FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

TrollBus, An Empirical Study Of Features For Troll Detection

Tiago Lacerda

Mestrado em Engenharia de Software

Supervisor: Prof. Carlos Soares

Co-Supervisor: Prof. Henrique Lopes Cardoso

July 28, 2020

TrollBus, An Empirical Study Of Features For Troll Detection

Tiago Lacerda

Mestrado em Engenharia de Software

July 28, 2020

Abstract

In today's social network context, the discussion of politics online has become a normal event. Users from all sides of the political spectrum are able to express their opinions freely and discuss their views in various social networks, including Twitter. From 2016 onward, a group of users whose objective is to polarize discussions and sow discord began to gain notoriety in this social network. These accounts are known as Trolls, and they have been linked to several events in recent history such as the influencing of elections and the organizing of violent protests. Since their discovery, several approaches have been developed to detect these accounts using machine learning techniques.

Existing approaches have used different types of features. The goal of this work is to compare those different sets of features. To do so, an empirical study was performed, which adapts these features to the Portuguese Twitter community. The necessary data was collected through SocialBus, a tool for the collection, processing and storage of data from social networks, namely Twitter. The set of accounts used to collect the data were obtained from Portuguese political journalists and the labelling of trolls was performed with a strict set of behavioural rules, aided by a scoring function.

A new module for SocialBus was developed, called Trollbus, which performs troll detection in real time. A public dataset was also released.

The features of the best model obtained combine an account's profile metadata with the superficial aspects present in its text. The most important feature set noted to be the numerical aspects of the text, with the most important feature revealing to be the presence of political insults.

Keywords: Twitter, Troll, Political Troll, Troll Detection, Troll Classification, Socialbus, Trollbus, Portuguese Twitter

Resumo

No atual contexto de redes sociais, a discussão política tornou-se um evento normal. Utilizadores de todos os segmentos do espetro político têm a possibilidade de expressar as suas opiniões livremente e discutir as suas visões em várias redes sociais, incluindo o Twitter. Desde 2016, um grupo de utilizadores cujo objetivo é polarizar discussões e semear a discórdia começou a ganhar notoriedade nesta rede social. Estas contas são conhecidas como Trolls, e têm sido ligadas a vários eventos na história recente, tais como a interferência em eleições e a organização de manifestações violentas. Desde a sua descoberta, vários trabalhos de investigação têm sido realizados de modo a detetar estas contas através de machine learning.

As abordagens existentes usaram tipos diferentes de atributos. O objetivo deste trabalho é comparar esses grupos de atributos. Para tal, um estudo empírico foi realizado, no qual estes atributos são adaptados à comunidade portuguesa do Twitter. O objetivo deste trabalho foi de analisar as múltiplas abordagens realizadas para a deteção de trolls, com uma descrição das suas features e a sua comparação, quer individualmente quer em grupo. Para tal, um estudo empírico foi realizado, em que estas features são adaptadas à comunidade portuguesa do Twitter. Os dados para este projeto foram recolhidos através do SocialBus, uma ferramenta para a recolha, processamento e armazenamento de dados de redes sociais, nomeadamente do Twitter. O conjunto de contas usado para a recolha de dados foi obtido a partir de jornalistas de política portugueses, e a anotação de trolls foi realizada através de um conjunto restrito de regras comportamentais, auxiliada por uma função de pontuação.

Um novo módulo para esta plataforma foi desenvolvido, chamado Trollbus, que realiza a deteção de trolls em tempo real. Um dataset público foi também disponibilizado.

Os atributos do melhor modelo combinam os metadados do perfil de uma conta com os aspetos superficiais presentes no seu texto. O grupo de atributos mais importantes revelou ser os aspetos numéricos dos dados, com o mais importante a revelar ser a presença de insultos políticos.

Acknowledgements

Firstly, I would like to thank both my supervisors, Prof. Carlos Soares and Prof. Henrique Lopes Cardoso, who always made me feel like every meeting and every advice was a major step forward towards the project's conclusion, with them always showing interest in what I was doing, making me feel that every experiment (even the least successful ones) was heading towards a greater goal.

I would like to thank my mom for making this masters possible, as well as my friends and rest of my family. A specific thanks to Arian Pasquali for his tremendous availability in helping me, even far away.

Finally, I would like to thank Benedita, who had to endure my constant stress regarding this project, making me even more annoying than usual, and always gave her best to assure me everything would be ok.

Tiago Lacerda

“May we arise from the ashes of these beaten but not bowed, a little more modest about our forecasting abilities, and a little less likely to repeat our mistakes”

Nate Silver

Contents

List of Figures

List of Tables

Abbreviations

ADT	Abstract Data Type
ANDF	Architecture-Neutral Distribution Format
API	Application Programming Interface
AUC	Area Under the Curve
CAD	Computer-Aided Design
CASE	Computer-Aided Software Engineering
CORBA	Common Object Request Broker Architecture
FEUP	Faculdade de Engenharia da Universidade do Porto
IT	Information Technologies
KNN	K-Nearest Neighbours
MCC	Matthews Correlation Coefficient
RBF	Radial Basis Function
RF	Random Forest
SMOTE	Synthetic Minority Over-sampling TEchnique
SVM	Support Vector Machine
STF	Superficial Text Features
UC	Use Case
UF	User Features
UNCOL	UNiversal COmpiler-oriented Language
WWW	*World Wide Web*

Chapter 1

Introduction

In the age of overly politicised social networks and strong reactions to everyday events, Twitter stands as the place where trends, news and events emerge and are rapidly debated, in a way that helps to form opinions, getting the right audience to the right people, or even offering aid to those who need it. On the other hand, discussions on social networks tend to be very divisive. There is a group of users whose aim is to exacerbate that divisiveness, called **Trolls**.

This group of users has characteristics that distinguish them from normal, non-malicious, accounts [112], and a record of destabilizing not only the social network itself, but also of creating more than simple discussion, such as their messages transitioning onto real world discussions or even events, such as trolls being linked to the organization of violent protests in America [10]. Twitter now spends some of its resources trying to stop the proliferation of these accounts. It does not, however, do so in every country, as is the case with Portugal. This leaves the majority of these users virtually unaffected by these measures, and there has never been an effort in the Portuguese community to label them in an automatic manner. A phenomenon that gained traction in 2010 [18] was the widespread use of Twitter by journalists, even the least skilled in IT. The platform proved ideal for this professional group, since it provided the structure to update or comment news as soon as they surfaced. The nature of Twitter itself has led to an unseen degree of community interaction with journalists and of news sharing, which led not only to Twitter adapting to journalists, but to journalists adapting to Twitter. In the present day, as a result of this phenomenon, Twitter is a primary news sharing platform, involving groups of people with different ideologies and purposes. Troll users in specific aim at polarizing the userbase, causing conflict, and may do so through deceptive tactics [14, 5]. In an attempt to aid in the problem of trolls escaping detection mechanisms, several researchers have proposed their own approaches (Section 3.2). These approaches defined the problem of troll detection as a classification task, and use different sets of features to address it. This work aims at analyzing those approaches (mainly their features) and understand which of them are better to detect trolls, when applied to the Portuguese Twitter community.

1.1 Motivation

By knowing the dangers and risks of trolls, their activity, and what they represent, the main motivation for this work is to assert which are the best features for detecting trolls in the Portuguese userbase of Twitter, in order to aid in efforts to stop their proliferation. The topic of trolls has only recently received attention from the Portuguese media and scientific communities. Until 2020, this was not a theme approached by any of the biggest news sources in the country, and in terms of research, there was also no study found on this matter for the Portuguese community. With more attention and resources being spent on this topic, this work aims at aiding whoever wishes to study trolls and troll detection in Portugal. Another motivation for this work is the lack of empirical comparison of the different types of features proposed in the literature for troll detection. This implies that the work of this dissertation aims at aiding the local, Portuguese, community, but also the scientific community in general, as it presents an empirical study inspired in the available troll detection works, including the most recent ones.

1.2 Goals

With the motivations defined, there were several goals expected to achieve:

- Investigate the applicability of existing methods to detect trolls in the Portuguese Twitter
- Compare different sets of features for troll detection
- Extend SocialBus with a troll detection module

With the goals for the project defined, the research questions for the project were then defined.

1.3 Research Questions

The research questions raised for this dissertation are the following:

- How to systematically label an account as a troll?
- How to adapt the proposed features for troll detection to the Portuguese Twitter?
- Which of the proposed feature sets leads to better results in the Portuguese Twitter?
- Is the information provided by the different sets of features complementary?

