to what this dissertation aims to use. Furthermore, the authors argue that there is a large intersection between hyperpartisan and fake news. The features on which the model was trained were:

- Number of sentences in the document;
- Average sentence length in words;
- Average sentence length in characters;
- The variance of sentence length in characters;
- Average word length in characters;
- The variance of word length in characters;
- Relative frequency of punctuation characters;
- Relative frequency of capital letters;
- The measure of vocabulary diversity and richness (ratio of unique tokens);
- Frequency of the k most frequent word *n*-grams;

The authors experimented with the following classifiers: support vector machines with linear kernels (SVM or LSVM), random forests (RF), and gradient boosted trees (GBT). The best performance achieved was with a random forest model supplied with linguistically-inspired features in addition to the 50 most frequent *n*-grams that produced an accuracy of 71.7%. The paper highlights some properties that emerge from analyzing feature importance, such as:

- The number of sentences and the frequency of capital letters are the most important features;
- Reliance on *n*-grams could present a problem, as these may refer to entities with a high variance of media attention;
- Hyperpartisan news have a higher number of sentences, but each with shorter length than mainstream articles, and with decreased vocabulary diversity;

[1] hyperpartisan news are news that exhibit an extreme bias towards a single side.

In this paper, another approach for the same dataset was attempted: Ensemble modeling[2]. Using a voting classifier that grouped models with the same objective as the one described in this paper, the model achieved an accuracy of 88.5%. The paper finishes by suggesting that using ensembles of individually distinct classifiers could be a promising option to explore more accurate detection system.

Horne and Adali (2017) [27], unlike the other works that focus on the main text, considers only the news headlines for detecting fake news. The authors believe that fake news is targeted at audiences who are not likely to read beyond titles. The paper uses three different datasets (two of which are novel) and draw some conclusions about them, such as:

- Fake news articles tend to be shorter in terms of content but use repetitive language and less punctuation;
- Fake titles are longer, use few stop words, and fewer nouns but more proper nouns;
- Fake news packs the main claim of the article into its title, which often is about a specific person and entity;
- The article (main text) tends to be short, repetitive, and less informative;

They extracted different features and arranged them into three categories, as follows:

- **Stylistic Features:** features that represent the readability of the text, such as:
  - PoS tagger count on each tag;
  - The number of stop-words;
  - The number of punctuation characters;
  - The number of quotes;
  - The number of negations (no, never, not);
  - The number of informal/swear words;
  - The number of interrogatives (how, when, what, why);
  - The number of all capital letters words;
- **Complexity Features:** the authors looked at two levels of intricacy:
  - Sentence level: the number of words per sentence and each sentence's syntax tree depth, noun phrase syntax tree depth, and verb phrase syntax tree depth;
  - Word level: the readability of each document;
- **Psychological Features:** features based on well studied word counts that are correlated with different psychological processes, and basic sentiment analysis.

[2]Ensemble modeling is a process of grouping multiple models to predict an outcome

With these features, the authors built a linear support vector Machine (LSVM) classifier with 5-fold cross-validation, achieving 71% accuracy when predicting fake news against genuine news.

To conclude, we can summarize all this information into two tables. Table 2.1 presents the features used in each paper.

Feature Type	Feature	Ahmed et al. (2017) [1]	Perez et al. (2017) [47]	Cruz et al. (2019) [15]	Horne and Adali (2017) [27]
Count	Number of all capital letters words				✓
	Number of interrogatives words (how, when, ...)				✓
	Number of informal/swear words				✓
	Number of POS tag		✓		✓
	Number of stop-words				✓
	Number of punctuation		✓		✓
	Number of quotes				✓
	Number of negations words (no, never, not)				
	Number of emotion words		✓		✓
	Number of function words		✓		
	Number of sentences			✓	✓
Average	Average word-length of sentences			✓	
	Average character-length of sentences			✓	
	Average character-length of words			✓	
Other	Readability of the text		✓		✓
	N-grams encoded as TF-IDF	✓	✓		
	Sentiment Analysis				✓
		92%	78%	72%	71%

Table 2.1: Comparison Table: Features used in each paper. The bottom row presents the best performing model's accuracy.

As can be seen in Table 2.1, the average accuracy of the works presented in this section is around 78%. However, it is actually very complex to compare the results, as we will mention later, in Section 2.4. This table also shows that the accuracy is always higher when the *n*-grams are used as features.

Table 2.2 shows which models were tested and which one produced the best results.

System	SVM	LSVM	KNN	GBT	DT	SGD	LR	RF	Accuracy
Ahmed et al. 2017 [1]	✓	★	✓		✓	✓	✓		92%
Perez et al. 2017 [47]		★							78%
Cruz et al. 2019 [15]	✓	✓		✓				★	72%
Horne et al. 2017 [27]		★							71%

Table 2.2: Comparison Table: Best classifier model for each paper with a linguistic approach. The column accuracy presents the best performing model's accuracy. A ✓ indicates which models were tested, and ★ represents which model gave the best results.

Regarding the table discerning the classifiers used by these related works 2.2, we could see that although they test many algorithms, the best in almost all cases is the LinearSVM. However, one of the works – Cruz et al. 2019 [15] – is the exception, presenting the Random Forest as its best performing classifier.

The linguistic approach is definitely the closest to what we are trying to achieve in this dissertation. However, there is another approach that can be used for the same purpose of detecting fake news: fact-checking.

2.2 Fact-Checking Approach

The fact-checking approach focuses on validating the veracity of facts present in a given text. Although this approach has its weaknesses, as mentioned in the introduction, it can result in good outcomes. This approach can be divided into two big stages: fact extraction and fact-checking [63].

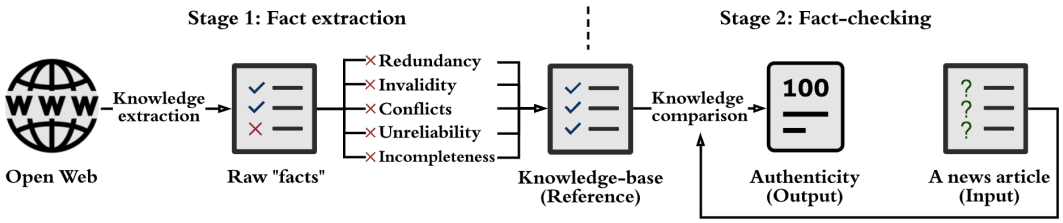


Figure 2.1: Fact-checking approach full process. Retrieved from [63].

The first stage is where the fact database, more commonly known as Knowledge Base, is collected and built, assuring that issues such as redundancy, incompleteness, conflicts, among others, do not happen. It can be done in two methods: single-source or open-source. The single-source method relies only on the information of one source to extract the knowledge. The open-source method attempts to merge the knowledge from different sources. Although the open-source method can be less efficient, it leads to more complete knowledge than the single-source method. This step is optional since we can always use an already existing Knowledge Base.

The second element, the fact-checking itself, has two steps: extract the claims (information) from a given text and then compare them against known facts, i.e., the Knowledge Base. To extract

claims, we use Information Extraction techniques that transforms the relations present in a given text into triples of information: Subject-Predicate-Object.

However, we consider that this approach has three issues. Firstly, the systems that use this approach typically are affiliated with a larger media outlet or are entrepreneurial [53], and we want these systems ideally to be as least biased as possible. Secondly, these systems still lack the robustness to perform a reliable verification of which information is false [64]. These systems show vulnerabilities in the Knowledge Base maintenance, which needs to be up to date with the most recent facts. Fake news is associated with newly emerging, time-critical events, which means there is a time-window where the Knowledge Base may not have the corroborating evidence to verify a claim appropriately [52]. The third issue is related to the fact that there are more types of fake news besides the ones with false facts. Although fact-checking methods are useful when facts are manipulated, they struggle when the truth in the news is distorted, exaggerated, or even placed out of context. Sometimes fake news may use valid and proven claims, but within an incorrect context [17].

Furthermore, this approach is slightly out of the scope of this dissertation. On the one hand, contrary to the first approach, this one does not rely entirely on Data Mining. A typical Data Mining problem uses algorithms to learn and extract patterns from a dataset. Instead, this approach normally uses rule-based algorithms to check if a claim is true by comparing it to a Knowledge Base. On the other hand, although it can be used as a fake news detecting system, detecting fake news is not the main goal of fact-checking. The main goal, as already stated, is to validate the veracity of facts.

2.3 Hybrid Approach

Implementing a fake news detection system can also take a hybrid approach and take advantage of both linguistic and fact-checking approaches. Although many works try to use this hybrid approach, many do not focus on fake news detection. Applications of this approach can be seen in other works:

In Popat et al. (2017) [48], the authors develop a model that could retrieve various articles about the claim implicit in a news article, with the primary goal being Rumor and Hoax Detection. The model created obtained the best results when modeling the mutual interaction between the stance (i.e., support or refute) of the sources, the articles' language style, the reliability of the sources, and the claim's temporal footprint on the web. For this dissertation, it is essential to understand what the authors used for characterizing the language style:

- Assertive and factive verbs (e.g., "claim", "indicate") capture the degree of certainty to which a proposition holds;
- Hedges are the mitigating words (e.g., "may") that soften the degree of commitment to a proposition;
- Implicative words (e.g., "preclude") trigger presupposition in an utterance;

- Report verbs (e.g., “deny”) emphasize the attitude towards the source of information;
- Modal verbs (e.g., “could”, “maybe”) capture the degree of confidence, perspective, and certainty in the statements;
- Lastly, a lexicon of subjectivity and bias captures the writer’s attitude and emotions while writing an article.

2.4 Summary

In this chapter, we reviewed the most common approaches to fake news detection. From our perspective, what we found was an understudied domain. These approaches are used in fact, but mostly in other contexts, where fake news detection is just a small percentage. Other works with different goals, such as rumor detection [2], satire detection [61], hyperpartisanship detection [15], or deception detection [33] are other examples of the use of these approaches. Such lack of research into fake news detection using approaches other than fact-checking is also evident in Portuguese.

Furthermore, we found that comparing the performance between the studied works was non-trivial due to the fact that the authors study the performance of the proposed models under different datasets, i.e., the authors did not assess the performance of the models with relation to a standard benchmark dataset.

Our methodology (described in more detail in Chapter 4) comes closer to linguistics-based approaches. However, contrary to other works, our approach builds upon analysis methods tried and tested in forensic linguistics, thus resembling, to some extent, the work done in Cruz et al. (2019) [15], but innovating from it.

Chapter 3 presents some NLP basic concepts that will help the reader keep abreast of the processes used in this project’s development.

Chapter 3

Natural Language Processing

This chapter presents the background of Natural Language Processing relevant for this work, to get the reader acquainted with a common ground of knowledge necessary to understand this thesis' contribution. We will detail what steps are usually involved in an NLP pipeline, including Preprocessing, Feature Extraction, and Data Representation. We will also describe the most commonly used algorithms and performance metrics in NLP classification problems. Moreover, we will explain why working with low-resourced languages (in this case, Portuguese) can affect the outcomes.

3.1 Introduction

Natural Language Processing, also known as NLP, is a branch of Artificial Intelligence (AI) that brings together a range of areas, from computer science to linguistics. The main objective is to help computers understand, interpret, generate, but fundamentally operate on natural languages. We can see Natural Language Processing in multiple applications [28] such as Text Classification, Sentiment Analysis, Chatbots, Virtual Assistants, Machine Translation, Text Summarization, among many others.

3.2 Preprocessing

In Natural Language Processing, the input is usually unstructured text with a variable amount of noise, and thus a big effort on data processing and cleaning is required. Therefore we usually start by building a pipeline. A pipeline in Machine Learning aims to streamline processes: break up the problem into small modular pieces and then solve each piece separately. These steps aim not only to clean and normalize the text but also to create new attributes, thus transforming unstructured information into features. The first step is preprocessing, which consists of normalizing and

cleaning the input text. There is no single pipeline for NLP, but a set of steps are normally applied to the input text before supplying it to the model.

3.2.1 Tokenization

Normally the NLP pipeline starts with tokenization. It is an essential step in both traditional NLP methods and DL-based architectures. There are multiple types of tokenization, but in this step, the goal is to break a given raw text into pieces called tokens. These tokens can be words, multi-word expressions, or punctuation, depending on the tokenization method used. The main purpose of tokenization is to help the model understand the context and the meaning of the text by analyzing the words' sequence.

In addition to breaking a given text into tokens, it can also be used to generate some features such as the count of question or exclamation marks, which is important to later train the model. An example of tokenization is presented in Figure 3.1.

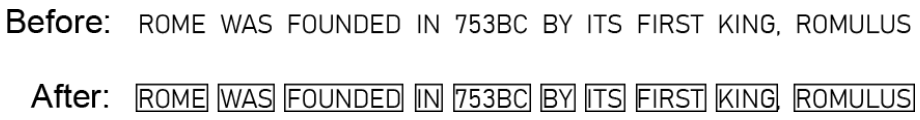


Figure 3.1: Example of tokenization.

3.2.2 Stop Word Removal

Stop words are words that do not contribute to the value or meaning of a sentence. Usually, these are the most common words in a language, i.e. grammatical words (as compared to lexical items). For example, "*for*", "*to*", "*the*", etc, in English, and "*a*", "*por*", "*um*", "*para*", etc, in Portuguese.

Depending on the ML task, it is crucial to remove this type of words because doing so provides many advantages. First of all, by removing words, the dataset size decreases and, consequently, the time to train the model. Not only that, but also much of the text's noise is removed in this step, only leaving the meaningful tokens, which reduce the number of features used. An example of stop word removal is presented in Figure 3.2.

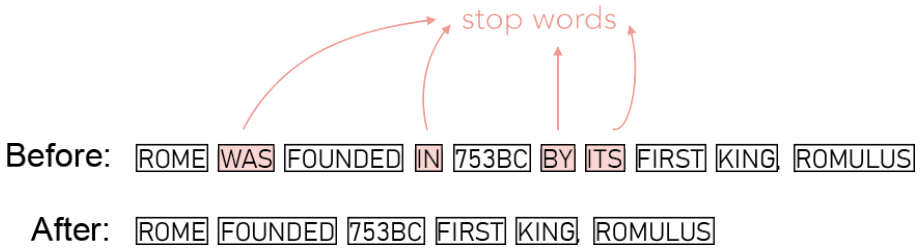


Figure 3.2: Example of Stop Words Removal.

3.2.3 Stemming and Lemmatization

In grammar, it is common for a word to have different forms but roughly the same meaning. We see this happening, for example, when conjugating verbs or when inflecting words in number as in using plural forms, etc. We have to make the model perceive a words with the same meaning as the equivalent. To achieve this, the words need to be normalized, and we can do this by using one of the two possible methods: stemming or lemmatization. The two methods have the same goal: reducing the words to their common root. However, they differ in how they operate and consequently so does their output. An example of stemming is presented in Figure 3.3.

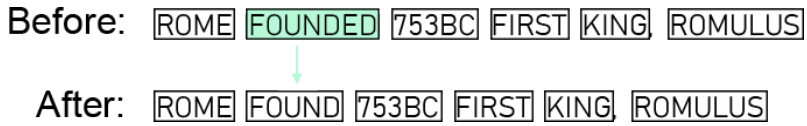


Figure 3.3: Example of stemming.

Stemming is the process of removing affixes of a word to its stem (root). Although simplistic and with some limitations, this method works fairly well in most cases.

Lemmatization has a more complex heuristic than stemming. This process considers the full vocabulary to apply a morphological analysis to words and return a lemma (root). A lemma is the root of all derived forms of a word, in contrast to a stem, which is just a part of a word. The Figure 3.4 shows an example of the difference between these two methods for the word “change”.

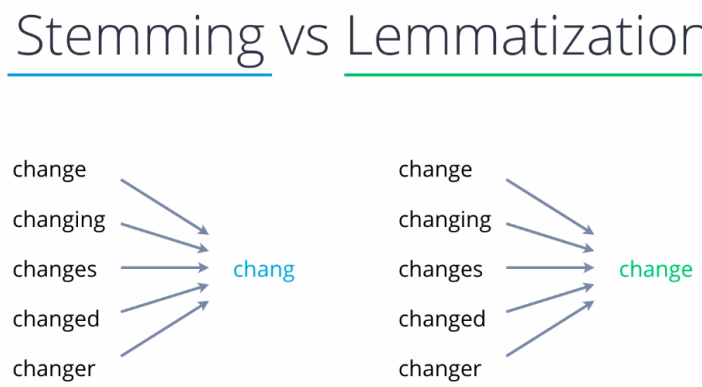


Figure 3.4: Difference between stemming and lemmatization. Adapted from [51].

3.2.4 Part-of-speech Tagging

A Part-of-Speech tagger, commonly called PoS tagger, is another important step in a common NLP pipeline. Its main goal is to identify and tag the lexical role of each token based on its definition

and context in a given text (As stated earlier, a token can be a word, an expression, a punctuation mark, etc.).

However, PoS tagging is neither an easy nor a generic task of mapping words to their part of speech. As stated, Natural Language can be highly ambiguous, which means that the role of a word in a sentence can only be established if the context of the sentence is known sentence's context because of how they are interrelated. For instance, homonyms can present a big challenge to training the model if not correctly dealt with. An example of Natural Language ambiguity can be observed in the following sentences:

(A) *The train* ***leaves*** *at seven.*
(B) *The* ***leaves*** *fell in the autumn.*

In sentence **A**, the word ***leaves*** acts as a verb, as opposed to sentence **B** where the word ***leaves*** is a plural form of the noun *leaf*. Although their form is identical, their meaning is strikingly different. This step of tagging the words with the respective PoS role in a given text is crucial. By performing this action, we clear most of the ambiguity, which will definitely help improve the model's accuracy. An example of stop word removal is presented in Figure 3.5.

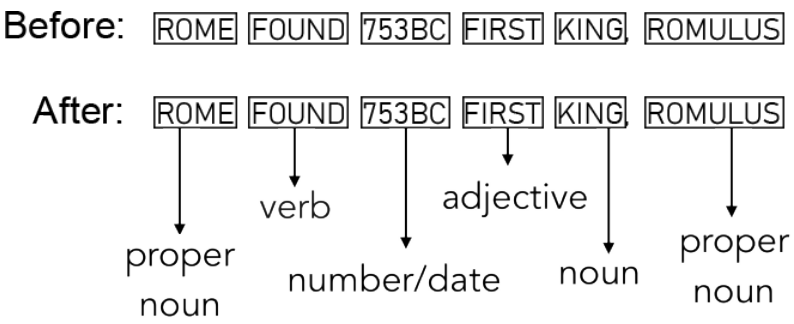


Figure 3.5: Example of Part of Speech Tagging.

3.3 Feature Extraction

After the data is normalized and clean, we can proceed to the next step in the pipeline. However, text cannot be used as input to train ML models before being converted into a numerical representation. Thus, the text needs to first be transformed in order to be served as input to the models. This transformation step is generally called Feature Extraction. In this section, we describe some of the most commonly used approaches.

3.3.1 Bag-of-Words

The Bag of Words model, or BoW for short, is a technique for extracting features from the text. This simple method collects a vocabulary of the words present in the text. After this, a measure (encoding) is applied for each document in the vocabulary to transform it into a embedding. This technique is used after the text is tokenized. Although optional, if used after removing stop words and stemming (or lemmatization), we can reduce the vocabulary and consequently the generated vectors. Depending on the task, this optional step can contribute to a performance improvement. Furthermore, this method loses any information about the order or structure of words in the text. An example of a Bag-of-Words from a corpus made of three sentences is as follows:

(S1) *A million dollars is not cool.*
(S2) *Do you know what is cool?*
(S3) *A billion dollars.*

From this corpus we can collect a BoW composed of the follwing words: “a”, “million”, “dollars”, “is”, “not”, “cool”, “do”, “you”, “know”, “what”, “billion” in no particular order.

3.3.2 *n*-Grams

The *n*-grams is another technique for extracting features from text. Similarly to BoW, it is normally used after the text is tokenized, and, when employed after stop word removal and stemming or lemmatization, it can reduce the extracted vocabulary and consequently the generated vectors. An *n*-gram is a sequence of items with length *n* in a text [1]. These items can be syllables, characters, or the most commonly used *n*-gram in NLP, words. Unlike Bag-of-Words, this technique groups sequences of words and can preserve the order or structure of words, to a limited extend. For instance, an example of *n*-grams collected from a sentence can be observed below:

(S1) *How are you today?*

From this sentence we can collect all 1-grams (also known as *unigrams*): “How”, “are”, “you”, “today”. All 2-grams (also called *bigrams*) that can be found are “How are”, “are you” and “you today”. Moreover, using the same sentence, the only 3-grams (also known as *trigrams*) we can get are “How are you” and “are you today”, and so on.

3.3.3 Data Representation

After we have our text split into *n*-grams or BoW, we can transform it into numerical vectors so that our ML model can process it. To do this transformation, we can apply three different encoding methods that will be presented next along with an example of BoW. For the example we used the corpus composed of two sentences:

(S1) *The mouse ran up the clock.*
(S2) *The mouse ran down.*

3.3.3.1 One-hot encoding

The simplest encoding method represents *n*-grams or BoW as a binary vector, indicating a token's presence (1) or absence (0) in the text. An example of a BoW with this encoding can be observed in Table 3.1.

	the	mouse	ran	up	clock	down
S1	1	1	1	1	1	0
S2	1	1	1	0	0	1

Table 3.1: Example of Bag of Words with One-hot encoding.

3.3.3.2 Count encoding

This encoding method represents *n*-grams or BoW as a vector indicating the frequency/count of a token appearance in the text. An example of a BoW with this encoding can be observed in Table 3.2.

	the	mouse	ran	up	clock	down
S1	2	1	1	1	1	0
S2	1	1	1	0	0	1

Table 3.2: Example of Bag of Words with Count encoding.