1.4 Dissertation Structure

This section details the structure of this dissertation, for better guidance.

Chapter 2 presents the background on what is Twitter, existing infrastructure regarding Twitter, and the necessary terminology to understand the latter sections and chapters of this document.

Chapter 3 describes what is a troll and presents a review on the state of the art regarding Troll detection, dividing the works found according to several dimensions, finding the common points among studies, comparing their results, and explaining how and where they diverge. This Chapter also allows to provide a familiarization with the used features and methods, which are then adapted for the usage in work of this dissertation.

Chapter 4 details the features inspired in the state of the art created for Troll detection and the method that was created to better detect trolls and to prepare the data mining process. After those sections, the implementation of the software tool that was created is described, with a detailing of the stages of the software development methodology that was followed.

Chapter 5 elaborates on the empirical study that was performed on the data, with the features described in the previous chapter. For that, the parameters for the functions and methods developed in the previous chapter are specified. After that specification, the complete setup is described, followed by the description and results of an exploratory data analysis experiment.

Chapter 6 summarizes the implemented solution and the results obtained. A reflection of the proposed solution is presented, along with the answers of the research questions proposed.

Chapter 7 delineates the implemented solution and the results obtained. A reflection of the proposed solution and some recommendations for future work are presented.

Chapter 2

Background

In this section, the necessary concepts regarding algorithms, platforms, techniques and terminology related to Troll detection will be discussed. These are meant to provide the necessary knowledge to understand both the work performed in this project and of projects whose area of study is similar to that of this dissertation. This chapter also intends to give an insight and explanations on the technology needed to put in practice the process of troll detection. This is meant not just for this dissertation but for other referenced works. These insights range from the technologies needed beforehand, to the algorithms to use, to the metrics to evaluate them.

2.1 Twitter and Twitter Software

This section details Twitter and its characteristics, as it serves as basis to all subsequent work. Along with Twitter's definition, there is also in the subsections below a detailing of Twitter related software.

2.1.1 Twitter

Twitter is a microblogging and social network service in which users post short messages called *Tweets*. These contain a maximum of 280 characters and can be shared by registered users, who can post, like, and retweet tweets, while unregistered users can only read them. Users access Twitter through its web interface, through Short Message Service (SMS) or its mobile device application [139]. In Portugal, according to a study from SAPO, Twitter is relatively unpopular, being the sixth most used network in the country, with about 5.7% user share (lower when compared to Europe's 9.8% average) [117]. The same study states that the average Twitter user in Portugal is between 15 and 24 years old.

Despite these numbers, the work of this dissertation is important due to factors such as the growing user base of Twitter in Portugal [78], the fact that it is a platform where national party leaders express their opinions frequently, and the active media coverage given to Twitter by the media. An example of this coverage are the several documented cases of Tweets being discussed in national media [116], even appearing cases of public controversy over specific tweets [126].

This implies that although the user base is lower than Europe's mean, it has the ability to influence both the Portuguese media and its general population.

2.1.2 TweeProfiles

Twitter holds (and generates) a vast amount of data, which it allows to access through its API. From there, it is possible to access billions of tweets, with information regarding their attributes listed in Table 2.1.

Table 2.1: Tweet Parameters (Adapted from [86])

Concept	*Description*
Retweet (RT)	Share another user's tweet
Mention (@ + username)	Identify a user in a tweet
Reply (@ + username)	Answer to a previous user tweet
Hashtag (# + topic name)	Association of a keyword to a tweet
Location	User's geo-coordinates when sending a tweet

The API has restrictions, however, including 15 requests per access token (security credentials for a login session), and only allowing 15 requests per each 15 minutes [140]. *Faculdade de Engenharia da Universidade do Porto* has developed, over the years, an infrastructure that enhances the searching of Twitter data and the visualization of it, whether by providing additional attributes not available through the standard Twitter API, or by displaying data in a way that is not provided by Twitter, such as displaying the tweets' proliferation in a regional or world map. The platforms here referred are specifically **SocialBus** and **TweeProfiles**.

TweeProfiles is a *Data Mining* tool for multi-dimensional representation and visualization of trends in Twitter. Its main function is to analyze and display the data produced on Twitter, providing new representations of the gathered information, not available through the platform itself. An example of this representation is the display of the geographical areas where a tweet has been retweeted the most. The visualizations provided allow to extract knowledge either from the tool itself (through retweet maps, for example), or from the visual interface it possesses, which allows the study of the gathered information, which may reveal trends, for example, and may be of value for the user (e.g. journalists). The types of information displayed include the spatial, temporal, content and social dimensions (see Table 2.2) of the data, which are grouped through the *DBSCAN* density-based clustering algorithm [86]. TweeProfiles works by finding patterns of diversified data retrieved from the SocialBus platform (described in Section 2.1.3). Each domain is analyzed separately, and through a series of matrices generated by distance functions, the clusterings performed are then able to display the information in a multidimensional way (e.g. Spatio-temporal).

Table 2.2: Dimensions and Representations of tweets in TweeProfiles

Domain	***Tweet Attributes***
Spacial	Where the tweets were shared
Temporal	When (Date/Time) was the tweet shared
Content	What words the tweets contained the most
Social	Who shared the tweets (and how they were connected)

The representations and the clustering algorithms themselves were refined during the various iterations of TweeProfiles, either through the enhancement of existing features [81, 51, 87, 86], or the addition of new ones (e.g. picture recognition) [41, 114]. In recent years some features were discontinued, like the social dimension representation, but some aspects remained constant throughout the project's lifetime. One of them was the tool's dependency on SocialBus, as is possible to note in its architecture (shown in Figure 2.1). TweeProfiles has even been described as "a tool to analyze SocialBus data" [114], so it is clear the two applications are intertwined.

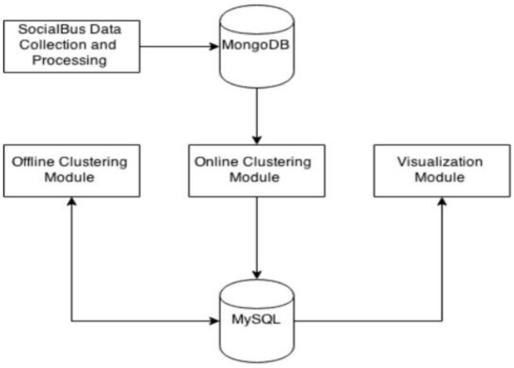

SocialBus Data Collection and Processing
MongoDB
Offline Clustering Module
Online Clustering Module
Visualization Module
MySQL

Figure 2.1: TweeProfiles Architecture. Adapted from [86]

In its current state, TweeProfiles presents the several dimensions interactively, allowing to adjust it according to one or more dimensions. The **geographical** distribution is visible in Figure 2.2, the **temporal** visualization is visible in Figure 2.3. and the **content** dimension is visible in Figure 2.4. The several dimensions can be viewed together in Figure 2.5.

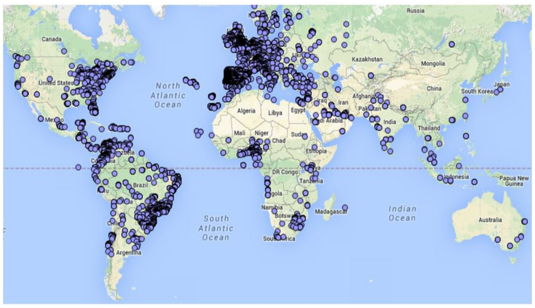


Figure 2.2: TweeProfiles Spatial Visualization. Reproduced from [87]

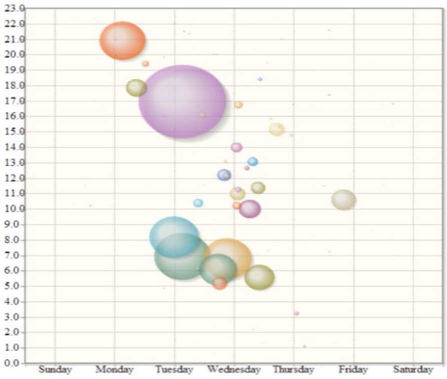


Figure 2.3: TweeProfiles Temporal Visualization. Reproduced from [81]

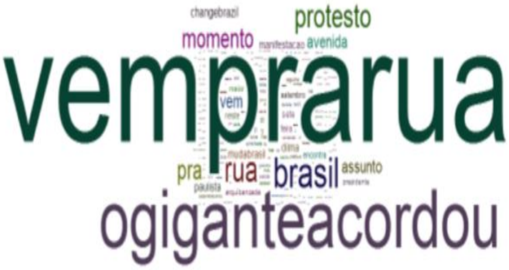


Figure 2.4: TweeProfiles Content Visualization. Reproduced from [81]