3.3.3.3 TF-IDF encoding

Term frequency-inverse document frequency, or TF-IDF for short, is a more complex method used to represent *n*-grams or BoW as vectors indicating the token importance in a corpus. Contrary to the first two methods, this one penalizes common tokens and gives more importance to unique ones. For instance, if a word occurs several times in a document while not appearing in the rest of the corpus, it means that it has a higher TF-IDF value. The TF-IDF value can be calculated by multiplying the number of times a word appears in a document (term frequency) with the number of documents the word appears in (inverse document frequency). The Equation 3.1 shows how TF-IDF is calculated:

$$TF\text{-}IDF(d,t) = TF(t,d) * IDF(t) \tag{3.1}$$

1. **Obtaining the TF (term frequency) for each pair (term, document).**
   For each document, we calculate the frequency of each word, and then we divide it by the

total number tokens.

$$TF(t,d) = \frac{feq(t,d)}{\sum_k feq(k,d)} \tag{3.2}$$

2. **Obtaining the IDF (inverse document frequency) for each term.**
   For each word, we calculate the number of documents in which that word exists. We then calculate the total number of documents in the dataset. Lastly, we obtain the inverse document frequency for each word by dividing the total number of documents in the dataset by the number of the word appearance and then applying the log2 function to the result.

$$IDF(t) = \log_2\left(\frac{|D|}{|d : t \in d|}\right) \tag{3.3}$$

An example of this encoding being applied to a BoW can be observed in Table 3.3.

	the	mouse	ran	up	clock	down
S1	0	0	0	0.16	0.16	0
S2	0	0	0	0	0	0.25

Table 3.3: Example of Bag of Words with TF-IDF encoding.

3.3.4 Word Embeddings

Even though the previous approaches are simple and, most of the time, effective, in some tasks they can have a significant disadvantage. For instance, if the task consists of finding the similarity between sentences or words, these methods can, in some cases, produce disappointing results, as demonstrated in the following example [5]:

(A) *How can I help end violence in the world?*
(B) *What should we do to bring global peace?*

Although these two sentences seem to have a very close meaning, if we analyze them using a frequency/count approach, the similarity will be zero since they have zero words in common. To tackle this issue, another approach can be used: word embeddings.

Word embeddings are feature vectors representing words, and contrary to techniques such as *n*-grams, where each word must represent a dimension, embedding vectors are short and dense. This approach understands the semantics of the word. By understanding this role of the word in a text, the word embeddings method can achieve more accurate results at detecting similar words. The cosine similarity metric is often used for measuring the similarity between two words. In this case, when two words are similar, their vectors will be similar as well. Many algorithms try to implement this approach, striving to learn word embeddings from data. Next, we will mention the most commonly used.

3.3.4.1 Word2vec

Word2vec (2013) [37] is one of the most popular context-free word embeddings implementation. Word2vec is based on prediction and it uses a two-layer neural network that is trained to predict a word's context in a text or a missing word out of a sequence. When training, this implementation assures that if two words are used in the same context, they should have similar embeddings (vectors). Thanks to that, relations between similar words are created, such as male-female, singular-plural, and cases like $King - Man + Woman = Queen$ occur where arithmetical operations are possible.

3.3.4.2 GloVe

GloVe (2014) [46] is another context-free word embeddings implementation. This method is based on count and it tries to generate vectors with ratios of co-occurrences of words from a corpus. This implies that the method is trained to retrieve the probability of a word appearing next to other words. One of the advantages of this method over the word2vec method is that is requires less training time.

3.4 Machine Learning Classifiers

Instead of using manually crafted rules (rule-based), ML classification algorithms, when supplied with a pre-labeled training set, observe and learn from data. With these observations, the algorithm produces a classification model that makes decisions and labels unclassified data. There are many ML algorithms for classification problems. We will describe the most widely used in this ML branch (NLP) for Text Classification, such as Support Vector Machines, Naive Bayes, Logistic Regression, among others.

3.4.1 Naive Bayes

Naive Bayes (NB) is a simple but powerful probabilistic algorithm in text classification. It is designated as *naive* because it makes the assumption that the features of the data are independently distributed, which means that, although it is wrong in most cases, it still performs reasonably well. In the case of text, we assume that every word in a sentence is independent of the others. This assumption makes the model focus on the presence of individual words and not their order and, consequently, makes it perceive sentences with the same words as being equivalent (e.g., "this car is nice", and "car nice is this" are classified the same). Naive Bayes is based on Bayes's Theorem, which is used to calculate the conditional probability of A occurring based on prior knowledge that the event B already happened and is defined by:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{3.4}$$

3.4.2 Logistic Regression

Logistic Regression (LR) is another classification algorithm based on the concept of probability. Similarly to Linear Regression, the Logistic Regression also computes a weighted sum of the input features (plus a bias term) [20], as presented in Equation 3.5:

$$h(x) = \beta_0 + \sum \beta_i x_i \tag{3.5}$$

The difference between both models is that the Logistic Regression model applies the logistic function, also known as sigmoid function, to the return Hypothesis value, forcing the model to output a probability value between 0 and 1, as presented in Equation 3.6:

$$h(x) = \frac{1}{1 + e^{-(\beta_0 + \sum \beta_i x_i)}} \tag{3.6}$$

3.4.3 Support Vector Machines

A Support Vector Machine (SVM) [14] is a supervised machine learning algorithm for classification that tries to find a line, denominated hyperplane, that splits a dataset into two classes (Figure 3.6). The optimal hyperplane is the one that maximizes the distance between this line and the closest points of each label, known as support vectors. SVMs usually attain good performance compared to other approaches and work well with small datasets. However, this algorithm shows its vulnerability when working on datasets with more noise or with overlapping classes, i.e., classes that are not linearly separable. Nevertheless, this is the algorithm that showed the best results in the literature review (Chapter 2).

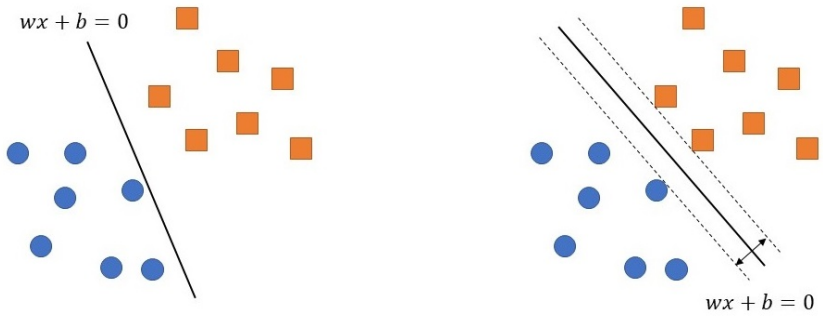


Figure 3.6: Two possible separating hyperplanes. The right-hand side figure shows a hyperplane that maximizes the margin. Adapted from [39].

3.4.4 Decision Tree

A Decision Tree (DT) is another supervised machine learning algorithm. It is an acyclic graph that can be used to make decisions, and in each branching node of the graph, a specific feature is examined. If the value of the feature is below a calculated threshold (in case we are using

numerical features but also work with nominal features), then one branch is followed; otherwise, the other branch is followed. Finally, on the last node, the leaf, the decision is made about which class the example belongs to [11]. This classifier has the significant advantage of being simple to understand and interpret. Figure 3.7 illustrates an example of a Decision Tree.

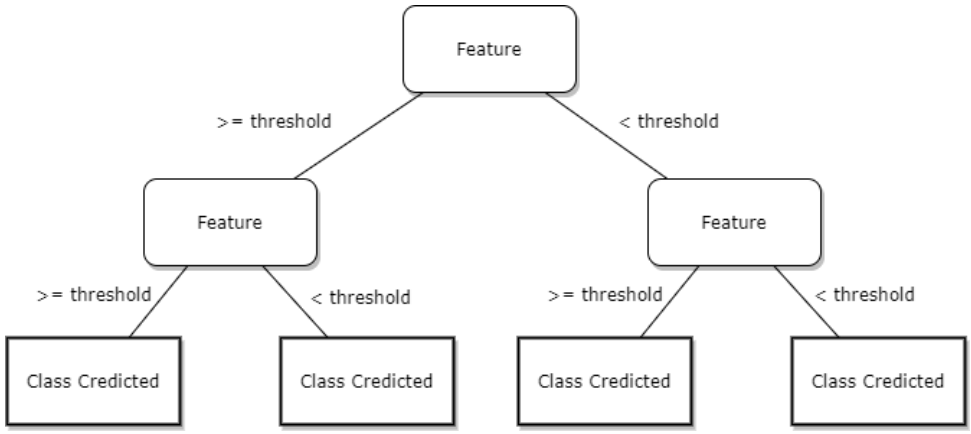


Figure 3.7: Example of Decision Tree.

3.4.5 Ensemble Learning

Ensemble Learning combines multiple models to improve the outcome. There are two types of Ensemble Learning: homogeneous and heterogeneous. The former means that all the models that contribute to the Ensemble, referred to as ensemble members, are of the same type. In the case of the latter, the ensemble members are of different types. Moreover, the Ensemble model can be sequential, parallel, or a combination of both. The main approaches, illustrated in Figure 3.8, are Bagging, Boosting, and Stacking.

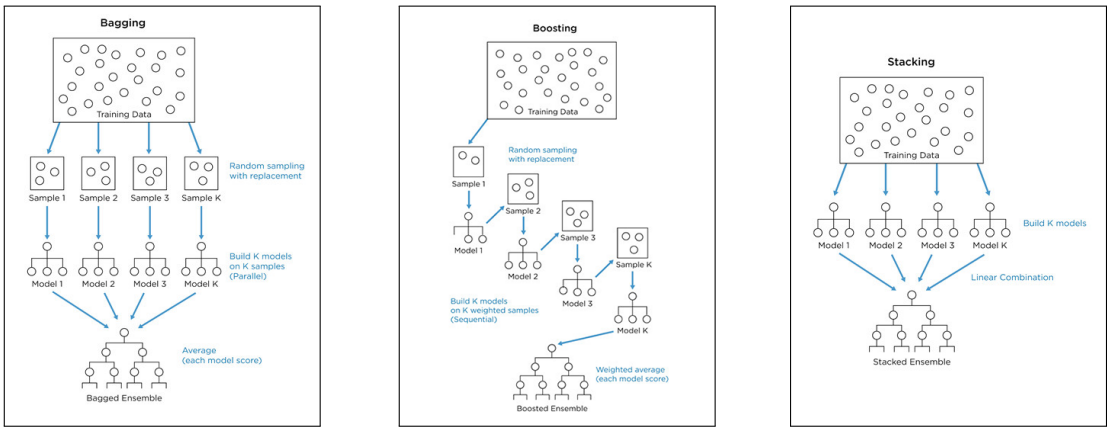


Figure 3.8: Images describing the three main approaches of Ensemble Learning. Retrieved from [10].

3.4.6 Random Forest

Random Forest is a non-linear classifier composed of a multitude of decision trees with a random set of features, and it is one of the most widely used ensemble learning algorithms [11]. Each decision tree in the random forest predicts a class, and the class with the most votes for the random sample of trees is the final prediction. It is a simple concept but one that works very well and yields excellent results. Figure 3.9 illustrates an example of a Random Forest.

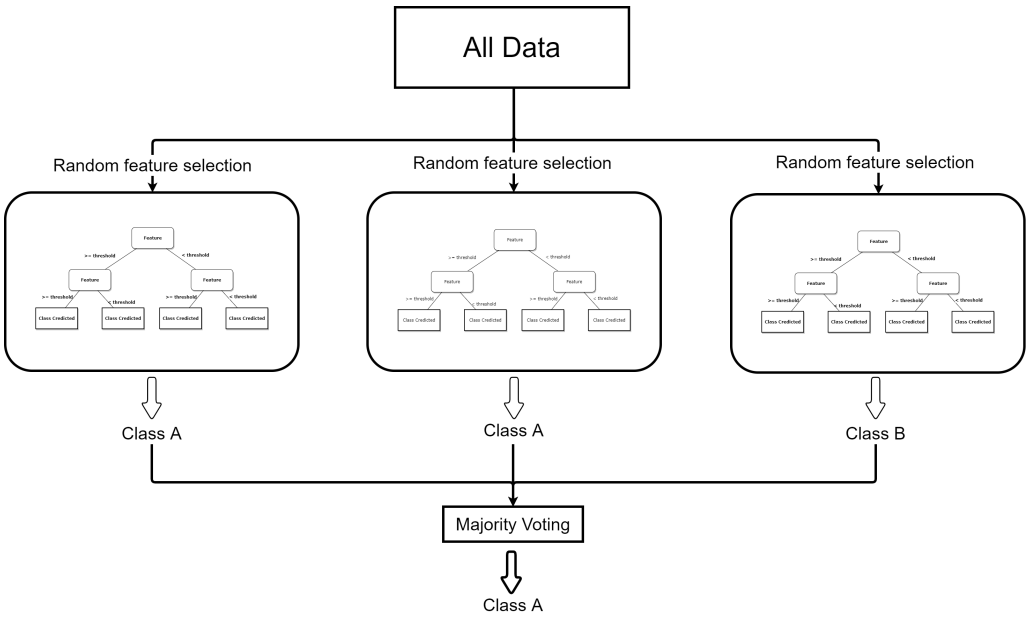


Figure 3.9: Example of a Random Forest.

3.4.7 Performance Assessment

Evaluating the model's performance is an important task to understand how the system will perform with different data. Many metrics can be useful to assess the performance of a classification model. In this project, we resort to the following widely used metrics to ensure we understand the results: confusion matrix, accuracy, precision, recall, and f1-Score. Evaluating the model with the same data with which it was trained presents a biased performance measurement, but we can overcome this by using cross-validation.

3.4.7.1 Cross Validation Techniques

The Holdout Method is useful to evaluate the model's performance quickly. This technique splits the training set into two parts: a training and a validation set. However, by doing this, we reduce the size of the training dataset. To overcome this challenge, we can use other cross-validation methods, such as K-fold Cross-validation.

The K-fold Cross-validation is a technique that splits the training dataset into k same-size folds. For each fold, a classifier model is trained with the remaining folds; in other words, at each iteration, the current fold serves as the test set, and the remaining folds serve as the training set. The performance measure reported by k-fold cross-validation is then the average of the k score values. In addition, we can also obtain the standard deviation of the k score values; with this measure, we can have an overall view of how accurate the model is.

3.4.7.2 Performance Metrics

Performance metrics have a crucial role in any ML pipeline. We can determine if we are making progress with these metrics and understand how good (determined by a score) the model created is.

Confusion Matrix

The confusion matrix is a concise way to evaluate a model's performance with information relative to the number of correct and incorrect predictions for each class, as shown in Figure 3.10. Although it is not a performance metric, the confusion matrix helps understand the overall view of the model's performance. It incorporates the following information:

- **True Positive (TP):** correctly predicted as positive;
- **True Negative (TN):** correctly predicted as negative;
- **False Positive (FP):** incorrectly predicted as positive;
- **False Negative (FN):** incorrectly predicted as negative;

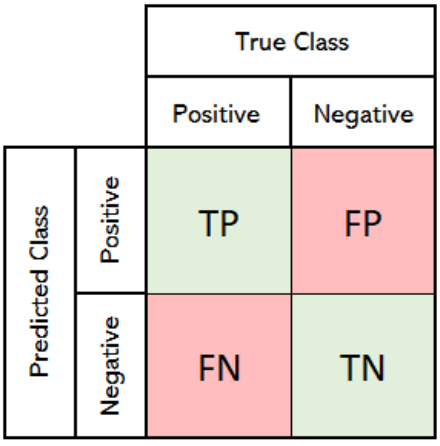


Figure 3.10: Confusion Matrix.

With the information present in the confusion matrix, we can calculate the performance metrics, such as accuracy, precision, recall, and f1-Score, as described below.

Accuracy

Usually, the models in a classification task, as shown in the Literature Review chapter (Chapter 2), are evaluated using the accuracy metric, which is defined by the number of correctly classified examples divided by the total number of classified examples [11]. Accuracy can be calculated as follows:

$$Accuracy = \frac{TP + TF}{TP + TF + FP + FN} \tag{3.7}$$

Precision

The precision metric is the ratio between correct positive predictions and the total number of positive predictions [11]. This metric can be calculated as follows:

$$Precision = \frac{TP}{TP + FP} \tag{3.8}$$

Recall

The recall metric is the ratio of correct positive predictions to the overall number of positive examples in the dataset [11]. This metric can be calculated as follows:

$$Recall = \frac{TP}{TP + FN} \tag{3.9}$$

F1-Score

The f1-score metric is the harmonic mean of the precision and recall into a single metric. The classifier will only get a high F1 score if both recall and precision are high. This metric can be calculated as follows:

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2 * TP}{2 * TP + FP + FN} \tag{3.10}$$

Area Under the ROC Curve

The ROC curve is a plot (Figure 3.11) of the True Positive Rate (or recall) against False Positive Rate (the proportion of negative examples predicted incorrectly) used to build up a summary picture of the classification performance [11]. By calculating the area under this curve (AUC), we get a simple way of quickly assessing the models' performance. The higher the area under the ROC curve (AUC), the better the classifier, with 1 being a perfect classifier, and 0.5 the baseline and as good as a random classifier.

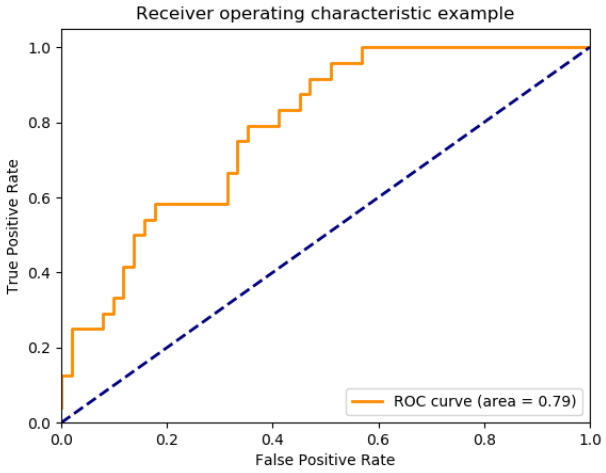


Figure 3.11: Example of AUC. Retrieved from [38].

3.5 Deep Learning Model

This sections explains the building blocks of neural networks that comprise the deep learning approach used by the present work. Deep learning is a group of methods inspired by how the human brain works when making decisions. It offers many benefits when placed against the more traditional methods, such as performing automatic feature extraction. One of the downsides of this architecture is the need for more training data than traditional algorithms. Some of the most known Deep Learning Models are BERT and GPT-2, both language models that will be described in the following sections. Language models have the objective of determining the probability of a given word sequence occurring in a sentence.

3.5.1 BERT

Bidirectional Encoder Representations from Transformers, more commonly known as BERT, is a recent implementation (2018) [16] of contextual word embeddings based on Transformer [58] models that can obtain state-of-the-art results in many NLP tasks. Transformer is an attention model capable of understanding the contextual relations between words in a text. BERT was trained with BooksCorpus (800M words) and English Wikipedia (2500M words), and contrary to context-free models, such as Word2vec or GloVe, contextual models such as this one can generate multiple embeddings (vectors) for a word. For instance, the word “bank” in one sentence can represent “bank deposit”, and in another a “river bank”. Furthermore, it is a bidirectional model, which means that it uses both left and right contexts to make a prediction. Using the same example as earlier, in the sentence “I made a bank deposit”, a unidirectional representation of bank is only based on “I made a” but not “deposit”; instead, BERT represents “bank” using both its left and right context — “I made a [____] deposit” [50].

3.5.2 GPT-2

GPT-2 (Generative Pre-trained Transformer 2) is another recent implementation of contextual word embeddings based on transformer models [49]. A direct successor to GPT, GPT-2 was trained with 40GB of text on several different topics collected from the internet. Its purpose is solely to predict the next word, given all of the previous words within some text, i.e., the text context. The data in which it was trained is from non-specific domains (e.g., news, books, or Wikipedia), and contrary to other language models trained with more specific training sets, this model can achieve state-of-the-art results at many NLP tasks.

3.5.3 Fine-tuning

Since training a model from scratch has some drawbacks, such as needing significant computing power, a substantial corpus of data, and a lot of time, another approach is used: fine-tuning a pre-trained model. Fine-tuning a model consists of training an already pre-trained model on a new dataset with a minimal learning rate. With the advantage of being computationally less intensive than just pre-training the model, this approach also has the potential to achieve meaningful improvements by incrementally adapting the pre-trained features to the new data. Figure 3.12 shows how the fine-tuning process works.

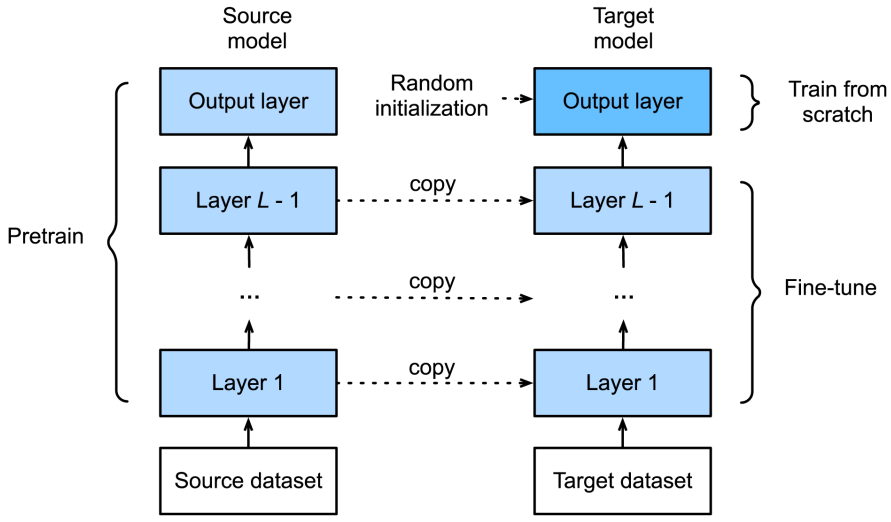


Figure 3.12: Fine-tuning a model. Retrieved from [62].

3.5.4 Perplexity

The perplexity metric is commonly used for evaluating language models and can be defined as the exponentiated average negative log-likelihood of a sequence, as can be observed in Equation 3.11.

$$\text{PPL}(X) = \exp\left\{-\frac{1}{t}\sum_{i}^{t}\log p_\theta(x_i|x_{<i})\right\} \tag{3.11}$$

This metric can be used to understand how close a sample is to the train data used. Therefore, a low perplexity indicates that the model is good at predicting the sample.

3.6 Language Barrier

A last vital topic to address is the current state of the NLP regarding Low Resourced Languages. The vast majority of research in NLP focuses on specific, well-resourced languages, leaving the rest understudied. Contrary to High Resourced Languages, such as English, which is by far the most well-resourced, Low Resourced Languages (LRL) are defined as less studied, resource-scarce, less commonly taught, or low-density languages [35].