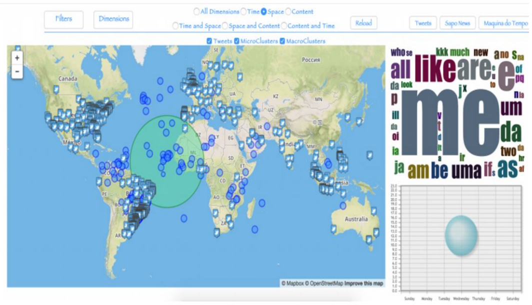

Filters
Dimensions
○ All Dimensions ○ Time ● Space ○ Content
○ Time and Space ○ Space and Content ○ Content and Time
Reload
Tweets
Sapo News
Maquina do Tempo
☑ Tweets ☑ MicroClusters ☑ MacroClusters
Canada
Australia
all
like
are
me
um
da
am
be
uma
as
© Mapbox © OpenStreetMap Improve this map

Figure 2.5: Several Dimensions at once in TweeProfiles. Reproduced from [86]

2.1.3 SocialBus

SocialBus (originally called TwitterEcho) is a platform developed by FEUP and SAPO labs researchers, which continuously gathers social network messages, supporting both Twitter and Facebook. It aims at aiding researchers in obtaining the needed quantities of data for data science in social media projects, with additional fields than those provided by the Twitter amd Facebook APIs. Messages are collected from an established connection to the appropriate API, such as Twitter's Streaming API, which are then sent to a message broker for the translation of the data into a more usable format (e.g. JSON). The following step handles message processing in two phases: stream processing, for operations such as language detection and tokenization, and batch processing to extract different kinds of knowledge. TweeProfiles uses these attributes, as it takes advantage of the stream processing to display the results (and how they change) in real time. The results of the stream processing phase are stored in a *MongoDB* database for posterior analysis[12]. The project has undergone some changes since its creation [81], with its last major change in 2016 [86]. SocialBus' most recent architecture is displayed in Figure 2.6.

An objective of this dissertation is to add a new dimension to SocialBus, allowing to extend it for troll detection. In practice, this new module will have to analyze accounts in real time and classify them as malicious or non-malicious according to a set of attributes in the user's profile and tweets.

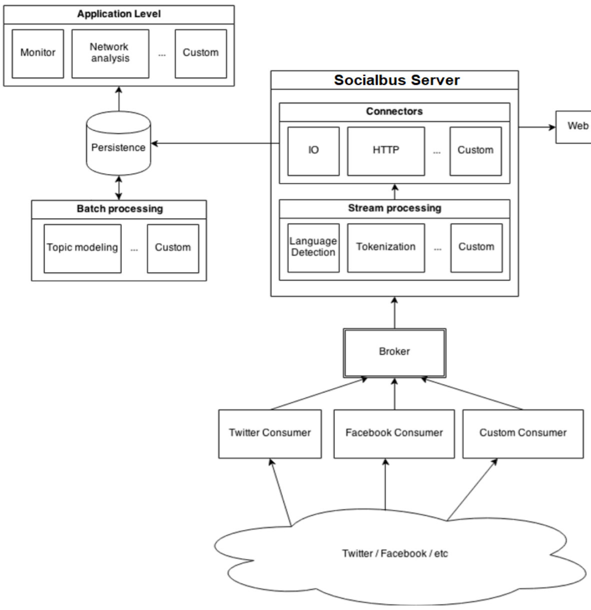


Figure 2.6: SocialBus Architecture. Adapted from [86]

2.2 Classification and Algorithms

When referring to machine learning, **Classification** is the problem of identifying to which set of categories a new observation belongs to, on the basis of a training set containing observations whose category membership is known. In the following subsections a formal definition of classification is given, as well as the detailing of algorithms that perform this task.

2.2.1 Classification

Classification belongs to the broader group of algorithms that perform **Data Mining** – the process of exploring large amounts of data with the goal of finding relevant patterns in it. This type of task is a supervised learning exercise, where the targets are also provided with the input data. There are many applications of classification in various domains such as in credit approval, medical diagnosis, target marketing, etc.

A formal definition of classification can be put as: Given a series of observations of pairs $D = \{(x_i, f(x_i))|i = 1...,n\}$, in which x_i represents an observation, and f represents an unknown function, a predictive data mining algorithm learns an approximation $\hat{f}$ of the unknown function f. That approximated function, $\hat{f}$, estimates the value of f for new observations x. According to the nature of f, it can perform classification tasks [44]:

$$y_i = f(x_i) \in \{c_1, ..., c_m\},$$

$$\text{in which} f(x_i) \text{ assumes values in a discrete, unordered set}$$

Some classification algorithms require a deeper explanation, since they are some of the most used in Troll detection approaches and require some knowledge about them to fully understand the following chapters of this document, as well as the most recent efforts in this area of study.

2.2.2 Support Vector Machines

Support Vector Machines (or **SVM**) is one of the most used algorithms in Troll classification today due to the nature of the data, such as the fact that most of the works in the area have binary labels (Troll/Non Troll). This algorithm works by finding a hyperplane in a higher dimensional space, which the features that describe the data points are mapped to. In the algorithm's domain, the data points belong to one of **two** categories. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a margin that is as wide as possible [82].

In the example visible in Figure 2.7 the classifications either belong to the blue group or the red group, with the hyperplane (green) dividing the feature space in a clear way. One example could be the division of a group of accounts into which ones were trolls and which were not.

This algorithm was designed for robustness to overfitting through the margin reduction, and this robustness comes from projecting the data to higher dimensionalities, having also the advantage of being memory-efficient. It does have disadvantages, however, such as its unsuitability for large datasets due to the resources required to store each result requiring memory that scales quadratically with the number of data points. Another disadvantage is the bad handling of noisy

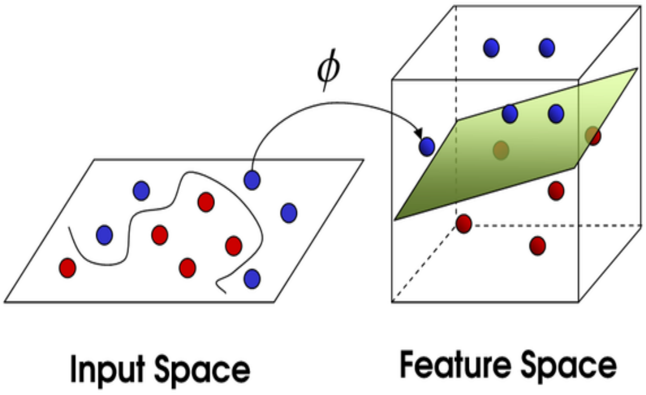


Figure 2.7: Examples of Hyperplanes in 2D (input space) and 3D (feature space), reproduced from [62]

data, and a record of underperforming there are more features than data points. It also produces no probabilistic explanation for the obtained classification [135].

2.2.3 Logistic Regression

Logistic regression is a statistical algorithm used to predict and/or classify effects and interactions with binary response data. It is one of the simplest tools for prediction, generating a discrete outcome from a set of variables, which may be continuous, discrete, dichotomous, or a mix of any of these. Logistic regression gives the conditional probability that an event will occur given the values of the independent variables as well as providing information on the causality of the different values [27]. This approach is popular among researchers in the analytical fields either as a solution or a baseline - a minimum or starting point to be used for comparisons with other, more complex algorithms.

The image seen in Figure 2.8 displays a simplified version of how logistic regression could be used to solve a troll detection task. The graph displays a binary target variable- **Troll or non Troll**, and a logarithmic curve (sigmoid function), that represents the **probability of being a troll**, which is dependent of the number of insults used.

This data mining method has the advantage that it is one of the simplest methods for binary classification, which is observable when analyzing its computational efficiency. It has also no necessity for the input features to be scaled or altered, as well as the fact that the results outputted are easy to understand. It does have disadvantages, however: it performs better by removing similar and unrelated features (requires data reduction efforts), and is unable to solve non-linear problems. More complex algorithms may produce better results for the same problem [34].

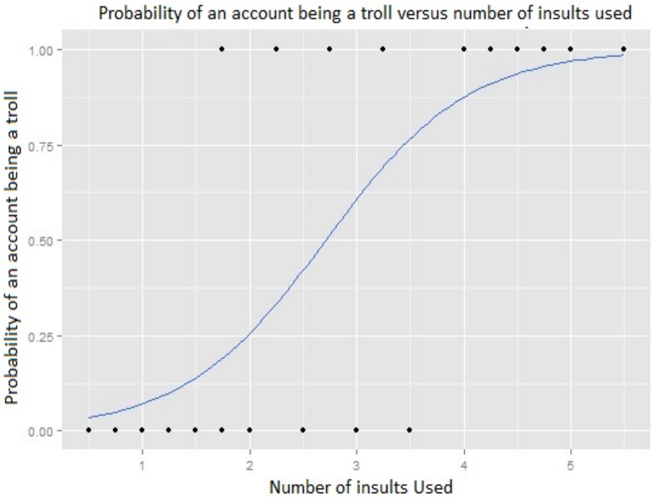


Figure 2.8: Logistic regression graph example: Probability of an account being a troll versus number of insults used, adapted from [85]

2.2.4 Decision Tree

Friedl and Brodley describe decision trees as “a classification procedure that recursively partitions a data set into smaller subdivisions on the basis of a set of tests defined at each branch (or node) in the tree” [13]. The tree is composed of a root node (formed from all of the data), a set of internal nodes (branches), and a set of terminal nodes (leaves). Each branch in a decision tree has only one parent branch and two or more descendant ones, while leaves have none. In binary classification, the tree’s leaves are only of two kinds.

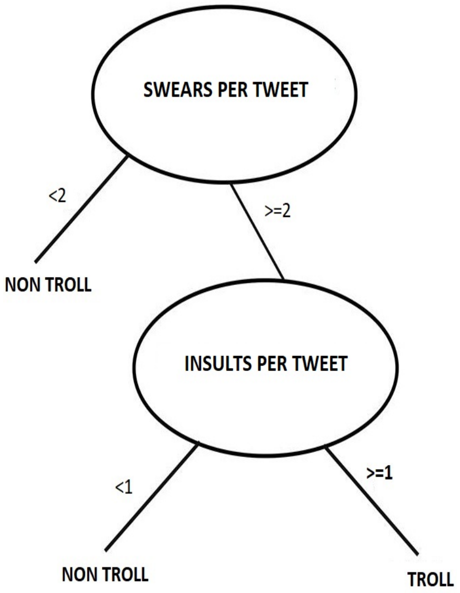


Figure 2.9: Example of a decision tree for binary classification, with a troll detection example

The decision tree depicted in Figure 2.9 is applied to the same example. The tree, in this case, is easy to interpret. However, as the tree grows, both the branches and the number of leaves grow as well, which may lead the tree to become more complex and difficult to comprehend. Decision trees are proven to be efficient in terms of computation, as they do not require normalization of data (scaling variables to have values between 0 and 1) and they have inherent mechanisms which makes them resistant to missing values [132]. These characteristics make decision trees one of the most popular algorithms, whether as a single solution by themselves, or by grouping several together, constituting a Random Forest. They are, however, especially sensitive to the training data as their error has a high variance value. This can lead to minor changes in the data originating highly different models, which makes this algorithm sensitive to overfitting.

2.2.5 Random Forest

The Random Forest algorithm is an *ensemble algorithm*— a set of individually trained classifiers, in this case, of decision trees, whose predictions are combined when classifying novel instances[96]. They operate by constructing a series of decision trees (Figure 2.10) when training the model. Once the creation of trees is finished, the algorithm outputs the class that is the most frequent (mode) verdict among the individual decision trees[53].

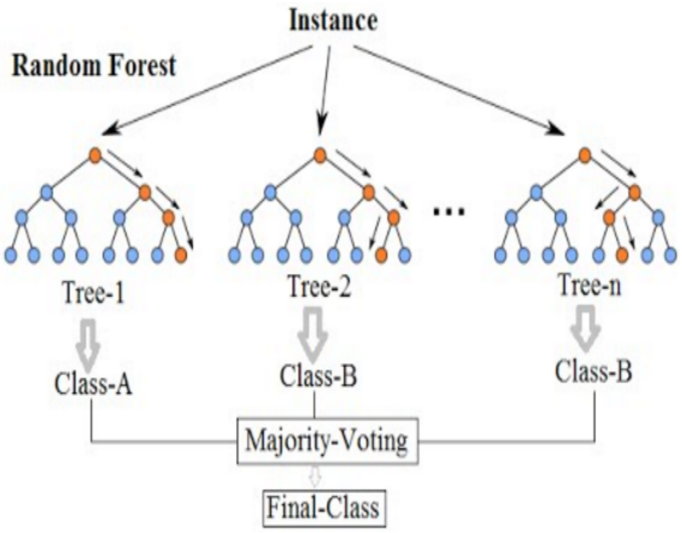


Figure 2.10: A representation of a random forest algorithm and the many trees that constitute it. Adapted from [63]

This algorithm has mechanisms that allow it to reduce the risk of overfitting that comes from the utilization of decision trees. Unlike the latter, they tend to be harder to interpret. Despite that inconvenience, studies exist that explicitly claim that random forests are the most likely to give accurate results for data mining problems [40], and are frequent in many classification studies (Section 3.2.10).

2.2.6 K-Nearest Neighbors

K-nearest neighbors (or **KNN**) is a supervised learning algorithm which estimates how likely a data point is to be a member of the existing classes, depending on what class the data points nearest to it are in. The algorithm determines the membership of an observation, based on the states of the points that are near it. The range (k), represents a sample of the data on which the algorithm will base its decision when assigning a new observation to a class: If the majority of the points are in group A, then it is likely that the data point in question will be A rather than B, and vice versa [134]. If the K value is 3, then the 3 nearest neighbors will decide to which class the analyzed sample will be assigned (Figure 2.11). The optimal value for K may vary depending on the problem in question.

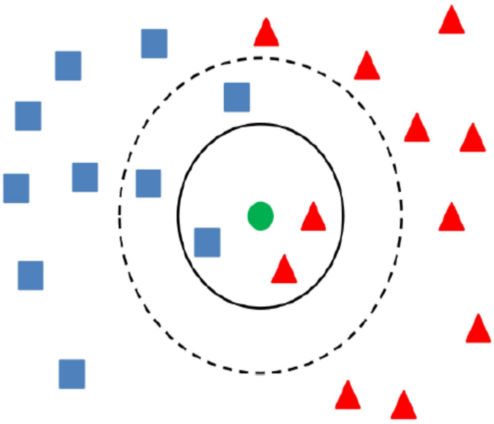


Figure 2.11: Representation of the K-nearest neighbors algorithm for a binary classification problem. Adapted from [3]

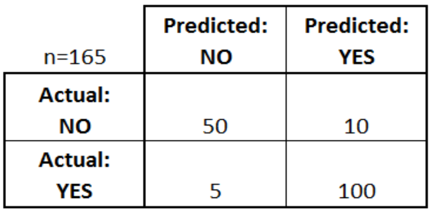
Applying this algorithm to troll detection, if a membership of an account was unknown, but his behavior was, his nearest neighbours would be the accounts that displayed similar behavior. If the K accounts in the group (his neighbours) closest to him were trolls, the KNN algorithm states that it is most probable that the account is a troll, and vice versa. KNN has relevant advantages, such as adapting as new training data is collected, allowing the algorithm to respond quickly to changes in the input during real-time use. Another advantage is the fact that it does not build a model and therefore needs no training step. It has significant disadvantages, however. A common disadvantage is that for large datasets, it tends to be slow and computationally heavy [46]. Another relevant disadvantage is the fact that it suffers from the **Curse of dimensionality**: KNN works well with a small number of input variables, but as the numbers of variables grow, the algorithm starts outputting wrong predictions regarding the output of new data points [83].

2.3 Algorithm Evaluation

To check whether the algorithm is producing interesting and overall correct results (in the case of this subject, if it is correctly identifying trolls and normal accounts), the models and their results need to be subject to a series of evaluation measures and metrics in order to verify that the outputted data is not only correct, but also in the range of values desired by the researchers. Evaluation measures play an important role in machine learning because they are used, as said by Lavra et al., "not only to compare different learning algorithms, but also often as goals to optimize in constructing learning models" [71]. In the following subsections, the most relevant measures for troll detection will be detailed.