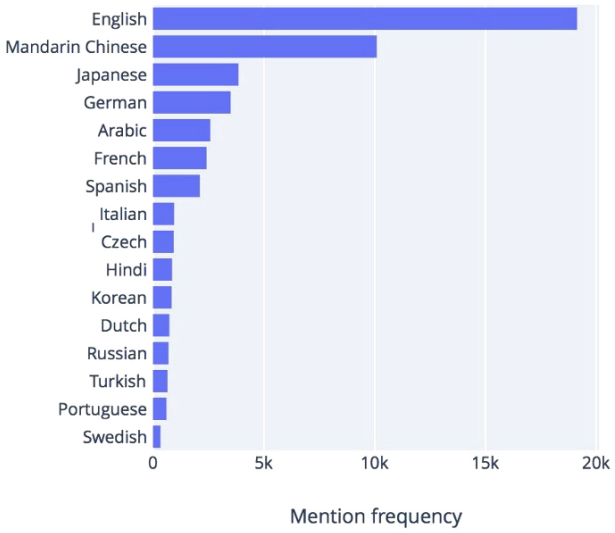


Figure 3.13: Languages addressed by ACL research. Retrieved from [32].

This project focuses on detecting fake news written in Portuguese. Even though Portuguese is one of the languages ACL research addresses the most (Figure 3.13), Portuguese still has limited resources (e.g., few annotated corpora), when compared to English. Due to limited resources, most NLP supporting tools have sub-optimal performance. Furthermore, Neural Network classifiers can also present a challenge since they require a considerable amount of data to train on. Nevertheless, we will use tools that already have features and support of Portuguese text to train the model, such as Spacy [26], Stanza [21] or even tools offered by research groups like NLX [43]. Furthermore, we will explore research networks such as Portulan Clarin [13] to find the needed corpora.

3.7 Summary

This chapter focuses on the basic concepts of NLP that will be required to understand the content of this dissertation. It starts by presenting the most critical steps to preprocess a text and feature extraction techniques. We then present the basics for machine learning classification, describing some classifier models and ways to assess and validate their performance. We also mention deep learning architectures such as BERT and GPT-2, known as state-of-the-art in many NLP tasks. We end the chapter with a discussion about choosing a low resourced language to work with and the associated limitations.

Chapter 4 presents the proposed methodology utilized for the development of this research work.

Chapter 4

Methodology

This chapter describes the methodology used in this project. We will explain in more detail all the resources used, from the collected corpus and the dataset generated to the choices related to NLP tools. Additionally, we will describe and discuss some characteristics of the dataset.

4.1 Resources

This section introduces the corpora and external resources used in our experiments to build the models capable of detecting fake news. As this project focuses on detecting fake news written in Portuguese, we need to consider some particular circumstances. Even though Portuguese is one of the most widely spoken languages [60], it still has limited linguistic resources available compared to English. Due to this limitation, most NLP supporting tools have sub-optimal performance. Nevertheless, we will use tools that already have features and support of Portuguese text to train the models.

4.1.1 Corpora

In any ML project, we start by collecting data. The success of an ML project depends, among several other aspects, on the quantity and quality of data. It was necessary to gather a significant corpus from web-based newspaper articles, both genuine and fake news.

Given the nonexistence of an annotated dataset telling apart fake and genuine news in Portuguese, we follow a silver standard approach [23] with automatically annotated data [12] when collecting news items for both classes. By using this approach, the label (fake or not) of scraped news articles from each website is considered to be the type of news that the website is known to publish. URLs of the news, which were collected between November and December 2020 and included in the dataset, are made available[1].

[1] drive.google.com/file/d/1jqiMxbcH6H4ozA3zbTnxphriQx1fKi4G/view

Furthermore, some scraped articles were deemed unusable for two specific reasons. Firstly, some genuine news articles collected were tagged, by the source, as opinion articles, which have a different structure from regular news. Secondly, some fake news authors identified them as having a humorous nature; thus, from our perspective, these should not be considered fake news.

We tried to obtain as much information as possible for every article, even though several articles were missing some parts. With this in mind, we aimed to obtain:

- **Title and subtitle:** Every newspaper article had a title, but some were missing a subtitle. Even though we do not directly use these features, it is a possible improvement, since it already has been used in previous works such as Horne and Adali (2017) [27];
- **Text:** The text of the news itself. This component was the most studied in this research.
- **Date:** The date of the news publication, which not all records had. With this feature, we tried to limit our collection in order to get the same period in each type of news.
- **Tags:** Tags specified in the news articles. Most of the records have this feature, but we do not focus on it. We mostly use this feature as a filtering method, since the tags indicate some cases of unusable news (e.g., articles that only describe an interview).
- **Author:** Name of the person who wrote the news. Although this information is diverse in genuine news, the same cannot be said for fake news, which often only have one author for each domain – or, in some cases, do not even have an author associated.
- **Domain and URL:** The news URL and the domain representing the newspaper website from which the articles were acquired.

As far as the collection procedure is concerned, we chose a different approach for each type of news, as described below:

4.1.1.1 Fake News

Although there are several online corpora of fake news[2], to the best of our knowledge none is focused on Portuguese. We create a corpus by scraping websites that are known to publish fake news contents[3] [4]. From those available, we have chosen five: *Bombeiros24*, *JornalDiario*, *MagazineLusa*, *NoticiasViriato*, and *SemanarioExtra*.

In order to scrape these websites, a Python script was created, using several techniques and libraries depending on what was most suitable for each domain. The first step of fake news collection was to gather the links to all the news on each website. Each website is divided into categories, which made the process simpler. The method was the same for every domain: go through all the news in each category until reaching the last one, collect all the available news links, and then collect the information required for each article. Some websites had pagination, and others had

[2] https://github.com/sumeetkr/AwesomeFakeNews

[3] https://www.sabado.pt/portugal/detalhe/be-pede-audicao-da-erc-para-esclarecer-registo-de-sites-de-fake-news

[4] https://www.dn.pt/edicao-do-dia/11-nov-2018/fake-news-sites-portugueses-com-mais-de-dois-milhoes-de-seguidores–10160885.html

a scroll-down interactive JavaScript program that dynamically loads more news when the page is scroll-down.

For websites with pagination, the Python library Beautiful Soup[5], capable of extracting data out of HTML and XML files, was used. This library allows the selection of HTML elements from a website. In this case, a simple process is followed: for each page, select all the news, collect their links, then click on the next page button and repeat this until we reach the last page and have collected all the news links.

The process was more complex for pages that used the scroll-down script. In this case the procedure was divided into two phases. The purpose of the first phase was to load all the news. We did this by using the Python library Selenium[6], capable of automating web browser interaction, that enabled us to simulate the click on the END key that caused the website to scroll down to the bottom, forcing it to load more news. This phase was repeated until no more news articles were loaded. In the second phase, all the news in the category were loaded, then we selected all the news and collected their links.

After the first step of the fake news collection, we had the links to all the news from every chosen website. The last step consisted of using Beautiful Soup one last time to select and collect all the information needed from each news article. In the end, we have a corpus of 10 343 news articles from five different newspaper sources (the newspaper source distribution can be observed in Figure 4.1) posted between 2017 and 2020. The chart below illustrates their distribution by domain.

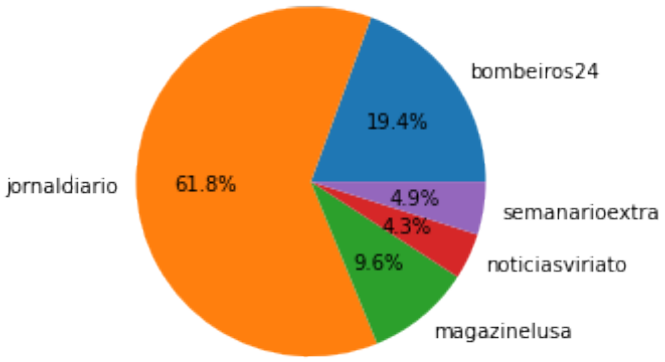


Figure 4.1: Distribution of fake news article domains after scrapping.

[5] https://www.crummy.com/software/BeautifulSoup/bs4/doc/
[6] https://www.selenium.dev/

4.1.1.2 Portulan Clarin Corpus

For genuine news, we first chose to get an already scraped corpus [34] from Portulan Clarin[7]. This corpus is a collection of Portuguese news articles from the whole year of 2016. The news articles are from different newspaper sources and categories that range from generic (daily) news and magazines to technology or sports news. In total, we have 603 000 news articles.

4.1.1.3 *Público* News Corpus

We ended up also collecting another corpus ourselves because the Portulan Clarin corpus had, from our perspective, some flaws that made it unsuitable for training a model: the corpus was composed of news only from 2016, and this could present some challenges when using *n*-grams since all the fake news were posted between 2017 and 2020; furthermore, contrary to the fake news sources, some news articles were from categories like technology or sports ; and lastly, some of the news had problems with the scrapping, such as having text that was not from the article itself. To solve this, we built the genuine news corpus by scraping news articles from *Público*, one of Portugal's most reputable quality newspapers. In total, we collected 110 066 news from the same period as the news included in the fake news corpus.

4.1.2 Machine Learning Resources

We explore multiple resources to get the best results for processing the news articles, mainly the spaCy[8], NLTK[9], and Stanza[10] libraries and some tools provided by the NLX-Group[11]. We ended up using a mix between NLTK for the Portuguese stopwords list (the full list can be found in Appendix A), the pySpellChecker[12] library for spell checking, the Textatistic[13] library for calculating the readability indices, and spaCy Portuguese model for being the most resourceful and capable of doing the other NLP tasks (specifically tokenization, part-of-speech tagging, named entity recognition, and lemmatization). We also use Scikit-Learn[14] implementations of the classifiers we have trained and the functions *CountVectorizer* and *TfidfVectorizer*, part of the same library, to calculate the *n*-grams encoded as count and tfidf, respectively.

[7] Portulan Clarin is a research infrastructure for the science and Technology of language.
[8] www.spacy.io/models/pt
[9] www.nltk.org/howto/portuguese_en
[10] stanfordnlp.github.io/stanza/models
[11] www.nlx.di.fc.ul.pt/tools
[12] www.github.com/barrust/pyspellchecker
[13] www.erinhengel.com/software/textatistic/
[14] www.scikit-learn.org

4.1.3 Deep Learning Resources

For the Deep Learning experiments, we employed the HuggingFace[15] library since it has all functionalities needed for our experiments, from already pre-trained models to the Trainer API capable of fine-tunning the models.

4.2 Approach

The primary goal of this dissertation is to produce a system to detect fake news using approaches usually adopted in forensic linguistic analysis. Contrary to fact-checking systems, our system takes more of a linguistics-based approach, as seen in Chapter 2. Forensic linguistics has been successful in other tasks, such as authorship attribution or bias detection in text [54], and we argue that it can have a significant potential to detect fake news.

During the dataset collection phase (described in Section 4.1.1), we analyzed the collected fake news to understand what made them different from the genuine news, and we found some interesting and probably valuable characteristics that can be used as features in our model.

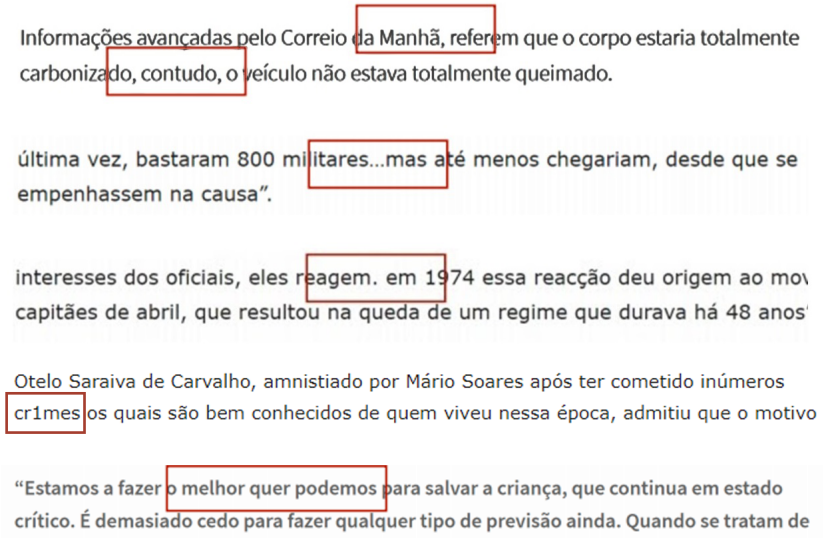


Figure 4.2: Example of misused punctuation and spelling mistakes in fake news articles.

The linguistic and stylistic characteristics that were most noticeable in fake news were spelling mistakes, but sometimes more than typical typos can be observed. For instance, some words were obfuscated and contained numbers or special characters (e.g., the word crime is often spelled *cr1me*, *cr!me* or even *c***e*). Furthermore, ellipsis, commas, or periods were used uncommonly frequently or in the wrong place, as shown in Figure 4.2. Moreover, we noticed some structural

[15] www.huggingface.co

differences. For example, news text is usually shorter (in number of sentences) in fake news when compared to genuine news.

After collecting the datasets, we analyzed and explored the data to confirm some of the observations made earlier and possibly obtain new features. Firstly, we confirmed most of our previous observations: (a) fake news articles are almost always shorter than genuine news (Figure 4.3), and fake news have obfuscated words, which is observed significantly less in genuine news (Figure 4.4). Moreover, we discovered new properties, such as the fact that fake news articles use more expressive punctuation such as the exclamation mark and the ellipsis (Figure 4.5 and Figure 4.6), compared to genuine news.

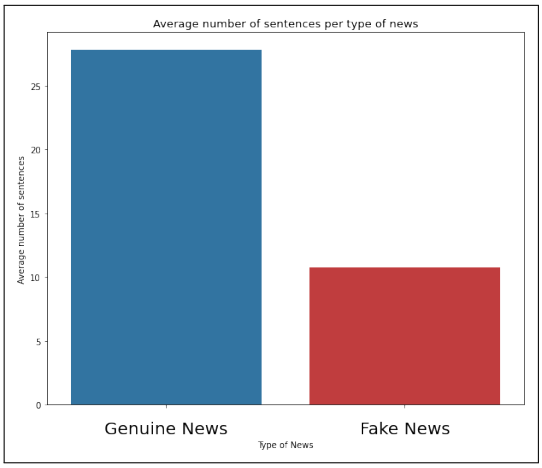


Figure 4.3: Average number of sentences per type of news, confirming initial observations.

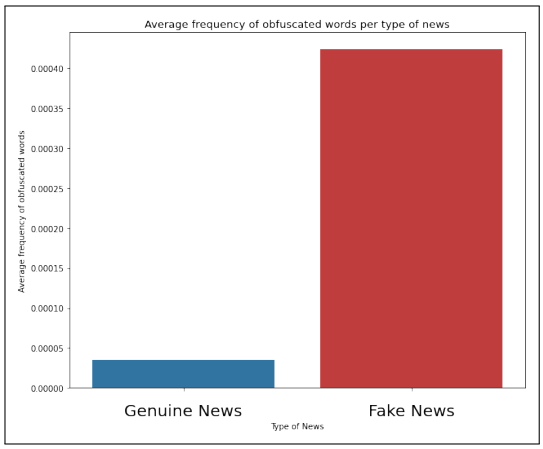


Figure 4.4: Average frequency of obsfucated words per type of news, confirming initial observations.

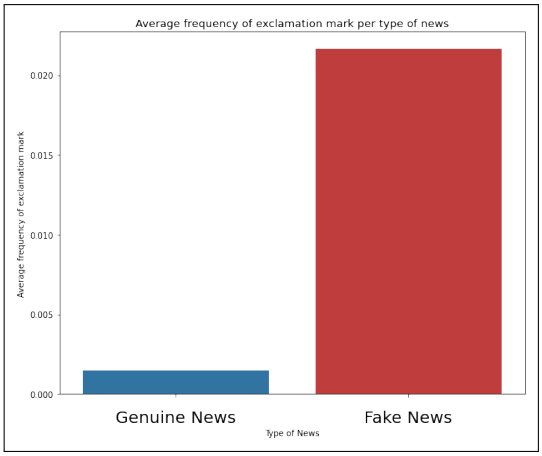


Figure 4.5: Average frequency of exclamation mark per type of news.

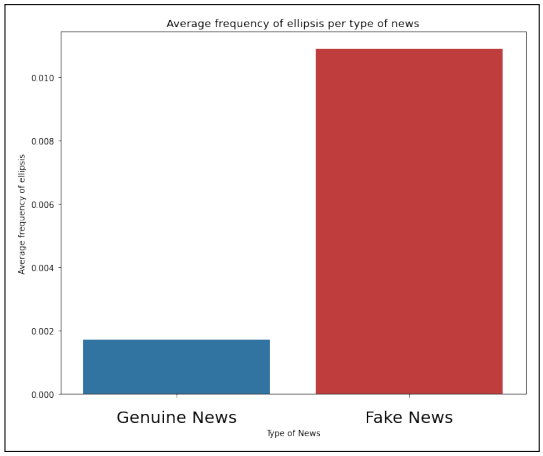


Figure 4.6: Average frequency of ellipsis per type of news.

Furthermore, we explored the bigrams to analyze the different expressions present in both types of news. With the bigrams we can see a clear difference between the 2 types of news (see e.g., Figure 4.7 and Figure 4.8; more details can be found in Appendix B).

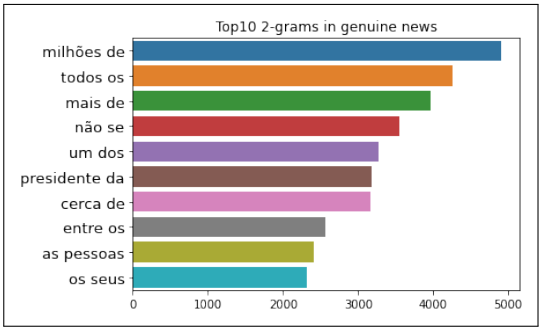


Figure 4.7: Top 10 bigrams in genuine news.

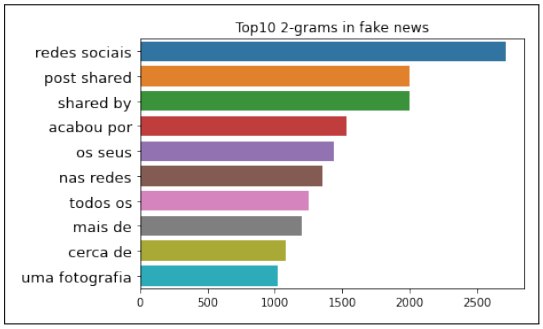


Figure 4.8: Top 10 bigrams in fake news.

With these observations, we can create a model using Machine Learning classification methods and use the mentioned linguistic observations and characteristics as features to train it. However, first, we need to have a pipeline defined to process the news articles. This pipeline can be divided into three steps: data pre-processing, feature extraction, and Classification Process. The first step will be responsible for cleaning and making sure there are no duplicated news or scraping errors. For this, we will use the methods mentioned in Section 3.2. The second step will be devoted to extracting and converting the news articles into linguistic features from the text – not only the vector representation of the news documents (i.e., Bag-of-words, *n*-grams) but also statistics of the texts such as the observations aforementioned. In the last step, we train several machine learning models with the representations obtained to hopefully detect fake news with high accuracy.

4.2.1 Feature Extraction

In order to train a classification model, we need to convert the main text of the news articles into a set of linguistic features. These features (described in more detail in Table 4.1) can be divided into four categories, explained in the following sections.

4.2.1.1 *N*-grams:

We calculate the vocabulary composed of all lemmatized tokens in the documents and subsequently extract a set of unigrams, bigrams, and trigrams, encoded as normalized counts and with TF-IDF. In order to avoid the influence of named entities, we adopt an approach that obfuscates them so we can focus on a forensic linguistics approach that is topic-independent and not affected by temporal issues. To obfuscate the entities, we make use of spaCy's named-entity recognition to replace classified entities with their respective label (e.g., "*Cristiano Ronaldo*" becomes "*[PERSON]*"). Although the English spaCy Named-entity Recognition (NER) model has 18 types of entities (including laws, languages, date, quantity, among others), the Portuguese model (the one we have used) only has three: person, organization, and location.

4.2.1.2 Frequencies:

We extract a collection of relative frequencies, including the frequency of each punctuation character (e.g., "freq !" and "freq ..."), the frequency of each Part-of-Speech tag (e.g., "freq interjections" and "freq adverbs"), and the frequency of each type of adverb (including negation, affirmation, interrogation, quantity, exclusion, inclusion, mode, time, local, connective, and doubt). All of the frequency-related features are defined in more detail in Appendix A.

4.2.1.3 Text Statistics:

We also obtain a set of statistical features: the number of paragraphs, sentences, tokens, stopwords, chars, and syllables. From these, we generate some averages: the average number of sentences per paragraph, words per paragraph, words per sentence and chars per word.

4.2.1.4 Readability:

We compute a set of features that measure how easy it is to read a text. These include vocabulary richness (i.e., how diverse the vocabulary used by an author is, in other words, the ratio of unique tokens), and ratios such as the percentage of long words (> 12 characters), obfuscated words [30] (words with numbers or special characters, e.g. "*cr1me*"), misspelled words, polysyllable words (> 2 syllables). We also make use of some well known readability indices, such as Flesch Reading-Ease [18], Flesch–Kincaid Grade Level [29], Gunning Fog [22] and SMOG [31]). We present a brief description of each readability index and the associated formula:

- **Flesch Reading-Ease**: This index presents a score between 1 and 100 that indicates a text passage readability, in which higher scores indicate an easier-to-read text. This readability index is calculated using the following formula:

$$206.835 - 1.015 \times \frac{word\_count}{sent\_count} - 84.6 \times \frac{sybl\_count}{word\_count}$$

- **Flesch–Kincaid Grade Level**: Despite this readability index being (inversely) correlated to the previous one (since it uses the same measures), they have different weighting factors. Instead of returning a score between 1 and 100, this index returns an estimation of the years of education required to understand a text. This readability index is calculated using the following formula:

$$-15.59 + 0.39 \times \frac{word\_count}{sent\_count} + 11.8 \times \frac{sybl\_count}{word\_count}$$

- **Gunning Fog**: Similar to the previous index, the Gunning Fog formula calculates the required grade level to read a text. Instead of using the syllable count, this index uses the count of words with three or more syllables. This readability index is calculated using the

following formula:

$$0.4 \times \left(\frac{word\_count}{sent\_count} + 100 \times \frac{polysybl\_count}{word\_count} \right)$$

- **SMOG**: Like the previous two, this index calculates the required years of education to understand a text. This readability index is calculated with the following formula:

$$3.1291 + 1.0430 \times \sqrt{30 \times \frac{polysybl\_word}{sent\_count}}$$

4.2.2 Dataset Description

To better understand the generated dataset, it is useful to investigate how feature values vary depending on the type of news. In Figure 4.9 we present most of the features in a normalized distribution per class (fake vs genuine), omitting *n*-grams and some less important frequency-related features. The outliers have been omitted from the plots, for greater readability. We can see that some features stand out in the collected datasets, due to the contrast between fake and genuine news.