2.3.1 Confusion Matrices

The Confusion Matrix is the baseline for evaluation measures in Data Mining. It is (usually) a 2x2 table layout that allows visualization of the performance of an algorithm. Each row of the matrix represents the instances of a predicted class, while each column represents the instances of the actual class [119]. The name arises from the fact that it allows to check if the model is confusing two classes (i.e. commonly mislabeling one as another). It allows to visualize the true positives, false positives, true negatives and false negatives, and from these, the calculation of several metrics is possible.



n=165	**Predicted: NO**	**Predicted: YES**
Actual: NO	50	10
Actual: YES	5	100

Figure 2.12: An example of a confusion matrix. Reproduced from [119]

In this example, the True Positives were 100 (Actual Yes and Predicted yes), with the True Negatives being 50. On the other end of the spectrum, there were 10 False Positives and 5 False Negatives. From these metrics, it is possible to calculate the measures that will be expanded upon below.

2.3.2 Precision

When referring to classification, precision (or confidence) denotes the proportion of predicted positive cases that are *True Positives*, as opposed to *False Positives*. It refers to the percentage of the results which are relevant. It can be calculated through the division of the True Positives value (TP) by the total number of Predicted Positives, which is the sum of true positives and false positives(FP)[106]:

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

2.3.3 Recall

Recall is the proportion of True Positive cases that are correctly predicted by the model. It is calculated from the ratio of the True Positive cases with the sum of the True Positive cases and the

False Negative cases (FN):

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

2.3.4 F1 Score

The F1 Score (or F-measure) is defined as the harmonic mean of precision (P) and recall(R)[118]. It serves as the compromise between precision and recall: a measure that can describe the model/algorithms performance, with the worst value it can have being 0 and the best being 1. It can be a better metric than accuracy for certain cases, as is the case for imbalanced datasets:

$$F = \frac{2PR}{P+R}$$

2.3.5 AUC - ROC Curve

A receiver operating characteristics (ROC) graph is a technique for visualizing and selecting classifiers based on their performance[38]. ROC is a probability curve and the *Area Under the Curve* (AUC) represents the degree or measure of separability: It tells how much a model is capable of distinguishing between classes given a certain threshold. This measure can estimate how good a model is at distinguishing classes, and presents itself as a better alternative over metrics like accuracy for certain cases (e.g.: imbalanced datasets). In general and applied to this work's area, the higher the AUC, the better the model is at distinguishing between classes, in this case trolls and non trolls.

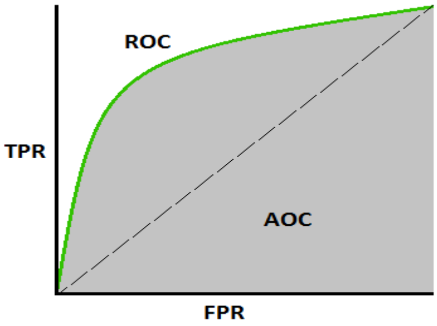


Figure 2.13: Visualization of the AUC ROC curve. Reproduced from [93]

In Figure 2.13, the area shaded in grey is the model's performance. The vertical axis is the True Positive Ratio (true positives/true+false positives), and the horizontal axis is the False Positive Ratio(false positives/true+false positives). The closer the area gets to the vertical axis, the better it is at distinguishing the two classes apart. The green line represents the performance of the model, which is usually at its best in the "elbow" section. From this metric, it is possible to visualize the probability that the classifier will rank a randomly chosen positive example higher than a randomly chosen negative example.

2.3.6 Matthews Correlation Coefficient

A threshold of a classifier determines the score above which an observation is assigned the positive class. The metric of AUC (Section 2.3.5) illustrates the diagnostic ability of a binary classifier system as its threshold is varied [38]. Increasing the threshold will result in fewer false positives, but more more false negatives, and vice versa. A higher threshold implies a higher recall and a lower precision, while a lower threshold implies a higher precision and a lower recall.

Classification algorithms generate models with a threshold of 0.5 by default. With imbalanced datasets, this value is usually inadequate.

Given the model's results, a possible method of visualizing the best threshold is through the plotting of the ROC curve and manually analyzing it, in order to find the most acceptable value given the experimental requirements. This method has the problem of possibly being imprecise and being difficult to apply in an automatic manner.

To solve this matter, Chicco and Jurman demonstrated that measures such as the AUC could display misleading results when applied to imbalanced datasets (conflicting with other studies), and advocated for the use of the *Matthews Correlation Coefficient* (MCC) [22]. According to the authors, this metric produces a more reliable statistical rate, since the MCC outputs a high score only if the prediction obtained good results in all of the four confusion matrix categories: true positives, false negatives, true negatives, and false positives.

$$\text{MCC}=\frac{TP\times TN-FP\times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$$

2.3.7 Class Imbalance

Troll detection is, in several cases, a problem with class imbalance. Some researchers opt for a simulation of the reality of Twitter, which possesses more normal accounts than malicious ones [57]. When generating a dataset with the aim of simulating Twitter's environment, the data possesses a disproportionate ratio both of troll to normal users and tweets. This creates an acute *class imbalance*, which can negatively impact classification tasks [115], and is known to directly impact algorithms used in troll detection exercises (Section 3.2.10), such as decision trees, and subsequently, random forest [23].

There are several ways of dealing with class imbalance, but for this project three main methods were considered:

- **Undersampling**: Class imbalance is lessened by reducing the majority class to a random sample, in order to obtain a distribution of, for example, 1:100, 1:10, 1:2, or even a 1:1 class distribution.

- **Oversampling**: Class imbalance is reduced by taking a random sample of the minority class and supplement the dataset with copies of the sampled classes. This process can be repeated more than once [48].

- **SMOTE**: Acronym for *Synthetic Minority Over-sampling Technique*, it arose as a solution for oversampling's limitation – Not adding new information to the dataset, which increased the risk of overfitting. It works by selecting examples that are close in the feature space, drawing a line between the examples in the space and drawing a new sample at a point along that line. A random sample from the minority class is first chosen, then the nearest neighbors for that sample are found (typically 5). A randomly selected neighbor is chosen and a synthetic example is created at a random point between the two examples in the feature space [72].

2.3.8 Performance Estimation

There are several techniques that aim to confirm that the values outputted by the generated model are correct. In practice, performance estimation denotes a task of estimating the loss that a predictive model will incur on unseen data, that is, how good a model is at predicting unseen data[7].

2.3.9 Holdout

Holdout is one of the simplest forms of model evaluation. The dataset is split into a train and test set. The training set is the segment of data the model is trained on, and the test set, composed of the remaining data, is used on the model, and after the predictions are made, evaluation measures are applied to assess how well a model predicts the classes of this new data. A common split when using the holdout method is using 80% of data for training and the remaining 20% of the data for testing [4].

2.3.10 K-Fold Cross Validation

Cross-validation is a re-sampling procedure used to evaluate machine learning models when the test data is limited. The dataset is split into k equal parts and a holdout is performed k times, where in each execution a different partition of data is used as the test set, while the others are used as the training set.

For each model generated from this method, its evaluation score is kept, while the model itself is discarded. In the end of the process, the performance metric for the model will be the *average* of all the obtained scores.

When using this technique in time-referenced data, long term predictions may not be viable with due to the dependence between observations. Additionally, trolls are known to adapt their behavior to avoid being detected through classification techniques, which may make the usage of this method questionable. One other (less relevant) aspect that needs care when performing this technique is the number of K, since a high number can create computational issues.

2.3.11 Leave One Out

Leave-one-out cross-validation is a special case of cross-validation where the number of folds equals the number of instances in the data set. According to the definition given by Sammut and Webb [70], the algorithm is applied once for each instance, using all other instances as a training set and using the selected instance as a single-item test set. In case of large datasets, this estimation method becomes too heavy computationally.

2.4 Natural Language Processing

Natural Language Processing (NLP), is a range of computational techniques for analyzing and representing text for the purpose of achieving human-like language processing for a range of tasks or applications [74]. This area of research is of fundamental importance regarding troll detection, as speech is the way trolls propagate their message (Section 3.1).

2.4.1 Data Preparation

In an NLP process, several stages in preparing the data are needed to ensure the best result possible for a given language processing task. Those steps include:

- Tokenization
- Stopword Removal
- Stemming
- Lemmatization
- Text Representation

As a note, these are cases where these steps are not always used, as they present no improvement in results. For the troll detection works which perform text analysis, they are in fact used, although with some noted exceptions (Section 3.2.9).

2.4.2 Tokenization

Tokenization is described by Trim as "the process of segmenting running text into words and sentences" [136]. Text is a linear sequence of symbols (characters, words or phrases), so before performing an NLP task, a pre-processing must be done. The requires a *tokenization* process – to be segmented into linguistic units such as words, punctuation, numbers, alphanumeric characters, etc. (Figure 2.14) [136].