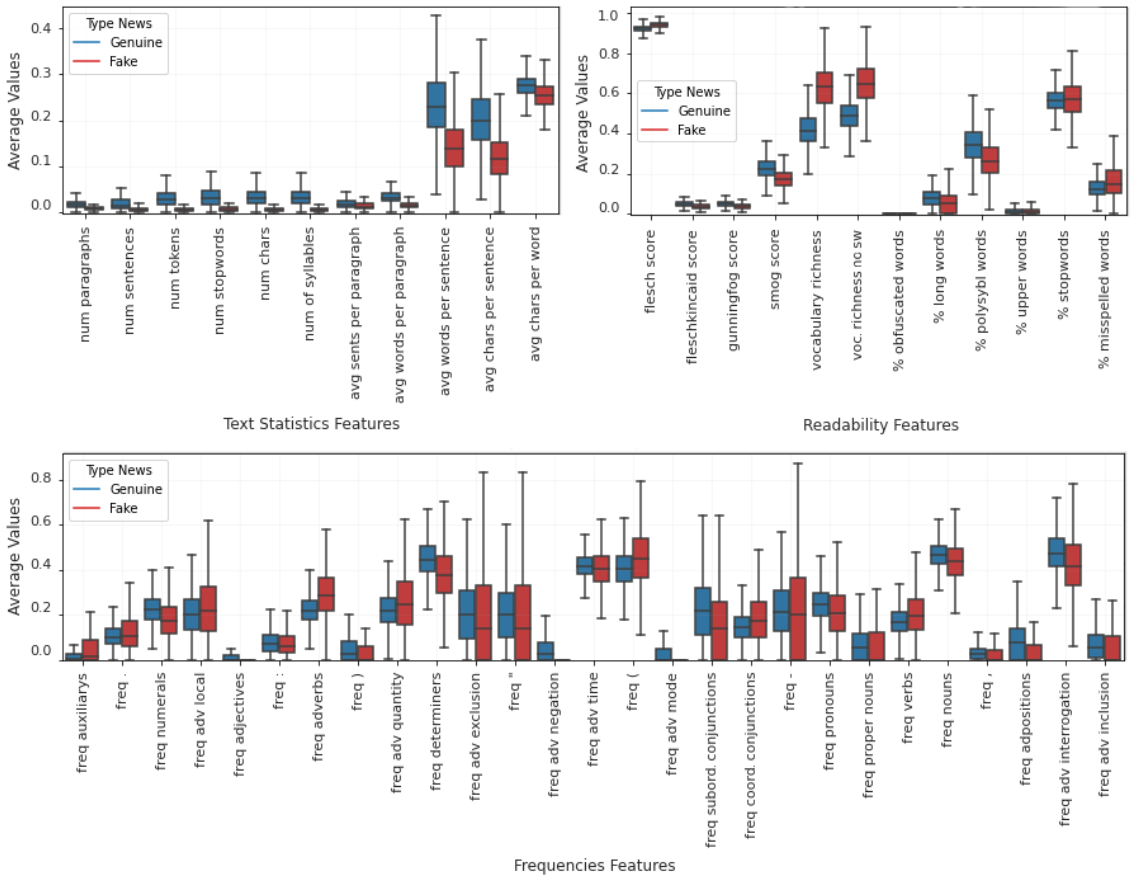


Figure 4.9: Distribution values per class for each feature set.

Feature	**Description**
Text Statistics	
num paragraphs	Number of paragraphs.
num sentences	Number of sentences [spacy].
num tokens	Number of tokens.
num stopwords	Number of stopwords [nltk].
num chars	Number of chars.
num of syllables	Number of syllables.
avg sents per para	Average number of sentences per paragraph.
avg words per para	Average number of words per paragraph.
avg words per sents	Average number of tokens per sentence.
avg chars per sents	Average number of characters per sentence.
avg chars per word	Average number of characters per word.
Frequencies	
freq punctuation *	Relative frequency of each punctuation character.
freq PoS tags *	Relative frequency of each PoS tag.
freq type of adverbs *	Relative frequency of each type of the adverb.
Readability	
vocabulary richness *	Measures of vocabulary diversity: ratio between the total number of words and the number of unique words – with or without stopwords.
readability indices *	Measures of text reading/understanding difficulty – *flesch*, *fleschkin-caid*, *gunningfog*, and *smog*.
% long words	Fraction of words with 12 or more characters.
% obfuscated words	Fraction of words containing punctuation or numbers.
% misspelled words	Fraction of words with spelling errors.
% uppercase words	Fraction of uppercase words.
% polysybl words	Fraction of of words with three or more syllables.
N-grams	
n-grams *	TF (counts) and TF-IDF of unigram, bigrams, trigrams. In total 600 *n*-grams.

Table 4.1: Features used to build the model for Fake News detection. A star ($*$) indicates that the feature is a feature set.

For instance, almost all text statistics features (e.g., "*num paragraphs*", "*num sentences*", "*avg sentences per paragraph*", among others) are more frequent in genuine news, since the text is usually longer than that of fake news.

We can also see in this contrast in some frequency features. For example, some POS tags like interjections and proper names are more frequent in fake news, while adjectives are more frequent in genuine news. Other observations suggest that fake news uses more diverse punctuation characters, showing that ellipses, exclamations, and question marks are more frequent in fake news.

Features related to readability indexes indicate that fake news articles are easier to read, in the sense that their writing is simpler. One unexpected value in this feature set is vocabulary richness, since the average value for fake news is higher than for genuine news. These values can be explained by the fact that fake news articles are almost always smaller than genuine news. Consequently, fake news has fewer chances of repeating the same token. Furthermore, genuine news pieces have more polysyllable and long words, while fake news items have significantly more obfuscated and misspelled words.

Regarding *n*-gram features, (lemmatized) word sequences such as "*primeiro ministro*" (prime minister), "*presidente*" (president), "*empresa*" (company), or "*milhão*" (million), occur much more often in genuine than in fake news. Conversely, words such as "*rede social*" (social media), "*mostrar*" (show), "*mulher*" (woman), or "*vida*" (life) are more frequent in fake news than in genuine news.

The dataset also shows that genuine news tend to reference entities more often than fake news, which results in a higher count in entity-related *n*-grams.

4.2.3 Feature-based Process

We conduct several experiments with each feature category and with multiple Machine Learning algorithms, specifically: Logistic Regression (LR), Linear Support Vector Machines (LSVM), Random Forest (RF), Decision Tree (DT), Gradient Descent (SGD), Naive Bayes (NB), and Gradient Boosting Classifier (GBC). We use Scikit-Learn's implementations of these algorithms and resort to using the default values of the hyperparameters as defined by the library, only specifying (when possible) the *class_weight* property to "balanced" (that makes the model use the inverse weighting from the training dataset, giving focus to the minority class), and for LR the Lasso penalty (l1).

To better assess the performance of each model, we use 5-fold stratified cross-validation. In each fold, we return the following metrics: accuracy, precision, recall, and f1-score. Although we report all these metrics, we mainly focus on two. The first is accuracy, which gives us a rough idea regarding the performance of the models due to the unbalanced nature of our dataset. However, we mainly rely on the macro average f1-score, due to the imbalanced nature of our dataset. Furthermore, we collect the feature importance for every model to understand the features that are found by each model to be more robust to distinguish between the fake and genuine news classes.

4.2.4 Deep Learning Process

We do some research with deep learning approaches since models like BERT and GPT-2 are considered to be state of the art in many NLP tasks. To obtain the most accurate comparison, we remake the tests described in the features-based section (Section 4.2.3) but this time without cross-validation and using a train test split of 80/20. Therefore, the tests performed for the deep learning approach were all done using the same train and test set. The work done can be separated into two main experiments, described in the following sections.

4.2.4.1 Contextual Word Embeddings Approach

In the first experiment, we use the BERT model to collect the word embeddings using the Transformers library from HuggingFace. For this purpose, we used two different BERT models: a multilingual BERT model[16] (bert-base-multilingual-cased) and a pre-trained BERT model for Brazilian Portuguese[17] (neuralmind/bert-base-portuguese-cased). We then employ the collected word embeddings from each news article as features for a regular classification model.

4.2.4.2 Perplexity Approach

In the second experiment, we make use of the perplexity metric given by a model after receiving a text to distinguish genuine from fake news. In order to make a model get better chances of distinguishing the two types of news, we needed to train it with our corpus. The problem is that training a model from scratch is very computationally intensive and requires a considerable amount of time. Instead, we opted for a less computationally intensive approach: fine-tuning a model, i.e., re-training a model on a new text corpus. We chose to fine-tune a GPT-2 model pre-trained with the Portuguese Wikipedia[18] (pierreguillou/gpt2-small-portuguese). We use GPT-2 because the perplexity measure is not well defined for masked language models like BERT.

After fine-tuning the model with a different number of news articles, we then calculate the perplexity for each article. To classify each article as fake or genuine, a simple classification approach that takes into consideration the perplexity score of each article was employed: we test a wide range of possible thresholds and choose the one that maximizes the macro average f1-score. The range of thresholds tested is defined with the help of each type of news' perplexity average and standard deviation (better detailed in Algorithm 1). For classifying a news article, if the perplexity is below the chosen threshold, we classify the article to be of the type of news that the model was fine-tuned on; if it is above that threshold, the article is classified as the other type (better detailed in Algorithm 2).

[16] https://huggingface.co/bert-base-multilingual-cased
[17] https://huggingface.co/neuralmind/bert-base-portuguese-cased
[18] https://huggingface.co/pierreguillou/gpt2-small-portuguese

Algorithm 1: Choosing the threshold that maximizes the macro average f1-score.

input : $train\_df$ - The training set.

```
// Calculate the perplexities for the train set.
```
$plxt\_df = calculate\_plxt(train\_df)$

```
// Divide the train set into two different perplexity
   dataframes: fake news and genuine news.
```
$fn\_plxt\_df = plxt\_df[plxt\_df['fakenews'] == True]$
$gn\_plxt\_df = plxt\_df[plxt\_df['fakenews'] == False]$

```
// Get the mean and standard deviation for both dataframes.
```
$fn\_plxt\_mean = mean(fn\_plxt\_df['perplexity'])$
$fn\_plxt\_std = std(fn\_plxt\_df['perplexity'])$
$gn\_plxt\_mean = mean(gn\_plxt\_df['perplexity'])$
$gn\_plxt\_std = std(gn\_plxt\_df['perplexity'])$

```
// Calculate the threshold range which to test. The lowest
   perplexity mean will always be the type of news with which
   the model was trained.
```
if $fn\_plxt\_mean \geq gn\_plxt\_mean$ **then**
  $max\_threshold = fn\_plxt\_mean + fn\_plxt\_std$
  $min\_threshold = gn\_plxt\_mean - gn\_plxt\_std$
else
  $max\_threshold = gn\_plxt\_mean + gn\_plxt\_std$
  $min\_threshold = fn\_plxt\_mean - fn\_plxt\_std$
end

```
// Predict with each threshold in the calculated range, and get
   the one that maximizes the macro average f1-score metric.
```
$threshold = max\_f1\_score(min\_threshold, max\_threshold)$

Alternatively to the simpler approach described – using a calculated threshold – we combine the calculated perplexity by one model trained with fake news and other trained with genuine news as features to an ML classification model.

4.3 Summary

This chapter presented the adopted methodology for conducting this research work. The feature extraction, all the external resources, as well as the proposed approaches were described in detail.

Chapter 5 reports the results for each experiment mentioned in this chapter, discussing some insights observed.

Algorithm 2: Classify a news piece as fake or genuine news.

input : *threshold* - The given threshold that will be used to classify between the two types of news.
input : *news_text* - News article main text.
input : *model_trained_with* - Type of news the model was trained with, in other words, the type of news that presents the lowest perplexity mean.

```
// Calculate the perplexity from a given text.
plxt = calculate_plxt(news_text)

// Use the given threshold to classify the calculated
   perplexity metric as fake or genuine.  If the perplexity is
   lower than the threshold, the text is classified as the type
   of news used to fine-tune the model.
if model_trained_with == FAKE then
    if plxt ≤ threshold then
        return FAKE NEWS
    else
        return GENUINE NEWS
    end
else
    if plxt ≤ threshold then
        return GENUINE NEWS
    else
        return FAKE NEWS
    end
end
```

Chapter 5

Results

This chapter shares, evaluates, and discusses the results obtained in this project for all the approaches proposed. Furthermore, we do feature analysis, exploring the most defining characteristics in a text that lead the models created into considering them fake.

5.1 Feature-based Approach

For running the experiments for this approach, we use an Intel® Core™ i7-3770K CPU with 3.50GHz (8 cores) with 32Gb of RAM. The results shown in Table 5.1 are the average performance scores for each model in the 5-fold stratified cross-validation setup in a binary (fake vs genuine news) classification problem. Almost all models report satisfactory performances in these experiments, and some are particularly good at classifying fake and genuine news. The Naive Bayes and LSMV are the worst-performing models tested, reporting the lowest accuracy and the lowest macro average f1-score. Both the Logistic Regression and Random Forest models achieve the best results and will be the models we will be discussing.

Model	Acc.	Weighted Average			Macro Average		
		P	**R**	**F1**	**P**	**R**	**F1**
Naive Bayes	0.77	0.93	0.77	0.82	0.63	0.85	0.64
Linear SVM	0.78	0.95	0.78	0.77	0.82	0.77	0.69
SGD	0.87	0.95	0.87	0.90	0.72	0.90	0.76
Gradient Boosting	0.92	0.93	0.92	0.92	0.75	0.83	0.78
Decision Tree	0.95	0.95	0.95	0.95	0.85	0.85	0.85
Logistic Regression	0.95	0.96	0.95	0.96	0.82	**0.95**	0.87
Random Forest	**0.97**	**0.97**	**0.97**	**0.97**	**0.96**	0.87	**0.91**

Table 5.1: Average results from 5-fold stratified cross-validation.

Tables 5.2 and 5.3 show, in more detail, the results obtained by the Logistic Regression and Random Forest models, respectively; we also report the results obtained when employing an ablation study, using each group of features in isolation. While we focus on these two models, the same type of tables for the remaining models can be found in Appendix C.

Features (number of features)	Acc.	Genuine News			Fake News		
		P	**R**	**F1**	**P**	**R**	**F1**
N-grams (600)	**0.96**	0.98	**0.98**	**0.98**	0.76	0.80	0.78
Frequencies (64)	0.88	0.98	0.88	0.93	0.39	0.81	0.53
Text Statistics (11)	0.90	**0.99**	0.90	0.94	0.46	0.90	0.61
Readability (12)	0.89	**0.99**	0.89	0.94	0.43	0.86	0.57
All Features (687)	0.95	0.65	0.94	0.77	**0.99**	**0.95**	**0.97**

Table 5.2: Scores of each feature's category fitted in a Logistic Regression model.

Features (number of features)	Acc.	Genuine News			Fake News		
		P	**R**	**F1**	**P**	**R**	**F1**
N-grams (600)	0.96	0.97	0.99	0.98	0.89	0.64	0.75
Frequencies (64)	0.96	0.96	0.99	0.98	0.91	0.55	0.69
Text Statistics (11)	0.96	0.97	0.99	0.98	0.81	0.65	0.72
Readability (12)	0.95	0.96	0.99	0.97	0.77	0.53	0.62
All Features (687)	**0.97**	**0.98**	**1.00**	**0.99**	**0.94**	**0.75**	**0.83**

Table 5.3: Scores of each feature's category fitted in a Random Forest model.

Logistic Regression obtains high accuracy scores for whichever set of features used, especially the model trained with *n*-grams or the one with all the features. The accuracy is even slightly higher on the model trained only with *n*-grams than the all-features model. However, if we examine the f1-score, we can see that although the *n*-grams model performs well in finding genuine news, it shows a poor performance at detecting fake news. This being a fake news detection problem, it makes sense to consider the best model trained with all the features, which achieves a macro-F1 score of 0.87 (as shown in Table 5.1).

All models trained with Random Forest also present very high accuracy scores, even surpassing Logistic Regression. Nevertheless, looking at the F1 score, the best model is the one where all the features are used for training. Although the Random Forest model is almost perfect at identifying genuine news, the same cannot be said about fake news. Comparing the best model of each algorithm, we notice that the f1-score for fake news is lower in the Random Forest model. Nevertheless, the model trained with Random Forest yields the best results, achieving the highest macro-F1 score among all models (as per Table 5.1).

We can also notice that the models trained using frequencies or readability properties alone result in comparatively poorer performance in both learning algorithms. Nevertheless, when the

model is trained with all the feature sets, the overall performance is improved. We can see that the *n*-grams always return the best results for both algorithms among all feature sets. Even though entities were obfuscated, these results may still exhibit some overfitting, as *n*-grams are very reliant on the vocabulary used. The fact that *n*-gram features alone perform very well also suggests that more standard text classification methods, performing well on other tasks, can also perform well in this dataset/task (e.g., it is likely that fine-tuning a pre-trained DL model would produce robust results, without the need for feature engineering).

Results with Logistic Regression also indicate that with the exception of *n*-grams, none of the feature sets can distinguish fake news with a precision higher than 0.5. However, when all of the features are used simultaneously, the model yields an excellent precision score for the fake news class. Additionally, although each feature set is quite good at distinguishing genuine news when all features are used, precision drops significantly.

5.1.1 Feature Analysis

We analyze the main features that each model trained with all the extracted features uses to predict the class label. For Random Forest, we use the *feature_importance_* property[1], while for Logistic Regression we use the *coef_* property[2]. Since each model has its own way of calculating feature importance, we cannot directly compare the values. Furthermore, the two classifiers make predictions in very different ways. Random Forest is a non-linear classifier composed of a multitude of decision trees, whilst Logistic Regression is based on a linear decision boundary and uses a weighted sum of the features to make predictions. This makes comparing feature importance between the models non-trivial. Nevertheless, one can establish the top ten features that each model considers the most important:

Logistic Regression

1. num stopwords
2. num syllables
3. avg words per paragraph
4. avg sents per paragraph
5. 1-gram counts milhão
6. 2-gram counts milhão euro
7. freq !
8. freq [
9. smog score
10. freq <<

Random Forest

1. num syllables
2. num chars
3. num tokens
4. vocabulary richness
5. avg words per paragraph
6. num stop words
7. vocabulary richness without sw
8. avg chars per sentence
9. avg words per sentence
10. 1-gram counts [ORG]

[1] scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier
[2] scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression

The feature analysis suggests noticeable differences in fake news items as compared to genuine news articles. While Random Forest relies mainly on features from the text statistics category, the Logistic Regression model considers that all feature sets are important.

Similar to Random Forest, the Logistic Regression model places more importance on text statistics vs. the other categories. However, Logistic Regression also places some importance on other feature sets: firstly, the *n*-grams "milhão" e "milhão euro" are more frequent in fake news, as mentioned in Section 4.2.2; secondly, the model uses punctuation frequencies, such as "!". This frequency can represent the author's emotions, which are expected to occur more often in fake news, as can be observed in Figure 4.5. The other two frequencies are more related to the style chosen by the authors, which may represent overfitting. Lastly, the model uses a readability score – SMOG. This metric performs a calculation based on the number of sentences and the number of polysyllable words (both metrics higher in genuine news) to grant a final score estimating the years of education needed to understand a text.

In addition to the features related to text statistics, the Random Forest model also uses unigram counts [ORG] and vocabulary richness features. The former means that it attaches importance to the number of entities identified as organizations; the latter measures language diversity, which is unexpectedly higher in fake news, as mentioned in Section 4.2.2.

5.1.2 Testing with Unforeseen Data

In order to further analyze the performance of our trained models on unseen newspapers, we chose four very distinct news sources from the Portulan Clarin corpus: *Correio da Manhã* (60 865 news), *Caras* (6 523 news), *Record* (47 040 news), and *Expresso* (3 796 news). The first is a generic daily newspaper considered by some sources [45] as having sensationalist characteristics. *Caras* is a celebrity news magazine. The *Record* newspaper is focused on sports-related news. The last one, *Expresso*, is also a generic daily newspaper, and it is the one that most closely resembles *Público*'s news, and consequently (in the model's perspective) genuine news.

We tested our best performing models – Random Forest and Logistic Regression – from the previous tasks on the classification of these four news sources. However, we should not forget that this is not an easy task since all these sources are unseen data for the models. Nevertheless, we test and analyze the results presented in Table 5.4.

Source / Model	*Correio da Manhã* (60 865 news)	*Caras* (6 523 news)	*Record* (47 040 news)	*Expresso* (3 796 news)
Logistic Regression	0.52	0.25	0.52	0.83
Random Forest	0.89	0.78	0.91	0.98

Table 5.4: Percentage of news considered genuine.

The Logistic Regression model predicted for each newspaper, respectively 52%, 25%, 52% and 83% of its news as genuine. These results show that the model struggles with news sources

on topics different from the generic news. Although *Expresso* has a good and somewhat expected result, the accuracy of the model with *Record* and the *Correio da Manhã* newspaper is very low. The *Caras* magazine is the most interesting, with a very low accuracy of 25%, which means that the model classifies 75% of the news articles as fake. Conversely, the Random Forest model predicts, for each newspaper, respectively, 89%, 78%, 91% and 98% of the news.

It is interesting to observe that the two models classify the magazine *Caras* very differently. Of the three news outlets with more expressiveness/emotion in writing, and often having shorter text, this magazine is the one that most resembles fake news outlets. What makes it classify differently is most likely the given importance to different features by each model, as described in Section 5.1.1. We believe that in the *Caras* magazine case, the logistic regression model makes a better prediction since the results indicate there are more fake news articles. These results seem coincide with the observations made earlier, where we mention the similarities between fake news and this source. We can also observe that the *Expresso* newspaper shows a higher ratio of genuine news in both models than the rest of the news sources. This score can be explained by the fact that *Expresso* is the newspaper that most closely resembles *Público* news, news that the model has been trained to identify as genuine news. Furthermore, the reported percentages of *Expresso*'s news considered as genuine by our models proves that we are not distinguishing Público news from fake news – the models are truly distinguishing genuine news from fake news.

5.1.3 Multi-class classification

In order to further investigate if the forensic linguistics approach could also produce good results at distinguishing different types of news in addition to fake and genuine news, we did some experiments with multi-class classification. So, instead of just having the two main types of news, we increased to 5 using some of the news sources aforementioned: fake news (FN), *Correio da Manhã* (CM - sensationalist), *Caras* (C - magazine), *Record* (R - sports-related news), and genuine news (GN). We did not use articles from *Expresso* so as to have a range of the most diverse types, and since the news pieces published in *Expresso* fall very close to those published in *Público* (as demonstrated by the results in Section 5.1.2), we decided to exclude this source.

Then, applying the same methodology in the classification process described in Section 4.2.3, we used the five different types of news as classes for our classification model. We only tested with a Random Forest model for these experiments since the Logistic Regression took too much time to finish the training process. The results can be observed in Table 5.5.

Model	Acc.	Weighted Average			Macro Average		
		P	**R**	**F1**	**P**	**R**	**F1**
FN vs CM vs GN	0.87	0.87	0.87	0.87	0.87	0.77	0.81
FN vs C vs GN	**0.95**	**0.95**	**0.95**	**0.95**	**0.92**	0.79	0.85
FN vs R vs GN	0.91	0.91	0.91	0.91	0.90	**0.83**	**0.86**
FN vs CM vs C vs GN	0.86	0.86	0.86	0.85	0.86	0.72	0.77
FN vs C vs R vs GN	0.89	0.89	0.89	0.88	0.88	0.73	0.79
FN vs CM vs R vs GN	0.82	0.82	0.82	0.81	0.81	0.74	0.77
FN vs CM vs C vs R vs GN	0.80	0.80	0.80	0.80	0.81	0.68	0.72

Table 5.5: Average results from Random Forest multi-class experiments with 5-fold stratified cross-validation.

Using only one of the three added sources in addition to fake and genuine news (one of *Caras*, *Record*, or *Correio da Manhã*), the results show a slightly less but close macro average f1-score than those shown in the binary task results. It is worth noting that the news from the three added sources (*Caras*, *Record*, or *Correio da Manhã* – from Portulan Clarin corpus) are all from the whole year of 2016, which may lead to overfitted models. Nevertheless, by looking at the results, we can draw some important insights. According to the results, we can see that the model considers the *Record* newspaper the least challenging to distinguish from genuine and fake news, which makes perfect sense given its sports-related news content.

Additionally, from the three chosen sources, the model struggles the most to distinguish the *Correio da Manhã* newspaper from genuine and fake news, possibly because the main category of this newspaper is the same as *Público* – generic daily news, which could present more room for error. If we look carefully at the confusion matrix (Figure 5.1) in one of the experiments that include the *Correio da Manhã* newspaper, we can see that a significant number of fake news (34%) were predicted as sensationalist news (from *Correio da Manhã*), leaving the fake news class with a low recall. Furthermore, although almost no news articles (1%) from *Correio da Manhã* were considered as fake news, a considerable number of news (17%) were classified as genuine.

These observations are also supported by the results shown in the experiments where we use two more classes in addition to fake and genuine news (e.g., FN vs C vs R vs GN). Furthermore, feature importance does not change that much as compared to the list mentioned in Section 5.1.1, continuing to demonstrate that the Random Forest model gives more importance to the text statistics features to discriminate between each type of news.

5.2 Deep Learning Approach

For running the experiments for this approach, we also use an Intel® Core™ i7-3770K CPU with 3.50GHz (8 cores) with 32Gb of RAM. Furthermore, we also use the Nvidia GeForce GTX 1080 (8Gb of memory) since it performed exceptionally faster than the regular CPU for some deep

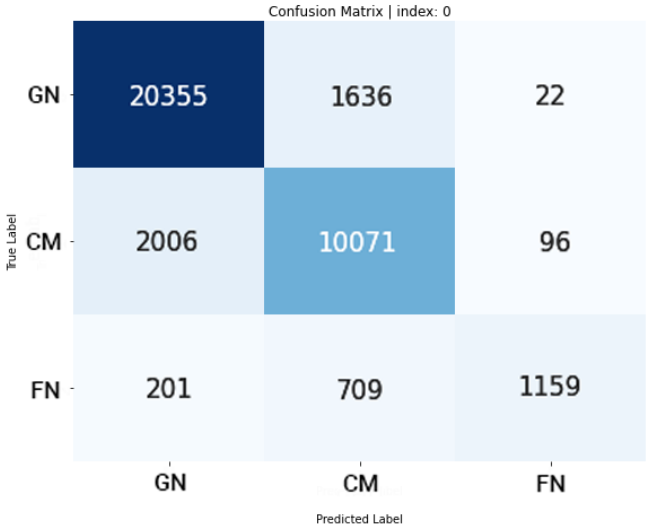


Figure 5.1: Confusion matrix from a fold in the FN vs CM vs GN experiment.

learning related tasks. For the deep learning approach, we present and analyze how state-of-the-art NLP models (e.g., BERT and GPT-2) compare to the models reported in Section 5.1. Unfortunately, one of the shortcomings of deep learning models is how inherently unexplainable they are, and with them, it is difficult to understand what helps the models make a decision. Nevertheless, here we present two experiments and their results, one using word embeddings and the other the perplexity metric. The Random Forest classifier was used for most experiments since it proved to be the most reliable and faster among all models in the first approach. Furthermore, as described in Section 4.2.4, we use a specific stratified train test split (80/20) to make sure that the results are comparable. So we retrained the best performing models from the previous approach and yielded the results presented in Table 5.6.

Model	Acc.	Weighted Average			Macro Average		
		P	**R**	**F1**	**P**	**R**	**F1**
Logistic Regression	0.95	0.96	0.95	0.95	0.79	0.94	0.85
Random Forest	0.98	0.97	0.98	0.97	0.96	0.85	0.90

Table 5.6: Results using all features considered in the feature-based approach.

5.2.1 Word Embeddings Approach

In this first DL approach, we chose a simple strategy to collect the word embeddings for each news article, using two BERT models: Multilingual and Portuguese (Brazilian). Since BERT can only input a sequence of no more than 512 tokens, we truncate the text and only work with the first part of it. BERT then outputs the final hidden state of the first token [CLS] as the vector representation of the whole sequence with a defined length of 768. We use this vector as the features for our ML classification model. The results are presented in Table 5.7.

Model	Acc.	Weighted Average			Macro Average		
		P	**R**	**F1**	**P**	**R**	**F1**
Multiligual cased BERT	**0.97**	0.97	0.97	0.96	0.96	0.80	0.86
Portuguese cased BERT	**0.97**	**0.98**	**0.98**	**0.97**	**0.98**	**0.85**	**0.90**

Table 5.7: Results using BERT word embeddings as features in a Random Forest model.

The results are surprisingly good, considering that only 512 tokens are used for each news article, no time was spent on feature engineering, and the best models produced in the feature-based approach had a very close macro average f1-score to the one produced in this one. One of the alternatives to using the full text would be to split the text into chunks and use the logic of voting ensembles, but this option would be highly computationally intensive. Even though the Portuguese BERT is trained with Brazilian Portuguese, the results are satisfactory. One possible improvement would be to fine-tune BERT embeddings with a set of news from our corpus to see if it would improve these results.

5.2.2 Perplexity Approach

This perplexity approach can be divided into two phases: first, a phase in which we fine-tune a Portuguese GPT-2 model in four different ways: one with 8 000 fake news and the other three with a varying number of genuine news (5 000, 50 000 and 75 000); and a second phase in which we calculate the perplexity and then, applying a heuristic, classify each news article. Next, we present the results in each phase.

5.2.2.1 Fine-tuning

We can empirically evaluate the fine-tuned models if we use the models to solve a text generation task. Some of the best examples are presented in Figure 5.2 and Figure 5.3, where the colored text indicates the input given to the model. If the input's color is blue, then the text was generated with the model fine-tuned with 75 000 *Público*'s news; if it is red, the text was generated using the model fine-tuned with fake news. Other generated text examples using the same input are exhibited in Appendix E.

Although some expressions make no sense (e.g., "três vítimas mortais que ficaram feridas", which translates to "three fatal victims who were injured"), we can discern many unique traits to each type of news that can be used to better distinguish the two. For instance, the use of ellipsis and the word "m*rtais" (obfuscated word) are characteristics that exist predominantly in fake news. Another characteristic that is noticeable in the generated fake news is the expressions and the themes involved in the article – for instance, the use of words related to tragedy (e.g., "vítimas" [victims]) and to social media (e.g., "internet", "partilhar" [share] and "redes sociais" [social media]). On the other hand, we can see more serious themes in the generated text from models trained with genuine news (e.g., "O Governo português" [The Portuguese government] and "estirpe de coronavírus" [coronavirus strain]).

Trump já não vai ser oficialmente anunciado. O Governo português garantiu nesta quinta-feira, em comunicado, a disponibilidade da Alemanha (...).

Mil pessoas foram hospitalizadas devido a esta estirpe de coronavírus. Segundo a Organização Mundial de Saúde, a China elevou para seis mortos e mais de 900 casos confirmados de infecção pelo coronavírus SARS-Cov-2.

Cristina Ferreira continua a ser uma das mais destacadas cantoras de gospel da actualidade, tendo sido uma das actrizes do musical You Need a Star, que ajudou a cantora a mudar os papéis e a assumir o seu real nome.

Figure 5.2: Text generated with GPT-2 Portuguese model fine-tuned with genuine news.

Trump já não está disponível na internet ... mas sabe-se que fica um excelente "spoiler" para ter também um grande número de visualizações, é para ver e partilhar nas redes sociais.

*Mil pessoas foram projetadas para fora da janela dos carros que transportavam a gasolina. "Foi nesse momento que houve três vítimas m*rtais que ficaram feridas", revelou a mesma fonte.*

Cristina Ferreira continua a trabalhar nas suas redes sociais que acaba por tirar o fôlego aos seus fãs. Veja e delicie-se na foto partilhada:

Figure 5.3: Text generated with GPT-2 Portuguese model fine-tuned with fake news.

5.2.2.2 Classifying

To make use of the fine-tuned models, we decided to apply the perplexity metric obtained by a model from a given text to classify a piece of news. Like in the case of BERT, the GPT-2 model has a maximum length of text segments that can be inputted. Therefore, to calculate the perplexity, we needed to truncate the text to 1024 tokens. The distribution of the calculated perplexity is shown in Figure 5.4, the scales are different in each plot to maximize readability.

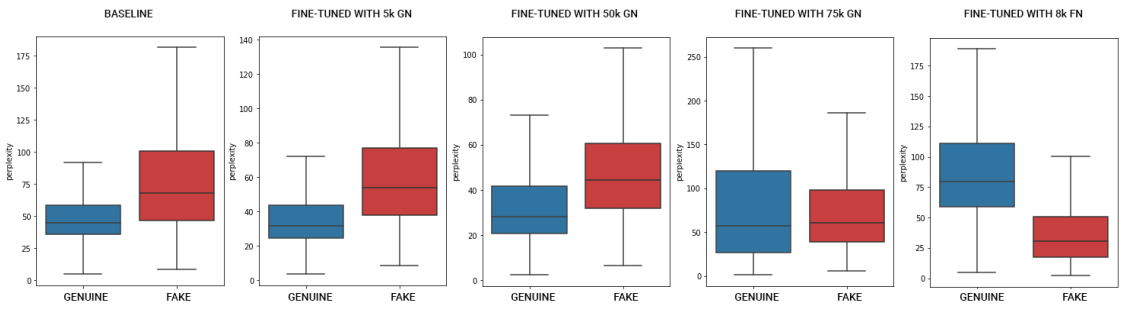


Figure 5.4: Perplexity distribution for each model using the test set.

Figure 5.4 shows that when the GPT-2 is fine-tuned with a specific type of news, the average perplexity for that type of news tends to be less than the other type, i.e., the model tends to be less perplexed by news of the type with which was trained. Furthermore, the model that most

distinguishes the two types of news (most distant averages) is the one fine-tuned with fake news. We noticed that when the model is fine-tuned with more news, this does not translate to more distant averages, as we can see in the model fine-tuned with 75k genuine news.

We tried the threshold approach to classify the two types of news. In order to use this approach, we first test a wide range of possible thresholds and choose the one that maximizes the macro average f1-score. The process of choosing the best threshold is described in more detail in Section 4.2.4.2. Figure 5.5 presents the accuracy and macro average f1-score for each threshold tested.

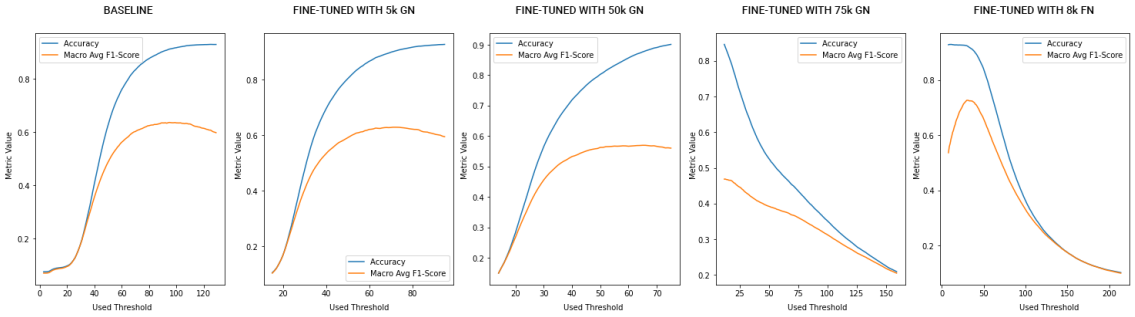


Figure 5.5: Tested thresholds for each model.

By using the threshold that maximizes the macro average f1-score, we can calculate the usual classification metrics. The results with the different fine-tuned models are presented in Table 5.8.

Model	Thld.	Acc.	Weighted Average			Macro Average		
			P	**R**	**F1**	**P**	**R**	**F1**
Baseline (Portuguese GPT-2)	95	0.91	0.90	0.91	0.91	0.66	0.62	0.64
Fine-tuned with 5k GN	72	0.90	0.90	0.90	0.90	0.64	0.62	0.63
Fine-tuned with 50k GN	65	0.88	0.88	0.88	0.88	0.57	0.57	0.57
Fine-tuned with 75k GN	12	0.85	0.85	0.85	0.85	0.47	0.47	0.47
Fine-tuned with 8k FN	30	**0.93**	**0.92**	**0.93**	**0.92**	**0.73**	**0.73**	**0.73**

Table 5.8: Results using a calculated threshold to classify each type of news.

The results show a disappointing performance, which is not even close to the ones obtained in the feature-based approach. The best model is the one fine-tuned with fake news, which was the only model able to improve the results from the baseline (the default GPT-2 model). These results can possibly be explained by the nature of the default model used. The GPT-2 model used is trained with the Portuguese Wikipedia, which includes European Portuguese and Brazilian Portuguese. Some fake news pieces show Brazilian Portuguese characteristics, and this could present some overfitting. Thus, focusing on the perplexity aspect could present an "advantage" to the fake over the genuine news.

Our suspicions mentioned above are also confirmed by the results exhibited in Table 5.9, proving that fine-tuning the model with more news does not mean the model is better at distinguishing

the two types of news. What we know for sure is that fine-tuning with more news makes the model yield lower perplexity. The problem is that perplexity is lower for both genuine and fake news, to the point where there is not much difference between them (e.g., when the model is trained with 75k news).

After doing this experiment, we combine the calculated perplexity from the best model fine-tuned with genuine news (i.e., fine-tuned with 5k GN), from the model fine-tuned with fake news (i.e., fine-tuned with 8k FN), and from the default GPT-2 model (without fine-tuning it) and use them as features in a ML classification model. The best results of each combination are presented in Table 5.9. For the complete list of results, see Appendix D.

Model	Acc.	Weighted Average			Macro Average		
		P	**R**	**F1**	**P**	**R**	**F1**
LR (8k FN + Default)	0.97	0.97	0.97	0.97	0.88	0.90	0.88
RF (5k GN + Default)	0.93	0.91	0.93	0.91	0.75	0.60	0.64
LR (8k FN + 5k GN)	0.99	0.99	0.99	0.99	0.95	0.95	0.95
LR (5k GN + 8k FN + Default)	0.99	0.99	0.99	0.99	0.96	0.96	0.96

Table 5.9: Best results from the different combinations of perplexities calculated by different models.

Using ML models with the calculated perplexity as input proves to be a much better solution than using a simple threshold to distinguish the two types of news, even outperforming the results obtained in the feature-based approach. Moreover, the results use just a part of the text (1024 tokens) considering the truncation needed. This limitation could be mitigated by splitting the text into chunks and using the logic of voting ensembles, just as discussed in the word embeddings approach (section 5.2.1).

5.3 Summary

In this chapter, different techniques were explored to solve the task of fake news detection.

First, we implement a feature-based approach consisting of linguistic and stylistic analysis methods that have been explored in forensic linguistics. The best results reported from this first approach are excellent, achieving an accuracy of 97% and a macro average f1-score of 91%. We also do some experiments with unseen data and feature analysis to understand what makes the models decide between the two types of news.

Secondly, we do some research with deep learning approaches since models like BERT and GPT-2 are considered to be state of the art in many NLP tasks. We use the BERT model to collect the word embeddings and with it train a model in a first DL approach. We also test GPT-2 and the perplexity metric in order to classify fake and genuine news. Although the results are equivalent or slightly better than those obtained with the first approach, our search suggests that the feature-based approach is comparable to state-of-the-art NLP models.

Chapter 6 concludes this research work by summarizing the purpose of this work, mentioning some possible threats to the validity, presenting the contributions of the research, and outlining a future research direction.

Chapter 6

Conclusions

Fake news is news that does not follow journalism principles. Instead, the producers of such news try to mimic the look and feel of real news and have a hidden agenda to disinform the reader. Although the internet is a fantastic tool, some problems are associated with its incredible speed and accessibility. It has never been easier to post anything on the internet, and this freedom can present an opportunity for fake news to spread. This phenomenon is a severe problem in our society, and in the last few years, the topic has become increasingly relevant.

Although fake news detection is an increasingly studied topic of research, few datasets are available online, especially datasets of Portuguese news. To address this issue, we collected and introduced two corpora: one composed of fake news from five different sources, and the other of genuine news published in the *Público* newspaper. Both corpora are from the same time frame and use a silver standard approach. We then performed feature engineering inspired on forensic linguistic analysis – the main contribution of this project.

We address the detection of Portuguese fake news, by proposing a feature-based ML approach that relies on heuristics from the field of forensic linguistics, i.e., using linguistic and stylistic features. Various features were generated and can be divided into four different categories: *n*-grams, relative frequencies, text statistics, and readability properties.

Other experiments are also conducted to evaluate the performance of the first approach – feature-based approach – with unseen newspapers. To do so, we test the created models against news from other sources (part of the Portulan Clarin corpus) and investigate if the forensic linguistics approach can produce good results at distinguishing different types of news other than fake and genuine news.

Moreover, we use BERT and GPT-2 in two different experiments to compare the work from the primary approach to the current state-of-the-art technology. Firstly, using BERT, we employ the word embeddings produced by BERT as features to a classification model. Secondly, using GPT-2, we resort to the perplexity metric to distinguish the two types of news.

To further strengthen the scientific contribution of this dissertation, **one article [41] was written during the dissertation development**.

6.1 Threats to Validity

We believe that there are two main threats in the choices made in this project that could potentially raise some doubts about its validity. One related to the lack of relevant works in automated Portuguese fake news detection, and the other related to the collected and used dataset.

The lack of relevant works in automated Portuguese fake news detection presents a problem of comparability. One possible solution is to compare the results obtained in this project against analyses conducted by humans or even forensic linguistics experts. With this comparison, we would have a better understanding of how good the results are.

Due to the lack of datasets of Portuguese news, as previously mentioned, we decided to collect a corpus. This corpus is a combination of multiple fake news outlets and a single genuine news source, *Público*. One might say that by making the choices we made, we identify the style of *Público* vs. fake newspapers instead of classifying fake news as distinct from genuine news. One possible approach would be to test the same settings and feature engineering with another dataset, possibly against an English annotated dataset, considering that many works for English fake news make use of public datasets. This solution would also solve the first threat since we would have the other works against which to compare the results.

In this exploratory work, we proposed to check if the forensic linguistics approach works and is reliable to detect Portuguese fake news, and that is what we have done with the available tools. Hopefully, this work will also spark new investigation on forensic linguistics for fake news detection and operate as a baseline for future works.

6.2 Results

For the primary approach – feature-based approach – our best model is the Random Forest classifier. It achieves the highest accuracy of 97% and a macro f1-score of 90% when using all the features extracted. Although our work yields better accuracy than those mentioned in the related work (see Chapter 2), comparing the results is a non-trivial task. First of all, the corpora used are different. Secondly, while related work has focused on using annotated datasets, we focus on distinguishing each news source considered to be fake or not (silver standard approach). Thus, we cannot just compare the results.

We further tested the models created in the first approach with news sources not present in the corpus used to train the model. The results were interesting. For example, the Random Forest model considers that 78% of the *Caras*' news articles are genuine, while the Logistic Regression (the second best performing) model only considers that 25% of the news articles are genuine. Other observations suggest that the Expresso newspaper have the highest volume of genuine news when compared to the other news sources, as inferred by the results provided by these two models.

This makes sense since it is the newspaper that most resembles *Público*'s news, which our models were trained to guess as genuine.

Furthermore, to compare the results given by the state-of-the-art technologies, we did two more experiments. The first experiment revealed that BERT, with zero to minimal work in the feature engineering phase, could achieve similar performances to those yielded in the feature-based task. The second experiment presented the best results of the whole work, achieving 96% of macro average f1; it used a classification model whose features are the perplexity attribute generated by multiple models (some fine-tuned). From a forensic linguistics perspective, the big problem with this second approach – Deep Learning – and the experiments done is that we cannot easily understand what made the model make these decisions.