The identification of units that do not need to be further decomposed for subsequent processing is of great relevance. Errors made at this stage are likely to induce more errors at later stages of text processing (e.g: lower accuracy), so tokenization must be done carefully.

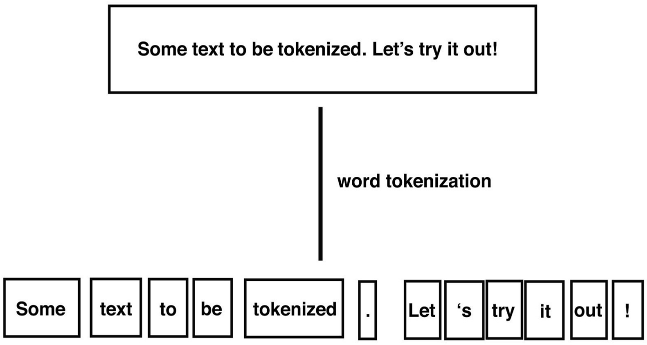


Figure 2.14: An example of a word tokenization process. Adapted from [63]

2.4.3 Stopword Removal

Stop-words are usually irrelevant to the text analysis, as they carry no information themselves (e.g. conjunctions). After removing stop-words in a document, the remaining text contains only the **relevant** words [79]. While there are studies that correlate troll language with stopword usage [57], these are exceptions and will be addressed in Section 3.2. In the normal data preparation process, stop word removal is a common step, although for more recent representations, such as BERT [31], this stage is discarded.

2.4.4 Stemming

Another phase in the data preparation process is *Stemming*, described by Pande et al as "the process for reducing inflected (or sometimes derived) words to their stem, base or root form" [98]. This process is done as a way to approach words of the same family or of similar meanings mainly by truncating prefixes or suffixes (for example, the words *Doggy*, *Doglike* and *Dogs* all get stemmed to the word **Dog**). This process may reduce redundancy in the corpus, which can result in performance and computational advantages. Its performance varies in the language and the type of words, however. One of the most used algorithms for this task and especially for the English language is the **Porter Stemming Algorithm**, which several studies have labelled as an effective stemming tool [56, 49]. This process, along with stopword removal, has had fewer usage with the appearance of newer approaches such as the ones based on subword tokenization, character embeddings or language models. For approaches that use the bag-of-words representation (Section 2.4.7), however, stemming is an important step, and is relevant for this work, as several troll detection works use the BoW representation.

2.4.5 Lemmatization

When performing stemming, lemmatization usually accompanies it. It is the algorithmic process of determining the *lemma* (the base form under which the word is entered in a dictionary, the simplest/canonical form of a word). Unlike stemming, lemmatization depends on correctly identifying the intended part of speech and meaning of a word in a sentence, as well as within the larger context surrounding that sentence, such as neighboring sentences or even an entire document. As a result, developing efficient lemmatization algorithms is an open area of research. If done correctly, it also provides a productive way to generate generic keywords for searching documents [103].

2.4.6 Text Representations

Text representation in NLP is seen as a broad term, referred by Chen et al. as "methods, systems, and apparatus, including computer programs encoded on computer storage media, for computing numeric representations of words" [19]. In practice, text representation is a fundamental problem in text mining, that aims at numerically representing unstructured text documents to make them mathematically computable. There are several algorithms that aim at doing so, and will be further explained in the subsections below.

2.4.7 Bag of Words

Bag of Words (**BoW**) is one of the most popular models for text representation. In this model, a text (such as a sentence or a document) is represented as the bag (multiset) of its words, disregarding grammar and even word order but keeping multiplicity [150]. This approach uses the tokenized words for each observation, and calculates the frequency of each token. An example of the BoW representation reproduced from [35]:

“It was the best of times”

“It was the worst of times”

“It was the age of wisdom”

“It was the age of foolishness”

In the BoW model, each sentence is treated distinctively. After that, a processed list of words is made from the four documents:

The next step is to create vectors for each document, where the frequency of the 10 distinct words is counted (Table 2.3).

With the vectors created, feature engineering is now possible. This representation has well known flaws, such as sparsity and sensitivity to overly common words, making a word with low significance have a high value of relevance attributed to it, simply because it appears commonly across a document or documents. One other limitation is that it doesn’t respect the semantics of a word, leading to the words "car" and "automobile" to be treated as two completely separate terms, for example. Because of these types of limitations, BoW has been losing usage over the years, with other representations (Section 2.4.8) or word embeddings occupying its space in the most recent studies being published.

Table 2.3: Bag of Words Vectors

it	was	the	best	worst	age	of	times	wisdom	foolishness
1	1	1	1	0	0	1	1	0	0
1	1	1	0	1	0	1	1	0	0
1	1	1	0	0	1	1	0	1	0
1	1	1	0	0	1	1	0	0	1

Table 2.4: An example of the vectors produced by the bag of words representation

2.4.8 Tf-idf

Term Frequency - Inverse document frequency (TF-IDF), is a text representation technique that aims to overcome the common difficulties that arise when using BoW. When analyzing the text, the tf–idf value increases proportionally to the number of times a word appears in the document and is offset by the number of documents in the corpus that contain the word. This metric is given by two formulas: *Term Frequency* and *Inverse Document Frequency*. Given the total number of times a term t appears in all the terms of a document d, the term frequency of the word in that document is calculated by the formula:

$$tf = t/d$$

The inverse document frequency measures how important that term is in the documents. The closer it is to 0, the more common a word is. This metric is calculated by the logarithm of the total number of documents, D divided it by the number of documents that contain a word:

$$idf = log(\frac{D}{count(d \in D : t \in d)})$$

Multiplying these two numbers results in the TF-IDF score of a word in a document. The higher the score, the more relevant that word is in that particular document.

$$tfidf(t,d,D) = tf(t,d) \cdot idf(t,D)$$

This helps to adjust for the fact that some words appear more frequently in general [133].

A survey published in 2015 claimed tf–idf as one of the most popular term-weighting schemes, with 83% of text-based recommender systems in digital libraries using it at that time [61]. However, with the increasing popularity of word embeddings (Section 2.4.9), that value has declined in the recent years.

2.4.9 Word Embeddings

An embedding is a mapping of a discrete –*categorical* – variable to a vector of continuous numbers. In the context of text, this means that words, sentences and whole documents are represented as mathematical values [64]. Word embeddings rose as an answer to a recurring problem in text representation - words with similar meanings were treated by the models as unique symbols. In this new approach to represent document vocabulary, words were represented as dense vectors, derived using various training methods inspired from neural-network language modeling [73]. This approach proved useful in several studies of different areas [137, 8], and as new models are developed, new studies arise, with interesting results [31]. Examples of models that rely on embeddings are Word2vec, FastText and GloVe. Word2vec, for example, is a group of related models that produce word embeddings. These models are shallow, two-layer neural networks that are trained to reconstruct linguistic contexts of words. Word2vec, as described in its original paper, "takes as its input a large corpus of text and produces a vector space, that is commonly of several hundred dimensions, with each unique word in the corpus being assigned a corresponding vector in the space" [90]. Word vectors are positioned in the vector space such that words that share common contexts in the corpus are located close to one another in the space (Figure 2.15) [84].

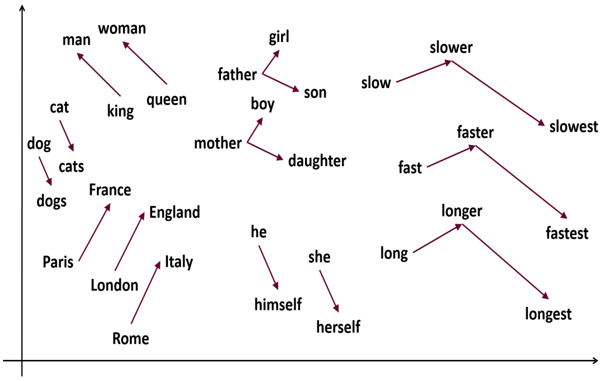


Figure 2.15: An example of a Word2vec mapping, with the position of the words in the corpus and their closest neighbours. Reproduced from [99]

This type of representation can bring advantages since the data can be fed online (requires little preprocessing and memory) and is intuitive to understand. It does have disadvantages, however, like the relationships between words not being explicitly defined: An example of this can arise from training the representation on a document where two words with no close meaning get used frequently together – the two words may be mapped as the closest to each other, which may generate confusing embeddings. Scaling to new languages requires re-training new embedding matrices, meaning cross-lingual use of the same model is not possible, and the model can be difficult to train in large vocabularies[92].

With a background regarding classification algorithms and natural language processing provided, it is now possible to describe in detail the classification works of a specific kind, in this case, the labelling of accounts as Trolls or Non Trolls.

Chapter 3

Troll Detection

The tools and techniques described in the Background (Chapter 2) are fundamental for the specific task of troll detection. This task is a classification exercise, in which Data Mining techniques attempt to separate the normal accounts from the troll ones. Firstly, however, a background on trolls must be given, as well as their definition, which may be subject to discussion, and as will be explained, can rely on accusations from the community. Once this definition is understood, the state-of-the-art on approaches to detecting trolls will be given, as well as a discussion of their results, methods, and overall effectiveness.

3.1 Trolls

Trolls are online accounts that are created and operated with the primary goal of manipulating public opinion. Their behavior has been the subject of several studies, and have multiple events tied to them over the years. In the following subsections, a formal definition for a troll will be given, as well as a contextualization of political trolls throughout the world and the years.

3.1.1 Troll Definition

Troll is a term used broadly in today's internet context. For this work, however, this term is narrowed to the definition of political trolls, which are a specific subgroup of trolls, who practice trolling (making unsolicited and/or controversial comments on internet forums with the intent to provoke an emotional reaction to engage in a fight or argument) specifically regarding political matters. An example of the behavior of this type of trolls is the promotion of divisiveness or conflict on some social or political issue[42]. Trolls have inclusively changed how information is consumed in today's online context, with an added level of suspicion over every argument and fact, as well as the accusations of being a troll at an all time high. They are different from bots, in the sense that they are not automated, and they differ from individuals who discuss politics online, since they display a clear intent to deceive or create conflict. Their short-term objective is to start quarrels, upset people and provoke emotional responses in them. Politics, sports and other kinds of discussion rely on logical argumentation, but it is not uncommon for these to be or become of the emotional kind. Trolls thrive on emotional discussions, and tend to avoid or shift logical discourse to a more emotional one. Their long-term goal is to sow discord and ultimately distract public opinion by normalizing tangential discussion.

Some authors summarize Troll behavior in three key points [123]:

- **Repetitiveness**: Trolls produce a large number of messages
- **Destructiveness**: Troll messages express negative sentiments to sow discord
- **Deceptiveness**: Troll messages may be deceptive to achieve their objective of sowing discord and normalizing tangential discussion (e.g. by sharing fake news)

The definition of Troll can be fluid and ambiguous, which may lead to arguable Troll accusations. There are inclusively cases where entire organizations were acting in concordance with the behavioral described above. An example of this ambiguity is the behavior of a major British political party, which was found spreading misinformation and engaging in what is considered Troll behavior, leading even to warnings by *Twitter, Inc* [25]. This case raises even the question of whether a public entity can be a troll account or not. An important note that must be present when classifying trolls is to avoid **censorship** – a troll is not an account whose opinions the researchers or the overall public does not agree with, but an account that deliberately aims at pushing agendas and normalizing discussion and harassment. Having loose criteria on what constitutes a troll may leave to incorrect flagging of users, and the subsequent raising of *freedom of speech* discussions.

One term that has been used when speaking of trolls, as well, is the reference to **Troll Farms**: State-sponsored anonymous Internet political commentators and trolls linked to a certain government. There are several countries suspected to have troll farms, such as Russia, USA, Turkey and the Phillipines [52, 122]. These act in group, and conduct large-scale orchestrated trolling and disinformation campaigns to promote propaganda. In most occasions, this type of propaganda in other countries extends to promoting far-right divisiveness parties, and are observed to appear in the months prior to elections [5].

3.1.2 Russian Trolls

In the United States, Troll accounts began gaining notoriety in the 2016 presidential elections, and were linked to Russia. According to an official Twitter statement, 1.4 million people interacted with Russian trolls during that time frame [145]. In the same year, in the election that determined that the United Kingdom would leave the European Union, Russian interference was discovered, and its extent is being understood years after [113]. This intent of destabilizing political discussion and of sharing fake news was notoriously present in 2 752 now deactivated Twitter accounts that were later identified as being tied to Russia's "Internet Research Agency", a known troll farm. The U.S. Congress released a list of these accounts as part of the official investigation of Russian efforts to interfere in that country's elections [6]. Accounts of this type are still active today, making a troll detection effort as relevant as it was in 2016, more-so with the 2020 American presidential elections approaching.

3.1.3 Trolls In The Portuguese Universe

In the Portuguese context, trolls never had a significant expression until 2019. This is partly because Twitter trolls tend to favor far-right movements [5], which have historically low numbers in Portugal [28], and because of the low usage of Twitter by the Portuguese population in general (section 2.1.1). In 2019 this context changed, however. As the campaign for the legislative elections began to intensify, fake profiles began to surface, commenting on popular hashtags and promoting discussions with users, with the majority of them criticising the left-leaning parties [11]. The disinformation campaign went on through the election process, affecting the user base and creating an environment filled with accusations from all sides of users being *paid trolls* for parties. In the end, one of the parties most accused of spreading disinformation, the far-right party *CHEGA* [77, 36, 100, 16], was elected, being the first party of its ideology in the Portuguese parliament in 45 years of democracy [76]. In December, in the voting for **Word Of The Year 2019** in Portugal, *misinformation* was one of the three finalists [105].

In spite of these events and signs of the proliferation of trolls in the country, the Portuguese media is only just starting to delve into this subject [16], as opposed to the troll profile uncovering performed by the press in other countries [131, 91, 25, 88].

3.2 Troll Classification

Troll detection is a classification problem applied to social networks and online communities, where the objective is to distinguish users with normal online activity from those with a malicious one. Works of the like have proven results in uncovering information that was not detected by human efforts [113].

Using the previously referred formalization (Section 2.2.1) to this specific context, x_i represents an unlabelled user, $\hat{f}$ is a function that allows to estimate the membership of f for new observations of x, $y_i = f(x_i) \in \{c_1, ..., c_m\}$, in which x_i is an array that contains different types of features and y_i is either Troll or Non Troll (Sections 3.2.8 and 3.2.9).

Some approaches for the classification of these users have been proposed, although not much due to the fact that it is a recent topic. There is an observable difference between the published articles from before and after 2016 (year of the American elections). After this event, trolls became more present in the public knowledge, the troll definition was restricted to *political* trolls, and started to be linked with Russia. There are few to no articles from after this date that do not mention this country.

Research on the field shows trolls and troll behavior are largely independent of the type of platform they're in (whether they are in different social networks, news forums, newspaper websites) [42]. This means that what is considered troll behavior in a platform is true for all platforms, and that premise is used here when showcasing the most recent studies on troll detection, not restraining it only to certain social networks or communities.

In order to identify the relevant papers of this area of study, the articles were searched firstly by keywords mostly referencing trolls, such as "troll","Russian Troll", "Troll detection","Antisocial behavior", etc. After this first method was employed, the references in the papers found were analyzed. Each of the papers found was characterized according to 13 dimensions (Table 3.1). The dimensions will be detailed in the following subsections.

Table 3.1: Dimensions defined for the approaches

Dimension	***Meaning***
Problem Description	High level task explanation
Data Source	Where the data was collected from
Data Collection	How the data was collected
Dataset	Data and Features used
Target	What was to be labelled (Users or Messages)
Label	What is the value used to classify as troll or not
ML Task	What is the type of machine learning task
Non Text Features	All features used besides text
Text Features	Features extracted from text
ML Algorithms	Machine Learning algorithms used in the task
Decision	How does the tool determine if troll or not
Feature Importance	How was the importance of features measured
Evaluation Measure	What measure was used to evaluate the model
Experimental Setup	Conditions used in the experiment

3.2.1 Problem Description

The goals and problems vary across the papers, although the general goal is the discovery of trolls and/or troll behavior. The dominant type of problem is centered around discovering malicious accounts in the spectrum of politics. There are exceptions, however, such as Trolls specific to a certain platform (its findings may be less transversal to other areas). When trying to identify trolls of the political type, the approaches may vary in language and platform, but the applied principles (what constitutes troll behavior) are generally the same, whether applied to American platforms [57, 144, 5], Spanish news forums [30, 29], or Bulgarian ones [89, 88]. In the type of studies that focus on messages, the problem can be put as finding an answer to the question “What classifies a message as trolling?” [123, 89]. In Seah et al.’s study, the main objective is to ascertain if troll messages can be identified in a specific local dialect [123], for example. In the more specific problems, there is the study conducted by Haque, where the subject was the discovery of troll users in the popular series "Twitch Plays Pokemon" [130], which is an extremely specific troll detection problem. Galan-Garcia et al.’s problem is the most specific, consisting of a school that had one malicious account practicing cyberbullying on the students, and their objective was to discover who was behind it [43]. While this last study is a troll classification effort, its problem is so specific that few other researchers may draw relevant ideas from them. An effort of uncovering the people behind the troll profiles in a non-closed environment is nearly impossible, furthermore, the benefits of uncovering trolls' identities are arguable (Section 3.1.2).