6.3 Research Findings

It is non-trivial to compare the results we found during this dissertation to the ones achieved in the literature review, since the authors of those works study the performance of the proposed models under different datasets. Nevertheless, although this domain remains understudied, **we conclude that a forensic linguistics approach for classifying fake news can be applied successfully**, which answers to the first research question (*Can an approach based on forensic linguistic analysis yield good results at detecting fake news?*). Moreover, to the best of our knowledge, this is the first work that applies this kind of approach to solve the problem of fake news detection in Portuguese texts.

As we have seen in Chapter 5, our two best-performing models consider two different feature sets to classify a text as fake or genuine. Even though some models use various features to discriminate between the two classes, we can answer the second question (*Which are the most relevant features to detect fake news in a forensic linguistics-based system?*) and report that the predominant feature category in the two best models is the text statistics category, which includes features such as the number of paragraphs, sentences, tokens, stopwords, characters, and syllables.

As far as the last research question (*How do systems based on forensic linguistic analysis compare to a modern Deep Learning approach?*) is concerned, we can say that our approach can achieve very close performances both in accuracy and macro average f1-score to the more modern DL-oriented approaches applied in this work, such as BERT and GPT-2. This work suggests that our approach is comparable to state-of-the-art NLP models. Despite some of the DL-related experiments yielding even better results than our primary approach (in the case of the GPT-2 perplexity approach), and despite the DL-oriented approaches having almost zero feature engineering effort, we can confidently report that our approach has its own value. From a forensic linguistics perspective, contrary to the deep learning models, our work enables the study of features that the models consider when distinguishing fake from genuine news.

6.4 Future Research Directions

For future work, we think that further analyzing the robustness of this approach is needed. It is essential to investigate how our model performs with other corpora and possibly compare it against manually annotated datasets. This task is likely a big step towards proving that this approach is not just a coincidence created by the chosen dataset. By using a manually annotated dataset, some of the topics discussed in Section 6.1 considering the possibility that the models are trained to distinguish the style of *Público* vs. other newspapers would be clarified since we would be classifying actual fake and genuine news already identified by humans. Moreover, another approach to clarify the potential of this work would be to test our models' fake news detection efficiency against humans, by measuring the time and accuracy.

Furthermore, a potentially good experiment would be to look at the problem from another perspective and, using the features collected, interpret fake news detection as a regression problem. Instead of looking at news articles as fake or genuine, we could have a score indicating how fake the news article is. With this strategy, we have several options on what could be done; for instance, we could even determine a threshold (after analyzing the score distribution) and use it to get a binary result and possibly compare it to the results presented in this dissertation.

Appendix A

Feature Extraction Details

de	a	o	que	e	é	do
da	em	um	para	com	não	uma
os	no	se	na	por	mais	as
dos	como	mas	ao	ele	das	à
seu	sua	ou	quando	muito	nos	já
eu	também	só	pelo	pela	até	isso
ela	entre	depois	sem	mesmo	aos	seus
quem	nas	me	esse	eles	você	essa
num	nem	suas	meu	às	minha	numa
pelos	elas	qual	nós	lhe	deles	essas
esses	pelas	este	dele	tu	te	vocês
vos	lhes	meus	minhas	teu	tua	teus
tuas	nosso	nossa	nossos	nossas	dela	delas
esta	estes	estas	aquele	aquela	aqueles	aquelas
isto	aquilo	estou	está	estamos	estão	estive
esteve	estivemos	estiveram	estava	estávamos	estavam	estivera
estivéramos	esteja	estejamos	estejam	estivesse	estivéssemos	estivessem
estiver	estivermos	estiverem	hei	há	havemos	hão
houve	houvemos	houveram	houvera	houvéramos	haja	hajamos
hajam	houvesse	houvéssemos	houvessem	houver	houvermos	houverem
houverei	houverá	houveremos	houverão	houveria	houveríamos	houveriam
sou	somos	são	era	éramos	eram	fui
foi	fomos	foram	fora	fôramos	seja	sejamos
sejam	fosse	fôssemos	fossem	for	formos	forem
serei	será	seremos	serão	seria	seríamos	
seriam	tenho	tem	temos	tém	tinha	
tínhamos	tinham	tive	teve	tivemos	tiveram	
tivera	tivéramos	tenha	tenhamos	tenham	tivesse	
tivéssemos	tivessem	tiver	tivermos	tiverem	terei	
terá	teremos	terão	teria	teríamos	teriam	

Table A.1: Full list of considered stop words.

adjective	adposition	adverb	auxiliary
conjunction	coordinating conj.	determiner	interjection
noun	numeral	particle	pronoun
proper noun	subordinating conj.	verb	other

Table A.2: All possible Part-of-Speech Tags that spaCy can recognize used in the feature extraction phase.

Adverb type	**Expressions**
Negation	não, nem, tampouco, nunca, jamais, nada
Affirmation	sim, deveras, decididamente, certamente, realmente, decerto, efetivamente
Interrogation	onde, como, quando, porque
Quantity	muito, pouco, mais, menos, demasiado, quanto, quão, tanto, tão, que, tudo, nada, todo, bastante, quase
Exclusion	apenas, exclusivamente, salvo, senão, somente, simplesmente, só, unicamente
Inclusion	até, inclusivamente, mesmo, também,
Mode	assim, bem, debalde, mal, depressa, devagar, alegremente, simpaticamente, agradavelmente, fortemente, velozmente, carinhosamente
Time	agora, ainda, amanhã, anteontem, antigamente, cedo, então, frequentemente, hoje, já, nunca, ontem, sempre, tarde
Local	abaixo, acima, acolá, adiante, aí, além, algures, ali, aquém, aqui, atrás, cá, defronte, dentro, fora, junto, lá, longe, perto
Connective	porém, contudo, todavia, primeiramente, seguidamente, consequentemente
Doubt	provavelmente, possivelmente, talvez, porventura, acaso, quiçá

Table A.3: All possible adverb types and the associated expressions used in the feature extraction phase.

!	"	#	$	%	&	’	( )
*	+	,	-	.	/	:	;
< >	=	?	@	_	–	[ ]	\
^	`	{ }	|	~	...	« »	€

Table A.4: All possible punctuation marks used in the feature extraction phase.

avg chars per word	num paragraphs	num sentences
num tokens	num stopwords	num chars
freq adjectives	freq adpositions	freq adverbs
freq auxiliarys	freq conjunctions	freq coordinating conj.
freq determiners	freq interjections	freq nouns
freq numerals	freq particles	freq pronouns
freq proper nouns	freq subordinating conj.	freq verbs
freq others	freq !	freq "
freq #	freq $	freq %
freq &	freq ’	freq (
freq)	freq *	freq +
freq ,	freq -	freq .
freq /	freq :	freq ;
freq <	freq =	freq >
freq ?	freq @	freq [
freq \	freq]	freq ^
freq _	freq `	freq {
freq |	freq }	freq ~
freq ...	freq «	freq »
freq –	freq EUR	freq adv negation
freq adv affirmation	freq adv interrogation	freq adv quantity
freq adv exclusion	freq adv inclusion	freq adv mode
freq adv time	freq adv local	freq adv connective
freq adv doubt	flesch score	fleschkincaid score
gunningfog score	smog score	num of syllables
vocab. richness	vocab. richness without sw	% obfuscated words
% long words	% polysybl words	% uppercase words
% stopwords	% misspelled words	avg sentences per paragraph
avg words per paragraph	avg words per sentence	avg chars per sentence

Table A.5: Features generated in the feature extraction phase (excluding the *n*-grams).

Appendix B

Corpus *N*-grams Details

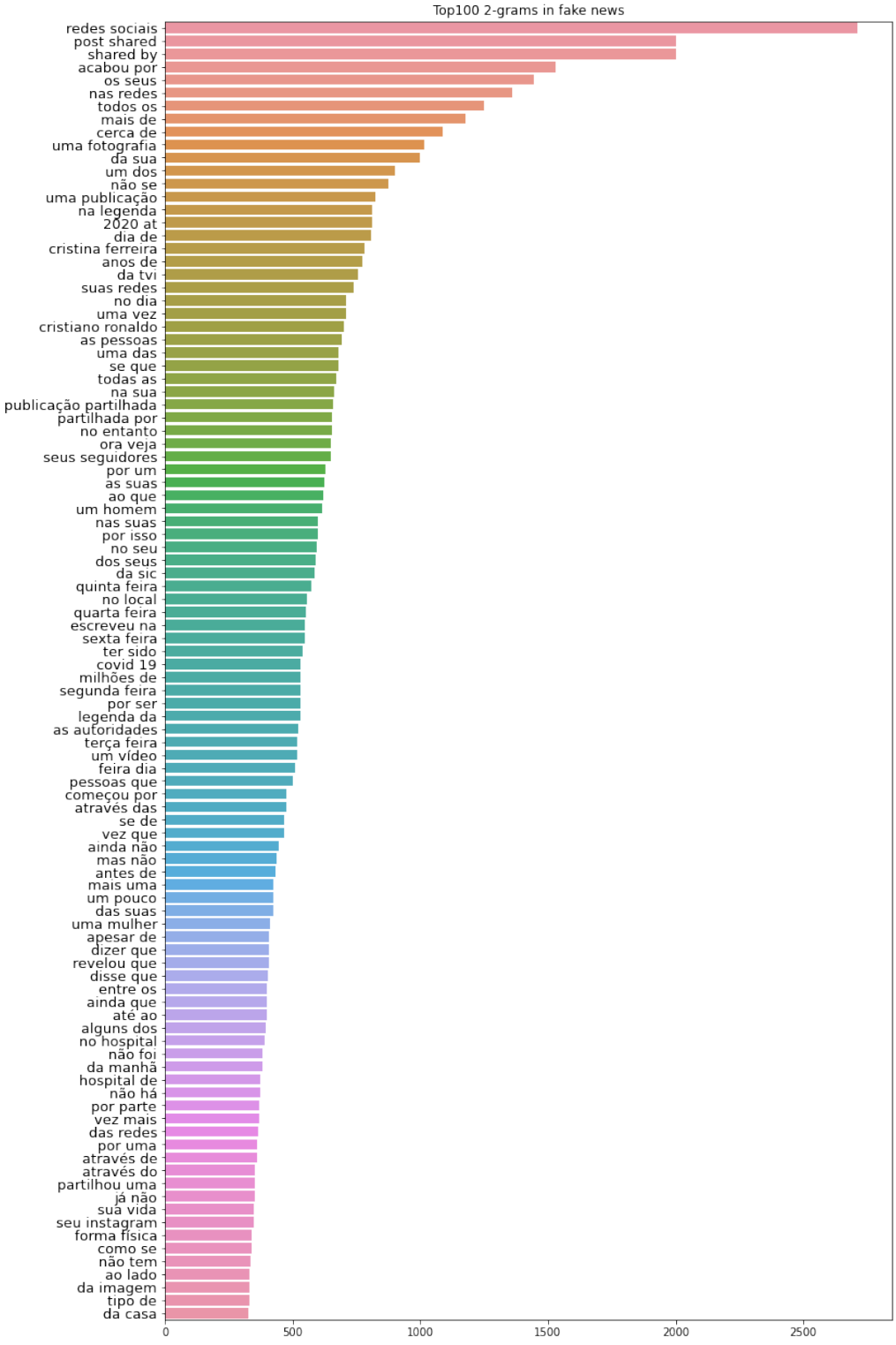


Figure B.1: Top 100 bigrams in fake news.

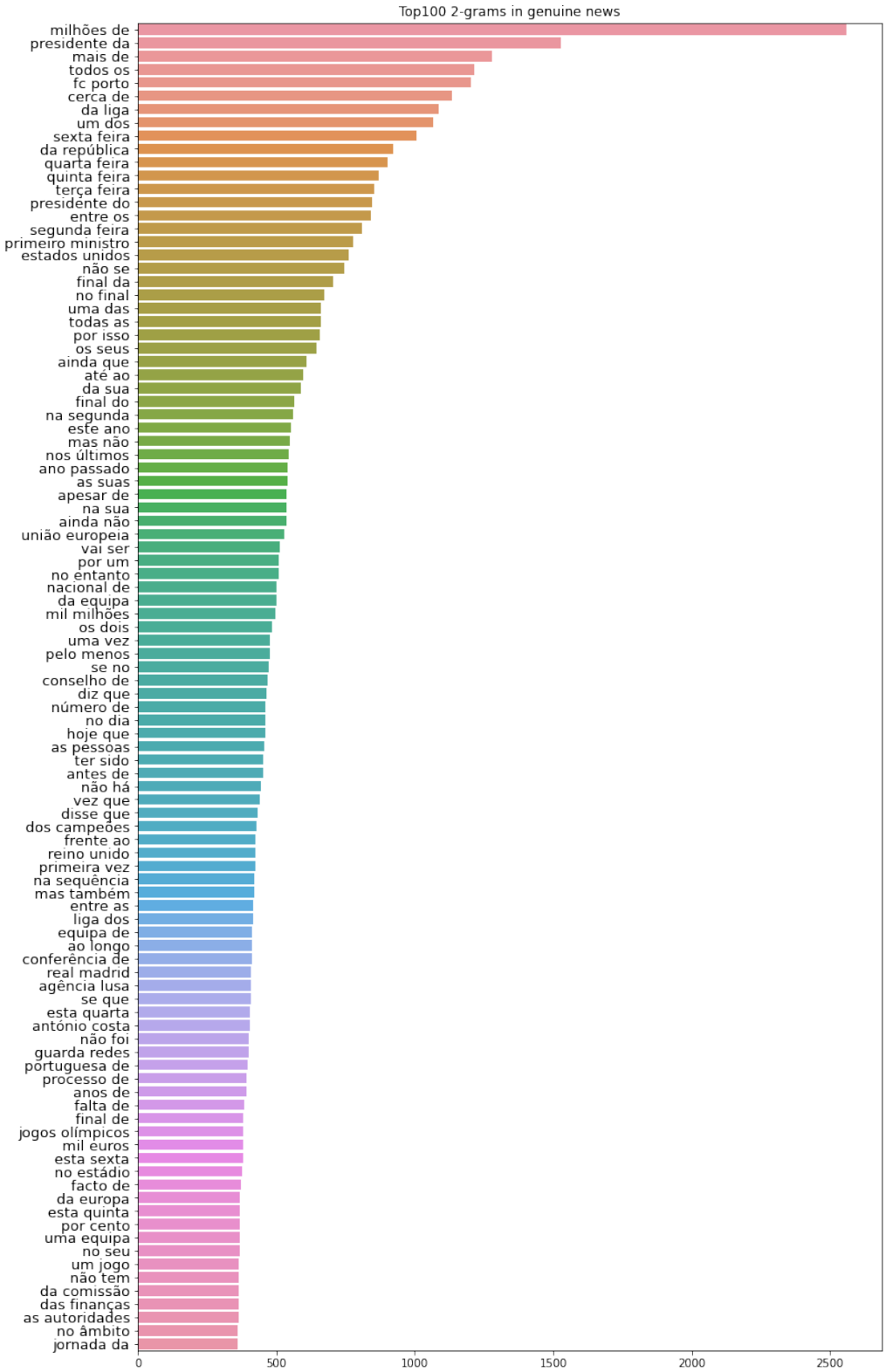


Figure B.2: Top 100 bigrams in genuine news.

Appendix C

Feature-based Task Remaining Results

Features (number of features)	Acc.	Genuine News			Fake News		
		P	**R**	**F1**	**P**	**R**	**F1**
N-grams (600)	**0.94**	**0.99**	0.95	**0.97**	0.62	**0.89**	**0.73**
Frequencies (64)	0.88	0.98	0.89	0.93	0.40	0.81	0.53
Text Statistics (11)	0.81	0.96	0.83	0.85	0.56	0.55	0.43
Readability (12)	0.93	0.94	**0.98**	0.96	**0.73**	0.38	0.46
All Features (687)	0.78	0.98	0.78	0.78	0.66	0.76	0.60

Table C.1: Scores of each feature's category fitted in a LinearSVM model.

Features (number of features)	Acc.	Genuine News			Fake News		
		P	**R**	**F1**	**P**	**R**	**F1**
N-grams (600)	0.94	**0.97**	**0.97**	**0.97**	0.64	0.66	0.65
Frequencies (64)	0.93	0.96	0.96	0.96	0.59	0.57	0.58
Text Statistics (11)	0.94	0.96	**0.97**	**0.97**	0.63	0.61	0.62
Readability (12)	0.92	0.96	0.96	0.96	0.54	0.52	0.53
All Features (687)	**0.95**	**0.97**	**0.97**	**0.97**	**0.73**	**0.72**	**0.73**

Table C.2: Scores of each feature's category fitted in a Decision Tree model.

Features (number of features)	Acc.	Genuine News			Fake News		
		P	**R**	**F1**	**P**	**R**	**F1**
N-grams (600)	**0.88**	**0.99**	**0.88**	**0.93**	0.44	**0.93**	**0.59**
Frequencies (64)	0.86	0.98	0.87	0.92	0.37	0.80	0.50
Text Statistics (11)	0.75	**0.99**	0.74	0.76	0.44	0.88	0.55
Readability (12)	0.79	**0.99**	0.78	0.86	0.31	0.89	0.45
All Features (687)	0.87	**0.99**	0.87	0.92	**0.46**	0.92	**0.59**

Table C.3: Scores of each feature's category fitted in a SGD model.

Features (number of features)	Acc.	Genuine News			Fake News		
		P	**R**	**F1**	**P**	**R**	**F1**
N-grams (600)	**0.96**	**0.97**	**0.99**	**0.98**	**0.82**	**0.69**	**0.75**
Frequencies (64)	0.92	0.94	0.98	0.96	0.58	0.31	0.40
Text Statistics (11)	0.92	**0.97**	0.93	0.95	0.51	0.73	0.60
Readability (12)	0.92	**0.97**	0.94	0.95	0.52	0.68	0.58
All Features (687)	0.92	**0.97**	0.93	0.95	0.51	0.72	0.60

Table C.4: Scores of each feature's category fitted in a GBC model.

Appendix D

Perplexity DL Task Results in More Detail

Model	Acc.	Weighted Average			Macro Average		
		P	**R**	**F1**	**P**	**R**	**F1**
Decision Tree	0.98	0.98	0.98	0.98	0.97	0.89	0.92
Random Forest	0.98	0.98	0.98	0.98	**0.98**	0.90	0.94
Logistic Regression	**0.99**	**0.99**	**0.99**	**0.99**	0.95	**0.95**	**0.95**

Table D.1: Results using the perplexity calculated by two models (8k FN + 5k GN) fine-tuned in different types of news.

Model	Acc.	Weighted Average			Macro Average		
		P	**R**	**F1**	**P**	**R**	**F1**
Decision Tree	0.96	0.96	0.96	0.96	0.92	0.77	0.83
Random Forest	0.96	0.96	0.96	0.96	**0.95**	0.77	0.83
Logistic Regression	**0.97**	**0.97**	**0.97**	**0.97**	0.88	**0.90**	**0.88**

Table D.2: Results using the perplexity calculated by two models (8k FN + Default model) fine-tuned in different types of news.

Model	Acc.	Weighted Average			Macro Average		
		P	**R**	**F1**	**P**	**R**	**F1**
Decision Tree	0.89	0.89	0.89	0.89	0.60	0.60	0.60
Random Forest	**0.93**	**0.91**	**0.93**	**0.91**	**0.75**	0.60	**0.64**
Logistic Regression	0.77	0.90	0.77	0.82	0.57	**0.70**	0.57

Table D.3: Results using the perplexity calculated by two models (5k GN + Default model) fine-tuned in different types of news.

Model	Acc.	Weighted Average			Macro Average		
		P	**R**	**F1**	**P**	**R**	**F1**
Decision Tree	**0.99**	0.98	**0.99**	0.98	0.98	0.91	0.94
Random Forest	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**	0.91	0.95
Logistic Regression	**0.99**	**0.99**	**0.99**	**0.99**	0.96	**0.96**	**0.96**

Table D.4: Results using the perplexity calculated by three models (5k GN + 8k FN + Default model) fine-tuned in different types of news.

Appendix E

Examples of text generated by the fine-tuned models

Trump já não é o patrão. "Se eu fosse um bom patrão, ele, a sua posição ou o seu plano [da CTT], teria saído", disse, sem se referir a Trump.

Trump já não consegue voltar a dirigir-se às redes sociais depois de ter chegado à frente das audiências com números impressionantes — tem 9,8 milhões de seguidores.

Trump já não tinha pensado no desafio e no objectivo da sua vida. Ouviram-nos conversar. A campanha das presidenciais brasileiras de Outubro teve uma explosão.

Trump já não tem o poder de fogo e não tem o poder de liderança e a voz", disse Trump. Na mesma ocasião, o Presidente norte-americano referiu-se ao seu rival democrata, dizendo que o Partido Democrata não fará campanha.

Trump já não sabia nadar. As suas iniciais apontavam para o solo onde "C. jollie" teria ido, de cano à cano. Com a chegada dos primeiros colonos, as margens do lago já se estreitavam.

Trump já não havia sido lançado em setembro de 2014; o próprio jogo foi anunciado em abril de 2014, porém a data ainda era desconhecida."

Trump já não consegue vencer a terceira temporada consecutiva de "Gossip Girl", sendo que a segunda edição foi realizada no dia 9 de Dezembro de 2015, e a quarta foi realizada dia 13 de janeiro de 2016.

Trump já não havia sido usado. Em 15 de Maio de 2015, o director do serviço "Ampel" disse que a equipe encontrou, com êxito, um modelo da Internet e comprou um carro de corrida de Fórmula 1.

Figure E.1: Text generated with input 'Trump já não' with both GPT-2 Portuguese model fine-tuned with genuine news and with fake news (the model was fine-tuned by: red - fake news | blue - genuine news).

Mil pessoas foram retiradas das suas casas e o número de feridos deverá chegar para dez. Há mais de 30 mil pessoas em campos de deslocados localizados na região de Ghouta oriental, que tem sido duramente atingido pela ofensiva militar do regime contra as minorias.

Mil pessoas foram levadas para a rua, com violência. Muitos são idosos. Há relatos de tiros e de tiros de caçadeira nas estradas, prédios ocupados, lojas com comida e outras pessoas.

Mil pessoas foram feridas e outras 35 estão desaparecidas, informou o Governo do estado de Jasudiago, citado pela BBC. "Há danos consideráveis e que não permitem perceber como os próximos dias serão", afirmou ao site Times of India o meteorologista.

Mil pessoas foram registadas em todo o país entre as 23h e as 3h desta terça-feira — sem contar com as habituais visitas nocturnas, em horário a cumprir por causa dos feriados.

Mil pessoas foram (incluindo duas crianças) mortas. As estimativas para vítimas são impossíveis devido a problemas legais enfrentados pela Guarda Civil Metropolitana de Salvador, que interditou as ruas e os canais de televisão.

Mil pessoas foram encontradas nas regiões oeste e norte da costa entre 1596 e 1604. Além disso, há indícios de ocupação das terras de "Maine and Waterford".

Mil pessoas foram mortas e várias igrejas foram danificadas e algumas dezenas de empresas e instituições em situação precária. A violência contra os cristãos parece ter sido um assunto que começou aos chegar em Portugal, sobretudo durante a segunda metade do século XVIII.

Mil pessoas foram mortas durante a madrugada, incluindo nove pessoas mortas após serem atacadas por torcedores da cidade de Paris, e o número de feridos era de 6,8 milhões de pessoas sem vida, segundo o governo francês.

Figure E.2: Text generated with input 'Mil pessoas foram' with both GPT-2 Portuguese model fine-tuned with genuine news and with fake news (the model was fine-tuned by: red - fake news | blue - genuine news).

Cristina Ferreira continua à frente da Sograpе por 6,1 mil milhões de euros, menos 0,2 pontos percentuais do que no quarto trimestre de 2019, e a Amorim apresentou resultados líquidos ligeiramente penalizados pela pandemia da covid-19

Cristina Ferreira continua a afirmar-se nos números, com cinco golos marcados e um expressivo segundo posto da classificação da II Liga, ultrapassando o seu registo de 2009/10, quando fez quatro golos. Se somarmos os três campeonatos nacionais que os separam

Cristina Ferreira continua a precisar de um empate (ou um prolongamento até ao intervalo). Depois de cinco finais perdidas em torneios realizados no Dubai, em Julho, e das duas que a Portugal somou neste domingo, em Doha, as quatro finais perdidas para

Cristina Ferreira continua à espera desta e de outras decisões sobre o fim da proibição, mas garante que a intenção foi concretizada. "Há várias semanas, quando discutimos a recomendação – que se aplicava ao sector das bebidas alcoólicas – a indicação

Cristina Ferreira continua a crescer o número de novos membros que o compõem. A Associação de Municípios com mais de 80 localidades cadastradas é mantida oficialmente. O município abrange o distrito de Santa Maria da Barra, além das cidades de Barra do Piraí.

Cristina Ferreira continua a ser uma figura destacada pela sua grande liderança intelectual, por seu engajamento cívico e por sua grande contribuição em prol da cultura da Região Nordeste do Brasil.

Cristina Ferreira continua, no setor educacional, da responsabilidade pela construção de estradas rurais, da produção agrícola e dos serviços à comunidade, contribuindo para o desenvolvimento social da região.

Cristina Ferreira continua a defender seus direitos de forma mais ampla, afirmando que a liberdade dos moradores da cidade são fundamentais para a afirmação dos valores democráticos da cidade, enquanto sua visão de futuro pode ter vida e sucesso.

Figure E.3: Text generated with input 'Cristina Ferreira continua' with both GPT-2 Portuguese model fine-tuned with genuine news and with fake news (the model was fine-tuned by: red - fake news | blue - genuine news).

Appendix F

EPIA2021 Accepted Paper

The paper [41], written during the dissertation development, was submitted and accepted in the peer-reviewed conference EPIA 2021. It explores the first approach described in this dissertation – feature-based approach. The paper is reproduced next.

Automated Fake News detection using Computational Forensic Linguistics

Ricardo Moura[1], Rui Sousa-Silva[2,3][0000−0002−5249−0617], and Henrique Lopes Cardoso[1,4][0000−0003−1252−7515]

[1] Faculdade de Engenharia, Universidade do Porto, Portugal
[2] Faculdade de Letras, Universidade do Porto, Portugal
[3] Centro de Linguística da Universidade do Porto (CLUP)
[4] Laboratório de Inteligência Artificial e Ciência de Computadores (LIACC)
up201604912@fe.up.pt, rssilva@letras.up.pt, hlc@fe.up.pt

Abstract. Fake news is news-like content that has been produced without following journalism principles. Fake news try to mimic the look and feel of real news to intentionally disinform the reader. This phenomenon can have a strong influence on society, thus being potentially a severe problem. To address this phenomenon, systems to detect fake news have been developed, but most of them build upon fact-checking approaches, which are unfit to detect misinformation when a news piece, rather than completely false, is distorted, exaggerated, or even decontextualized. We aim to detect Portuguese fake news by following a forensic linguistics approach. Contrary to previous approaches, we build upon methods of linguistic and stylistic analysis that have been tried and tested in forensic linguists. After collecting corpora from multiple fake news outlets and from a genuine news source, we formulate the task as a text classification problem and demonstrate the effectiveness of the proposed features when training different classifiers for telling fake from genuine news. Furthermore, we perform an ablation study with subsets of features and find that the proposed feature sets are complementary. The highest results reported are very promising, achieving 97% of accuracy and a macro F1-score of 91%.

Keywords: Fake News Detection · Forensic Linguistics · Natural Language Processing · Text Classification · Disinformation · Misinformation.

1 Introduction

Technology has evolved significantly in recent years, and its development and adoption have become increasingly fast and easy. One of the technologies that came to define and influence the next generations is new computer-mediated communication channels, such as social media, messaging services, and blogs. These channels made it possible for anyone to share anything about any topic at any time, instantly and effortlessly. As a result, people are more connected than ever. Companies are aware of this phenomenon and try to use it for their own advantage, e.g. the media now share news on social media. In fact, studies report

that people are shifting away from traditional news sources to social media and messaging services to find their news [25]. Even though these platforms have many advantages, they raise a serious problem: the so-called fake news. Because those platforms give all users the freedom to share everything they want at any time, fake news can emerge very easily and rapidly spread disinformation.

The fake news phenomenon can be defined in several different ways and be of multiple types, from satire to fabrication [20], and some of them are even permissible (i.e., satire). The definition of fake news has mutated throughout the years and began to be applied under wrong circumstances [23]. In the context of this paper, fake news is news that does not follow the journalism principles of factuality, objectivity, and neutrality [13,3]. Instead, fake news pieces try to mimic the look and feel of real news [24] with the intent to mislead the reader. Here lies the distinction between mis- and disinformation: unlike the latter, the former does not intend to mislead.

Although untruthful news accounts have always existed, their use as a way of manipulation and control has recently gained more attention, due to their fast and immediate propagation through social media, without any kind of curation or filtering. Lay people are attracted to this kind of news because of their alluring headlines (used as *clickbait*) and often give more attention to this kind of news than to truthful accounts [4].

Currently, there are two widely used methods to detect fake news: a manual alternative with human intervention and an automatic alternative with Machine Learning methods [8]. The former places the responsibility to assess the news' veracity and accuracy entirely on humans, who then have to flag it depending on their judgment. However, this is not the best option because it has a limited scalability and humans (frequently non-experts) are not sufficiently skilled to distinguish fake from genuine news. The latter alternative to detect fake news consists of using sophisticated computer systems. However, most existing systems are based on fact-checking methods, which fall short of the desired effectiveness, as these systems still lack the robustness to perform a reliable verification of which information is falsely presented [8]. Additionally, detecting fake news goes beyond identifying false information; fact-checking methods are useful when facts are manipulated, but less so when the truth in the news is distorted, exaggerated, or even decontextualized.

This paper presents a system that, contrary to fact-checking, does not depend on the veracity of the facts. Instead, we focus on how the author communicates and how the news is written. In light of this, we address the fake news phenomenon using an approach based on forensic linguistic analysis, i.e. an analysis that considers linguistic and stylistic methods which have been tried and tested in forensic contexts, e.g. to attribute authorship or detect bias in texts [22]. These include, but are not limited to: text statistics (e.g., average text, paragraph, sentence and word length, and n-gram sequences); spelling; and lexical choices (e.g., Part-of-Speech). We claim that these approaches have a significant potential to also detect fake news.

Using two corpora collected from multiple sources, we conducted a series of experiments to understand what linguistic characteristics are intrinsic of fake

news. Our experiments show promising results with an accuracy of up to 97% and a macro average of F1-score of 90%.

This paper is structured as follows. Section 2 briefly presents previous work on fake news detection using methods similar to the ones applied in this paper. Section 3 introduces the resources used in our experiments, specifically the corpora (Section 3.1) and external resources (Section 3.2). Section 4 describes the process, from extracting the features to building the model. Next, in Section 5, we share, evaluate, and discuss our results. Finally, in Section 6 we draw some conclusions, give a perspective into the project's current stage, and discuss what could be the next steps and future work.

2 Related Work

Fact-checking is the predominant approach to detect fake news. Notwithstanding, there are alternative methods that seek to make a decision based on linguistic patterns present in the text. The reasoning being that, when someone writes a lie or a deceiving text, they strategically write the text in a way to avoid suspicion [12]. However, not all traces and patterns can be hidden, and hence linguistics-based approaches are often employed for detecting lies, despite being somewhat understudied in the literature.

Ahmed et al. (2017) [1] propose fake news detection using only n-gram analysis. The authors reached the best performance when using Term Frequency-Inverse Document Frequency (TF-IDF) as a feature extraction technique and a Linear Support Vector Machine (LSVM) as a classifier, with an accuracy of 92%. This accuracy is better than the results obtained by Horne and Adali (2017) [14] (see below). However, this high accuracy score can represent a Population Bias or Representation Bias [19]: as Cruz et al. (2019) [6] highlight, relying only on n-gram analysis could present a problem because the results of this feature extraction method may vary depending on media content throughout the years.

Perez et al. (2017) [21] made a set of experiments to identify linguistic properties predominating in fake content. The authors constructed two datasets: one was collected via crowd-sourcing covering six news domains; the other was obtained by scraping data from the web, and covers celebrity fake news. They built a fake news detector that achieved the best performance (78% accuracy) using LSVM. The features used were: n-grams encoded as TF-IDF values; count of punctuation characters; psycho-linguistic features, such as summary categories (e.g. analytical thinking or emotional tone), linguistic processes (e.g. function words or pronouns) and psychological processes (e.g. affective processes or social processes); and features related to readability, such as the number of characters, complex words, long words, number of syllables, word types and paragraphs, among other content features.

Differently from works that focus on the main text, Horne and Adali (2017) [14] consider solely news headlines for detecting fake news. The authors build on the assumption that fake news are targeted at audiences that are not likely to read beyond headlines. They extracted different features and arranged them into

three categories: Stylistic Features (e.g. number of stopwords, number of all capital letter words, PoS tagger count on each tag, etc.); complexity features (e.g. readability scores); and psychological features (e.g. number of emotion or informal/swear words). With this set of features extracted from a corpus from 2016 US Election news (retrieved from BuzzFeed) and other scraped news websites related to US politics, the authors have built a LSVM classifier, achieving 71% accuracy.

Overall, these findings show that linguistic-based approaches are understudied. These approaches are, in fact, used but mostly in other contexts and with different goals, such as rumor detection [2], deception detection [18], or hyperpartisanship detection [6]. Such lack of research into fake news detection using approaches other than fact-checking is also evident in Portuguese. Comparing the performance between the works studied is non-trivial, because the authors target different datasets.

3 Resources

In this section, we introduce the corpora used in our experiments, as well as the external resources used to build the classifier models used to detect fake news. This project focuses on detecting fake news written in Portuguese. Although Portuguese is one of the most widely spoken languages [26], it still has limited linguistic resources available when compared to English. Due to this limitation, most tools supporting NLP show sub-optimal performance. Nevertheless, we will use tools that already have features and offer support of Portuguese to train the model.

3.1 Corpora

Given the nonexistence of an annotated dataset distinguishing fake from genuine news, we follow a silver standard approach [11] with automatically annotated data [5] when collecting news items for both classes. By using this approach, each news article is labeled (fake or not) according to the category associated with the website where it is published. URLs of the news, which were collected between November and December 2020 and included in the dataset, are made available[5].

Fake News Corpus
Although there are several online corpora of fake news[6], to the best of our knowledge none is based on Portuguese. We create a corpus by scraping websites that are known to publish fake news contents[7].From those available, we have chosen

[5] drive.google.com/file/d/1jqiMxbcH6H4ozA3zbTnxphriQx1fKi4G/view
[6] https://github.com/sumeetkr/AwesomeFakeNews
[7] a) sabado.pt/portugal/detalhe/be-pede-audicao-da-erc-para-esclarecer-registo-de-sites-de-fake-news
b) dn.pt/edicao-do-dia/11-nov-2018/fake-news-sites-portugueses-com-mais-de-dois-milhoes-de-seguidores–10160885.html

five: *Bombeiros24*, *JornalDiario*, *MagazineLusa*, *NoticiasViriato*, and *SemanarioExtra*. Some scraped news articles were deemed unusable since they were tagged, by the source, as opinion articles, which have a status that differs from regular news. Our fake news corpus contains 10 343 news pieces posted between 2017 and 2020.

***Público* News Corpus**
We build the genuine news corpora by scraping news articles from *Público*, one of the most reputable news outlets in Portugal. Some scraped articles were deemed unusable since the authors categorized them as parody; hence, they should not be considered fake news. Thus, 110 066 news in total were collected from the same period as part of the fake news corpus.

3.2 Natural Language Processing Resources

We explored multiple resources to get the best results for processing the news articles and ended up using a mix between NLTK[8] for the Portuguese stopwords list, the pySpellChecker[9] library for spell checking, and spaCy models for Portuguese[10] for the other tasks (specifically tokenization, part-of-speech tagging, named entity recognition, and lemmatization). We also use Scikit-Learn[11] implementations of the classifiers we have trained and the function CountVectorizer, from the same library to calculate the n-grams.

4 System Description

Our fake news detection approach includes two phases. The first is a feature extraction phase, where we convert the news articles into a feature-based representation. Subsequently, we train several machine learning models using the representations obtained.

4.1 Feature Extraction

The main text of the news articles is converted into a set of linguistic features. These features (described in more detail in Table 1) can be divided into four categories:

***n*-grams:** We calculate the vocabulary composed of all lemmatized tokens in the documents and subsequently extract a set of unigrams, bigrams, and trigrams, encoded as normalized counts and with TF-IDF. In order to avoid the influence of named entities, we adopt an approach that obfuscates them and focuses on an approach used in forensic linguistic analysis. We use spaCy's named-entity recognition to replace classified entities with their respective label – person, organization, and location (e.g. "*Cristiano Ronaldo*" becomes "*[PERSON]*").

[8] www.nltk.org/howto/portuguese_en
[9] www.github.com/barrust/pyspellchecker
[10] www.spacy.io/models/pt
[11] www.scikit-learn.org

Frequencies: We extract a collection of relative frequencies, including the frequency for each punctuation character, the frequency for each Part-of-Speech tag, and the frequency of each type of adverb.

Text Statistics: We also obtain a set of statistical features: the number of paragraphs, sentences, tokens, stopwords, characters and syllables. From these, we also generate some average counts: average number of sentences per paragraph, words per paragraph, words per sentence and characters per word.

Readability: We compute a set of features that measure how easy it is to read a text. These include vocabulary richness (i.e., how diverse the vocabulary used by an author is), readability indices (e.g. Flesch [9], Flesch-Kincaid [15], Gunning Fog [10] and SMOG [17]), and ratios such as the percentage of long words (> 12 characters), obfuscated words [16] (words with numbers or special characters, e.g. "*cr1me*"), misspelled words, and polysyllable words (> 2 syllables).

4.2 Dataset Description

Figure 1 shows the distribution of the features that seem to differ the most between fake and genuine news. Feature values were normalized and outliers were hidden to facilitate understanding.

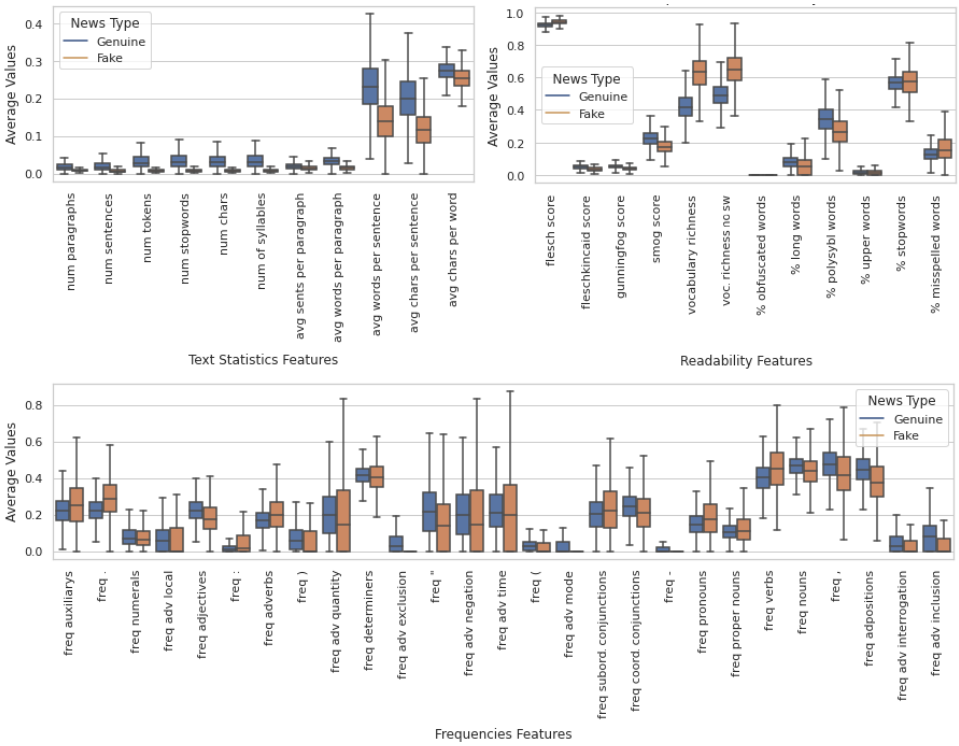


Fig. 1: Distribution values per class for each feature set.

Feature	**Description**
Text Statistics	
num paragraphs	Number of paragraphs.
num sentences	Number of sentences [spacy].
num tokens	Number of tokens.
num stopwords	Number of stopwords [nltk].
num chars	Number of chars.
num of syllables	Number of syllables.
avg sents per para	Average number of sentences per paragraph.
avg words per para	Average number of words per paragraph.
avg words per sents	Average number of tokens per sentence.
avg chars per sents	Average number of characters per sentence.
avg chars per word	Average number of characters per word.
Frequencies	
freq punctuation *	Relative frequency of each punctuation character.
freq PoS tags *	Relative frequency of each PoS tag.
freq type of adverbs *	Relative frequency of each type of the adverb.
Readability	
vocabulary richness *	Measures of vocabulary diversity: ratio between the total number of words and the number of unique words – with or without stopwords.
readability indices *	Measures of text reading/understanding difficulty – *flesch*, *fleschkincaid*, *gunningfog*, and *smog*.
% long words	Fraction of words with 12 or more characters.
% obfuscated words	Fraction of words containing punctuation or numbers.
% misspelled words	Fraction of words with spelling errors.
% uppercase words	Fraction of uppercase words.
% polysybl words	Fraction of of words with three or more syllables.
N-grams	
n-grams *	TF (counts) and TF-IDF of unigram, bigrams, trigrams. In total 600 n-grams.

Table 1: Features used to build the model for Fake News detection. A star ($*$) indicates that the feature is a feature set.

As far as n-gram features are concerned, (lemmatized) word sequences such as "*primeiro ministro*" (prime minister), "*presidente*" (president), "*empresa*" (company), or "*milhão*" (million), are far more frequent in genuine than in fake news. Conversely, words such as "*rede social*" (social media), "*mostrar*" (show), "*mulher*" (woman), or "*vida*" (life) are more frequent in fake news than in genuine news. The dataset also shows that genuine news tend to reference entities more often than fake news, which results in a higher count of entity-related n-grams.

4.3 Classification Process

We conduct several experiments with each feature category and with multiple Machine Learning algorithms, specifically: Logistic Regression (LR), Linear Support Vector Machines (LSVM), Random Forest (RF), Decision Tree (DT), Gradient Descent (SGD), Naive Bayes (NB), and Gradient Boosting Classifier (GBC). We use Scikit-Learn's implementations of these algorithms and resort to the default values of the hyperparameters as defined by the library, only specifying (when possible) the class_weight property to "balanced" to make the algorithms handle both classes with equal importance, and for LR the Lasso penalty (l1).

To better assess the performance of each model, we use 5-fold stratified cross-validation. In each fold, we return the following metrics: Accuracy, Precision, Recall, and F1-score. Although we pay attention to all these metrics, we mainly focus on two. The first is Accuracy, which is the metric consistently presented in the related works section (see Section 2). However, due to the imbalanced nature of our dataset, the second metric we focus on is the macro average F1-score. Furthermore, we collect the feature importance for every model to understand the features that each model deems more important to choose between the fake and genuine news classes.

5 Experimental Results

The results shown in Table 2 are the average performance rates for each model in the 5-fold stratified cross-validation setup. We can observe that Logistic Regression and Random Forest achieve the best results.

Model	Acc.	Weighted Average			Macro Average		
		P	**R**	**F1**	**P**	**R**	**F1**
Naive Bayes	0.77	0.93	0.77	0.82	0.63	0.85	0.64
Linear SVM	0.78	0.95	0.78	0.77	0.82	0.77	0.69
SGD	0.87	0.95	0.87	0.90	0.72	0.90	0.76
Gradient Boosting	0.92	0.93	0.92	0.92	0.75	0.83	0.78
Decision Tree	0.95	0.95	0.95	0.95	0.85	0.85	0.85
Logistic Regression	0.95	0.96	0.95	0.96	0.82	**0.95**	0.87
Random Forest	**0.97**	**0.97**	**0.97**	**0.97**	**0.96**	0.87	**0.91**

Table 2: Average results from 5-fold stratified cross-validation.

Tables 3 and 4 show, in more detail, the results obtained by the Logistic Regression and Random Forest models, respectively; we also report the results obtained when using each group of features individually.

Features (number of features)	Acc.	Genuine News			Fake News		
		P	**R**	**F1**	**P**	**R**	**F1**
N-grams (600)	**0.96**	0.98	**0.98**	**0.98**	0.76	0.80	0.78
Frequencies (64)	0.88	0.98	0.88	0.93	0.39	0.81	0.53
Text Statistics (11)	0.90	**0.99**	0.90	0.94	0.46	0.90	0.61
Readability (12)	0.89	**0.99**	0.89	0.94	0.43	0.86	0.57
All Features (687)	0.95	0.65	0.94	0.77	**0.99**	**0.95**	**0.97**

Table 3: Scores of each feature's category fitted in a Logistic Regression model.

Features (number of features)	Acc.	Genuine News			Fake News		
		P	**R**	**F1**	**P**	**R**	**F1**
N-grams (600)	0.96	0.97	0.99	0.98	0.89	0.64	0.75
Frequencies (64)	0.96	0.96	0.99	0.98	0.91	0.55	0.69
Text Statistics (11)	0.96	0.97	0.99	0.98	0.81	0.65	0.72
Readability (12)	0.95	0.96	0.99	0.97	0.77	0.53	0.62
All Features (687)	**0.97**	**0.98**	**1.00**	**0.99**	**0.94**	**0.75**	**0.83**

Table 4: Scores of each feature's category fitted in a Random Forest model.

Logistic Regression obtains high accuracy scores regardless of the set of features used, especially the model trained with n-grams or the one trained with all the features. The accuracy is even slightly higher when the model trained only with n-grams is used, compared to the all-features model. However, if we examine the F1-score, we can see that although the n-grams model performs well in finding genuine news, it shows a poor performance when detecting fake

news. Since this a fake news detection problem, it makes sense to consider the best model trained with all the features, which achieves a macro-F1 score of 0.87 (as shown in Table 2).

The models trained with Random Forest also present very high accuracy scores, even outperforming Logistic Regression. Nevertheless, we will use the F1-score once more. The best model, in this case, is the one where all the features are used for training. Although the Random Forest model is almost perfect at identifying genuine news, the same cannot be said about fake news. Comparing the best model of each algorithm, we notice that the F1-score for fake news is lower in the Random Forest model. Nevertheless, the model trained with Random Forest yields the best results, achieving the highest macro-F1 score among all models (as per Table 2).

In both learning algorithms, we can also notice that the models trained using frequencies or readability properties alone result in comparatively poorer performance. Nevertheless, when combining with the remaining feature sets, the overall performance is improved. Among all feature sets, we can see that the n-grams always return the best results for both algorithms. Even though entities were obfuscated, these results may still exhibit some overfitting, as n-grams are highly reliant on the vocabulary used.

Results with Logistic Regression also indicate that with the exception of n-grams, none of the feature sets can distinguish fake news with a precision higher than 0.5. However, when all of the features are used simultaneously, the model yields an excellent precision score for the fake news class. Additionally, although each feature set performs rather well at distinguishing genuine news when all features are used, precision drops significantly.

5.1 Feature Analysis

We analyze the main features used by each model to predict the class label. For Random Forest, we use the *feature_importance_* property[12], while for Logistic Regression we use the *coef_* property[13]. Since each model has its own way of calculating feature importance, we cannot directly compare the values. Furthermore, the two classifiers make predictions in very different ways. Random Forest is a non-linear classifier composed of a multitude of decision trees, whilst Logistic Regression is based on a linear decision boundary and uses a weighted sum of the features to make predictions. This makes comparing feature importance between the models non-trivial. Nevertheless, what we can do is compare which are the top ten features each model considers the most important:

[12] scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier
[13] scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression

Logistic Regression	**Random Forest**
1. num stopwords	1. num syllables
2. num syllables	2. num chars
3. avg words per paragraph	3. num tokens
4. avg sents per paragraph	4. vocabulary richness
5. 1-gram counts 'milhão'	5. avg words per paragraph
6. 2-gram counts 'milhão euro'	6. num stop words
7. freq !	7. vocabulary richness without sw
8. freq [	8. avg chars per sentence
9. smog score	9. avg words per sentence
10. freq <<	10. 1-gram counts [ORG]

The feature analysis suggests noticeable differences in fake news articles as compared to genuine news. While Random Forest relies mainly on features from the text statistics category, the Logistic Regression model considers that all feature sets are important.

Similar to Random Forest, the Logistic Regression model places more importance on text statistics, when compared to the other categories. However, Logistic Regression also places some importance on other feature sets: first, the n-grams "milhão" and "milhão euro", which are more frequent in fake news, as mentioned in section 4.2. Next, the model uses punctuation frequencies, such as "!". This frequency can represent the author's emotions, which are expected to occur more often in fake news. The other two frequencies are more related to the style chosen by the authors, which may represent overfitting. Lastly, the model uses a readability score – SMOG. This metric performs a calculation based on the number of sentences and the number of polysyllable words (both metrics are higher in genuine news) to grant a final score estimating the years of education needed to understand a text.

In addition to the features related to text statistics, the Random Forest model also uses unigram counts [ORG] and vocabulary richness features. The former means that it gives importance to the number of entities identified as organizations. The latter measures language diversity, which is unexpectedly higher in fake news, as mentioned in section 4.2.

6 Conclusions

Fake news is news that does not follow the principles of journalism. Instead, the authors of such news try to mimic the look and feel of real news, and have a hidden agenda to disinform the reader. This phenomenon is a severe problem in our society, and the topic has become increasingly relevant in recent years.

For this paper, we collected a corpus of fake news and a corpus of genuine news from the same time frame using a silver standard approach. We then performed feature engineering inspired on approaches used by forensic linguistic analyses.

Although this remains understudied, we conclude that a forensic linguistics-grounded approach for classifying fake news can be applied with great success.

To the best of our knowledge, this is the first work that applies this kind of approach to solve the problem of fake news detection to Portuguese texts.

For future work, we intend to further analyze the robustness of this approach. To do so, we will investigate how our model performs on other corpora and possibly with manually annotated datasets. Furthermore, we will consider exploring the problem in a multi-class formulation exploring different text genres (e.g. fake, genuine, sensationalist news, and so on). We also believe that using neural language models, such as BERT [7], can be a promising direction, and is thus worth exploring.

Acknowledgments

This research is supported by project DARGMINTS (POCI/01/0145/FEDER/031460), CLUP (UIDB/00022/2020), and LIACC (FCT/UID/CEC/0027/2020), funded by Fundação para a Ciência e a Tecnologia (FCT).

References

1. Ahmed, H., Traore, I., Saad, S.: Detection of online fake news using n-gram analysis and machine learning techniques. In: International conference on intelligent, secure, and dependable systems in distributed and cloud environments. Springer (2017)
2. Alkhodair, S.A., Ding, S.H., Fung, B.C., Liu, J.: Detecting breaking news rumors of emerging topics in social media. Information Processing & Management (2020)
3. Bender, J., Davenport, L., Fedler, F., Drager, M.: Reporting for the Media. Oxford University Press (2012)
4. Browne, R.: 'junk news' gets massive engagement on facebook ahead of eu elections, study finds. CNBC (2019), `https://www.cnbc.com/2019/05/21/junk-news-gets-higher-engagement-on-facebook-ahead-of-eu-elections.html`, accessed: 19-04-2021
5. Chowdhury, M.F.M., Lavelli, A.: Assessing the practical usability of an automatically annotated corpus. In: Proceedings of the 5th Linguistic Annotation Workshop. pp. 101–109. Association for Computational Linguistics, Portland, Oregon, USA (Jun 2011), `https://www.aclweb.org/anthology/W11-0412`
6. Cruz, A., Rocha, G., Sousa-Silva, R., Lopes Cardoso, H.: Team fernando-pessa at SemEval-2019 task 4: Back to basics in hyperpartisan news detection. In: Proceedings of the 13th International Workshop on Semantic Evaluation. pp. 999–1003. Association for Computational Linguistics, Minneapolis, Minnesota, USA (Jun 2019). https://doi.org/10.18653/v1/S19-2173
7. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding (2019)
8. Álvaro Figueira, Oliveira, L.: The current state of fake news: challenges and opportunities. Procedia Computer Science (2017). https://doi.org/10.1016/j.procs.2017.11.106
9. Flesch, R.: A new readability yardstick. Journal of applied psychology (1948)
10. Gunning, R.: The Technique of Clear Writing. McGraw-Hill (1952)
11. Hahn, U., Tomanek, K., Beisswanger, E., Faessler, E.: A proposal for a configurable silver standard. In: Proceedings of the Fourth Linguistic Annotation Workshop. pp. 235–242. Association for Computational Linguistics, Uppsala, Sweden (Jul 2010), `https://www.aclweb.org/anthology/W10-1838`

12. Hancock, J.T., Curry, L.E., Goorha, S., Woodworth, M.: On lying and being lied to: A linguistic analysis of deception in computer-mediated communication. Discourse Processes **45** (12 2007). https://doi.org/10.1080/01638530701739181
13. Harrower, T.: Inside Reporting: A Practical Guide to the Craft of Journalism. McGraw-Hill Companies,Incorporated (2007)
14. Horne, B.D., Adali, S.: This just in: Fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news (2017)
15. Kincaid, J.P., Aagard, J.A., O'Hara, J.W.: Development and test of a computer readability editing system (cres). Tech. rep., TRAINING ANALYSIS AND EVALUATION GROUP (NAVY) ORLANDO FL (1980)
16. Laboreiro, G., Oliveira, E.: What we can learn from looking at profanity. pp. 108–113 (10 2014). https://doi.org/10.1007/978-3-319-09761-9_11
17. Laughlin, G.H.M.: Smog grading-a new readability formula. Journal of Reading **12**(8), 639–646 (1969), http://www.jstor.org/stable/40011226
18. Litvinova, O., Seredin, P., Litvinova, T., Lyell, J.: Deception detection in russian texts. In: Proceedings of the Student Research Workshop at the 15th Conference of the European Chapter of the Association for Computational Linguistics (2017)
19. Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., Galstyan, A.: A survey on bias and fairness in machine learning. arXiv (8 2019), http://arxiv.org/abs/1908.09635
20. Mourão, R.R., Robertson, C.T.: Fake news as discursive integration: An analysis of sites that publish false, misleading, hyperpartisan and sensational information. Journalism Studies **20**(14), 2077–2095 (2019). https://doi.org/10.1080/1461670X.2019.1566871
21. Pérez-Rosas, V., Kleinberg, B., Lefevre, A., Mihalcea, R.: Automatic detection of fake news. arXiv preprint arXiv:1708.07104 (2017)
22. Sousa-Silva, R.: Computational forensic linguistics: an overview of computational applications in forensic contexts. Language and Law/Linguagem e Direito **5**(2), 118–143 (2019)
23. Sullivan, M.: What it really means when trump calls a story 'fake news'. https://www.washingtonpost.com/lifestyle/media/what-it-really-means-when-trump-calls-a-story-fake-news/2020/04/13/56fbe2c0-7d8c-11ea-9040-68981f488eed_story.html (2020), accessed: 20-04-2021
24. Tandoc, E., Lim, Z., Ling, R.: Defining "fake news": A typology of scholarly definitions. Digital Journalism **6** (08 2017). https://doi.org/10.1080/21670811.2017.1360143
25. Vorhaus, M.: People increasingly turn to social media for news. https://www.forbes.com/sites/mikevorhaus/2020/06/24/people-increasingly-turn-to-social-media-for-news/ (2020), accessed: 05-04-2021
26. Weber, G.: Top languages. The World's **10** (2008)

References

[1] Hadeer Ahmed, Issa Traore, and Sherif Saad. Detection of online fake news using n-gram analysis and machine learning techniques. In *International conference on intelligent, secure, and dependable systems in distributed and cloud environments*. Springer, 2017.

[2] Sarah A Alkhodair, Steven HH Ding, Benjamin CM Fung, and Junqiang Liu. Detecting breaking news rumors of emerging topics in social media. *Information Processing & Management*, 2020.

[3] Supanya Aphiwongsophon and Prabhas Chongstitvatana. Detecting fake news with machine learning method. In *2018 15th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*. Institute of Electrical and Electronics Engineers Inc., 1 2019.

[4] Daniel Avelar. Whatsapp fake news during brazil election 'favoured bolsonaro'. *The Guardian*, 2016. Accessed: 14-11-2020.

[5] Josh Barua. Word embeddings versus bag-of-words: The curious case of recommender systems. https://medium.com/swlh/word-embeddings-versus-bag-of-words-the-curious-case-of-recommender-systems-6ac1604d4424, 8 2020. Accessed: 05-01-2021.

[6] J.R. Bender, L. Davenport, F. Fedler, and M.W. Drager. *Reporting for the Media*. Oxford University Press, 2012.

[7] Aaron Blake. A new study suggests fake news might have won donald trump the 2016 election. *The Washington Post*, 2018. Accessed: 14-11-2020.

[8] Charles F. Bond and Bella M. DePaulo. Accuracy of deception judgments. *Personality and Social Psychology Review*, 10, 8 2006.

[9] Ryan Browne. 'junk news' gets massive engagement on facebook ahead of eu elections, study finds. *CNBC*, 2019. Accessed: 19-04-2021.

[10] Jay Budzik. Many heads are better than one: The case for ensemble learning. https://www.kdnuggets.com/2019/09/ensemble-learning.html, 2019. Accessed: 20-01-2021.

[11] Andriy Burkov. *The Hundred-Page Machine Learning Book*. Kindle Direct Publishing, 1 edition, 2019.

[12] Md. Faisal Mahbub Chowdhury and Alberto Lavelli. Assessing the practical usability of an automatically annotated corpus. In *Proceedings of the 5th Linguistic Annotation Workshop*, pages 101–109, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[13] PORTULAN CLARIN. Portulan clarin. https://portulanclarin.net/. Accessed: 27-12-2020.

[14] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Mach. Learn.*, 20(3):273–297, September 1995.

[15] André Cruz, Gil Rocha, Rui Sousa-Silva, and Henrique Lopes Cardoso. Team fernandopessa at SemEval-2019 task 4: Back to basics in hyperpartisan news detection. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 999–1003, Minneapolis, Minnesota, USA, June 2019. Association for Computational Linguistics.

[16] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[17] Song Feng, Ritwik Banerjee, and Yejin Choi. Syntactic stylometry for deception detection. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 171–175, Jeju Island, Korea, July 2012. Association for Computational Linguistics.

[18] Rudolph Flesch. A new readability yardstick. *Journal of applied psychology*, 1948.

[19] World Wide Web Foundation. History of the web. https://webfoundation.org/about/vision/history-of-the-web/, 2020.

[20] Aurlien Gron. *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, Inc., 1st edition, 2017.

[21] Stanford NLP Group. Stanza models. https://stanfordnlp.github.io/stanza/available_models. Accessed: 04-01-2021.

[22] R. Gunning. *The Technique of Clear Writing*. McGraw-Hill, 1952.

[23] Udo Hahn, Katrin Tomanek, Elena Beisswanger, and Erik Faessler. A proposal for a configurable silver standard. In *Proceedings of the Fourth Linguistic Annotation Workshop*, pages 235–242, Uppsala, Sweden, July 2010. Association for Computational Linguistics.

[24] Jeffrey T. Hancock, Lauren E. Curry, Saurabh Goorha, and Michael Woodworth. On lying and being lied to: A linguistic analysis of deception in computer-mediated communication. *Discourse Processes*, 45, 12 2007.

[25] T. Harrower. *Inside Reporting: A Practical Guide to the Craft of Journalism*. McGraw-Hill Companies,Incorporated, 2007.

[26] Matthew Honnibal. Spacy models. https://spacy.io/models. Accessed: 04-01-2021.

[27] Benjamin D. Horne and Sibel Adali. This just in: Fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news, 2017. Accessed: 10-01-2021.

[28] D Jurafsky, J H Martin, P Norvig, and S Russell. *Speech and Language Processing*. Pearson Education, 2014.

[29] J Peter Kincaid, James A Aagard, and John W O'Hara. Development and test of a computer readability editing system (cres). Technical report, TRAINING ANALYSIS AND EVALUATION GROUP (NAVY) ORLANDO FL, 1980.

[30] Gustavo Laboreiro and Eugénio Oliveira. What we can learn from looking at profanity. In Jorge Baptista, Nuno Mamede, Sara Candeias, Ivandré Paraboni, Thiago A. S. Pardo, and Maria das Graças Volpe Nunes, editors, *Computational Processing of the Portuguese Language*, pages 108–113, Cham, 2014. Springer International Publishing.

[31] G. Harry Mc Laughlin. Smog grading-a new readability formula. *Journal of Reading*, 12(8):639–646, 1969.

[32] Janna Lipenkova. Major trends in nlp: a review of 20 years of acl research. https://towardsdatascience.com/major-trends-in-nlp-a-review-of-20-years-of-acl-research-56f5520d473, 7 2019. Accessed: 02-01-2021.

[33] Olga Litvinova, Pavel Seredin, Tatiana Litvinova, and John Lyell. Deception detection in russian texts. In *Proceedings of the Student Research Workshop at the 15th Conference of the European Chapter of the Association for Computational Linguistics*, 2017.

[34] Agência Lusa. Feup news corpus – portulan clarin. https://hdl.handle.net/21.11129/0000-000D-F8C2-0. Accessed: 27-12-2020.

[35] Alexandre Magueresse, Vincent Carles, and Evan Heetderks. Low-resource languages: A review of past work and future challenges, 2020. Accessed: 18-01-2021.

[36] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *arXiv*, 8 2019.

[37] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013. Accessed: 12-01-2021.

[38] Aditya Mishra. Metrics to evaluate your machine learning algorithm. https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234, 2 2018. Accessed: 14-01-2021.

[39] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. The MIT Press, 2nd edition, 2018.

[40] Emanuel Monteiro. Hotéis de ronaldo vão ser transformados em hospitais para receber doentes com coronavírus? *Poligrafo*, 3 2020.

[41] Ricardo Moura, Rui Sousa-Silva, and Henrique Lopes Cardoso. Automated fake news detection using computational forensic linguistics. In Goreti Marreiros, Francisco Melo, Nuno Lau, Henrique Lopes Cardoso, and Luís Paulo Reis, editors, *EPIA2021 - 20th EPIA Conference on Artificial Intelligence*, volume Lecture Notes in Artificial Intelligence (LNAI). Springer, 2021.

[42] Rachel R. Mourão and Craig T. Robertson. Fake news as discursive integration: An analysis of sites that publish false, misleading, hyperpartisan and sensational information. *Journalism Studies*, 20(14):2077–2095, 2019.

[43] NLX-Group. Language tools and applications. http://nlx.di.fc.ul.pt/tools.html. Accessed: 11-02-2021.

[44] Hannah Jane Parkinson. Click and elect: how fake news helped donald trump win a real election | us elections 2016. *The Guardian*, 2016. Accessed: 14-11-2020.

[45] João Pedro Gonçalves Mendes Peixoto. *The CM effect: ongoing changes in portuguese journalism?* PhD thesis, University of Minho, 2017.

[46] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics.

[47] Verónica Pérez-Rosas, Bennett Kleinberg, Alexandra Lefevre, and Rada Mihalcea. Automatic detection of fake news. *arXiv preprint arXiv:1708.07104*, 2017.

[48] Kashyap Popat, Subhabrata Mukherjee, Jannik Strötgen, and Gerhard Weikum. Where the truth lies: Explaining the credibility of emerging claims on the web and social media. In *Proceedings of the 26th International Conference on World Wide Web Companion*, 2017.

[49] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. Technical report, OpenAI, 2019.

[50] Google Research. Bert. https://github.com/google-research/bert, 2018. Accessed: 19-01-2021.

[51] Prateek Sawhney. Introduction to stemming and lemmatization (nlp). https://medium.com/swlh/introduction-to-stemming-vs-lemmatization-nlp-8c69eb43ecfe. Accessed: 07-01-2021.

[52] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. Fake news detection on social media: A data mining perspective. *ACM SIGKDD explorations newsletter*, 19(1):22–36, 2017.

[53] Jane B. Singer. Fact-checkers as entrepreneurs. *Journalism Practice*, 12(8):1070–1080, 2018.

[54] Rui Sousa-Silva. Computational forensic linguistics: an overview of computational applications in forensic contexts. *Language and Law/Linguagem e Direito*, 5(2):118–143, 2019.

[55] Margaret Sullivan. What it really means when trump calls a story 'fake news'. https://www.washingtonpost.com/lifestyle/media/what-it-really-means-when-trump-calls-a-story-fake-news/2020/04/13/56fbe2c0-7d8c-11ea-9040-68981f488eed_story.html, 2020. Accessed: 20-04-2021.

[56] Edson Tandoc, Zheng Lim, and Rich Ling. Defining "fake news": A typology of scholarly definitions. *Digital Journalism*, 6, 08 2017.

[57] Reality Check team. Us election 2020: Fact-checking trump team's main fraud claims. https://www.bbc.com/news/election-us-2020-55016029, 11 2020. Accessed: 01-12-2020.

[58] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017.

[59] Mike Vorhaus. People increasingly turn to social media for news. https://www.forbes.com/sites/mikevorhaus/2020/06/24/people-increasingly-turn-to-social-media-for-news/, 2020. Accessed: 25-12-2020.

[60] George Weber. Top languages. *The World's*, 10, 2008.

[61] Fan Yang, Arjun Mukherjee, and Eduard Dragut. Satirical news detection and analysis using attention mechanism and linguistic features. *arXiv preprint arXiv:1709.01189*, 2017.

[62] Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola. *Dive into Deep Learning*. 2020. https://d2l.ai.

[63] Xinyi Zhou and Reza Zafarani. A survey of fake news: Fundamental theories, detection methods, and opportunities. *ACM Computing Surveys*, 53, 12 2018.

[64] Álvaro Figueira and Luciana Oliveira. The current state of fake news: challenges and opportunities. *Procedia Computer Science*, 121:817–825, 2017. CENTERIS 2017 - International Conference on ENTERprise Information Systems / ProjMAN 2017 - International Conference on Project MANagement / HCist 2017 - International Conference on Health and Social Care Information Systems and Technologies, CENTERIS/ProjMAN/HCist 2017.