3.2.2 Data Source

The source of the data varies across the studies, with the target platform varying with the problem itself: studies whose objective is to find Political Trolls focus on local news forums (the majority), or in the case of social networks, the researchers tend to follow accounts that are linked to or talk primarily about politics, for example. In other types of troll detection studies, like the more population specific effort of Seah et al.[124], the platform is more specific to the country, with the large majority of its user-base limited to that location.

Table 3.2: Source of The Data

Source	***Literature***
Twitter	[57] [43] [59] [42]
Other Social Networks	[33] [50] [144]
News Forums	[89] [20] [29] [68] [88] [30]
Community Forums	[123] [32]

As Table 3.2 indicates, studies on troll detection have varied data sources, with no platform being dominant, but a vast number of news forums (most of them whose user base is limited to its origin country) being covered. Knowledge gained from these studies is interchangeable amongst the various approaches, which, as will be presented, can be reflected with the features used (Sections 3.2.9, 3.2.8). Due to this fact, there is a distinguishable convergence being observed: There is a common group of works that is regularly referenced in the most recent troll/ antisocial behavior detection efforts [42, 57, 50, 6, 88].

3.2.3 Data Collection

The collection process of the data is a fundamental step in the data mining process, since a faulty collection may compromise the learning performed by the algorithms, or may miss important attributes in the domain. The data collection dimension has three key concepts subject to it: Technology, sampling and annotation process. When there is a need for a collection technology method, the usage of APIs and crawlers is the most frequent[123, 57, 68]. There are occasions where the usage of additional technology is not needed. In those cases, forums and social networks give researchers their own suspected troll datasets. In these instances, the platforms use the feedback they get from the user-base and build a collection of suspect users and user data, providing free access to it [144]. In these cases, the feedback from which the dataset is built is not available for every platform, and even that method is subject to having faults, however, it is an effective annotation process. To all these methods there is the exception of Jansson and Oskar, whose troll messages are simulated by the researchers themselves [59]. There can never be a way of verifying that no important words, users or threads were missed during the data collection stage. It is possible to argument that by getting data from the platforms' APIs, for example, the corpus will always be restricted to keywords, threads, or users selected by the entities responsible for the information gathering, as well as the labelling of troll or not – **Faulty Sampling**. The data collection from the APIs will have no way to verify if the entirety of important data was gathered, but is still a viable and accessible way to create either the negative example set, or even the entirety of the set itself.

3.2.4 Dataset

The constitution of the dataset and its classes varies across the various approaches. The majority of cases use datasets on which half of all messages/posts/comments come from trolls, and the other half from *normal* users. This technique is the most common in classification works, since the majority of algorithms used for binary classification (like random forest) display better results when using a balanced dataset, and if it is too unbalanced, these can have good accuracy on the majority class but poor accuracy on the minority class(es). This happens due to the influence that the larger class has on traditional training criteria [45]. Despite proving better for this type of algorithms, a balanced dataset diverges from the existing context, as real world data is far from homogeneous: it is believed that trolls constitute roughly 1% of all Twitter accounts. Im et al. try a closer approach to this measure, and use a dataset of 29 million tweets, with only 1.6% of these being from trolls, provenient from the list released by the US congress on suspected Russian trolls (Section 3.1.2). The distribution of cases constitutes an "extremely unbalanced dataset"[57]. Using this data distribution led to a different result analysis. According to the authors, values such as accuracy become less relevant as these metrics lose pertinence due to the dataset's imbalance. This type of data has also a concern: the impact of falsely labelling an account as a troll is much more considerable to the classifier than in the balanced datasets, therefore when measuring performance, the precision metric was prioritized by the authors. There are several studies with an imbalanced dataset approach (mimicking reality): Weller and Woo opt for a 80% normal users and 20% troll dataset [144], and in the study performed by Jansson and Oskar, negative example tweets were gathered from 150 users, with the positive examples coming from 4 (simulated) accounts, creating another imbalanced dataset.

3.2.5 Target

When performing the labelling of trolls, the Target is *what* is to be classified as troll or not: either the target is to classify the **Message** or the **User**. The studies that were found to focus on messages were found exclusively in forums. In these cases, the target was not the users themselves but the content of their messages. Of the studies that perform message classification, Mihaylov and Nakov's approach differs from the remaining studies as there are more than two classes: Firstly a troll message classification is performed, and after there is a *known troll* classification effort[89]: After distinguishing which comments were from trolls and which were from normal users, the researchers performed a second classification exercise, in which they distinguished which were *paid* trolls (entity sponsored), and which were *referenced* trolls (propagated paid trolls' messages). The studies that target the users also perform an analysis on the messages, but use them as features of a larger set.

The studies whose target differs from message/user are the ones that are specific detection efforts, focused on a set of particular attributes, as is the case with Dollberg et al. [33] or Dlala et al. [32]. These studies aim at validating a hypothesis that a certain set of attributes or formulas used in the data is relevant enough on its own to be sufficient to detect trolls. The results presented in them are sufficient to validate that premise for their specific conditions, but one critique for both cases is that they claim that for future work, the precision could improve by using message analysis related features, which makes these studies not a possibility of a full approach to troll detection, but for a useful dimension in the final model.

Both types of target objects showcase that for a more complete process of troll detection, message classification is a *fundamental* process. This applies to every instance where such is possible, even if not used not as the main feature group.

3.2.6 Label

A label, in machine learning, can be described as the output of the model, in this case, the result of the classification. In troll classifiers, the labels are predominantly binary: **Troll** or **Not Troll**, in some cases with different terminologies (e.g: Malignant, Benign [68]). There are exceptions to the above observation: In Mihaylov and Nakov's approach, there is a label for classifying users as Trolls or not, and then there is an effort in classifying whether those labelled positively are paid trolls or not[89]. In the previously referenced study by Galan-Garcia et al. [43] (section 3.2.1), their label, called *authorship percentage*, determines the likelihood of a certain profile being behind the cyberbullying account.

3.2.7 Type of Machine Learning Task

Although the emphasis of this survey is in classification, not all of the approaches found were **classification tasks**. Despite in the end being outputted a verdict on whether it was a troll message or user, not all of the solutions required algorithms of this category. Clustering is the task of grouping a set of objects in such a way that observations in the same group are more similar to each other than those of other groups [149]. This is a type of *unsupervised learning*, in which there are no defined labels beforehand. Applying this type of tasks in troll detection exercises is less popular, judging by the number of studies in the area that use it. One reason for this is that it may require a certain specificity in features that may be difficult to achieve without a high error margin. Nevertheless, there are cases where the problem has the right degree of specificity to have a clustering algorithm applied to it. Such is the case in Haque's study [50], where the (registered) users had to type the button or option they wanted pressed in the chat. Since the number of buttons and actions is limited, users could be divided into the ones that were typing allowed buttons and the ones that either weren't, or were constantly voting for the anarchy option available (effectively trying in a constant way to create disorder). By dividing them into clusters, the majority of non-conflict raising users would be automatically separated, allowing after for further operations on the users in the malicious clusters (no input of valid options, constant votes for anarchy). Another exception to the norm is the work of Kumar et al., where a graph mining approach is preferred to a traditional classification task. This study takes advantage of the functioning of the news forum *Slashdot*, and represents the users as nodes and the connections between them as edges of a graph [68]. They take this premise and attempt to detect trolls through the means of *node decluttering* (grouping edges with similar relationships) and graph simplifying. This approach did not generate a model, and recurred solely to an algorithm, created by them, with positive results in terms of accuracy and performance. The researchers also claim that their algorithm (called TIA algorithm) can detect trolls for any signed social network. While they were able to represent the connections because of an existing platform mapping users and their connections to a graph, called Slashdot Zoo, the usage of this approach in other social networks would require the representation of the gathered data as a graph. This can be time and resource consuming. Another critique is that the algorithm works for contained amounts of data. In a real world situation, full of connections and with large graphs, the alleged rapid performance would be affected by such a volume of data, therefore, in large graphs, the algorithm has the potential to become computationally expensive. This study proves as an exception to the norm, due to the fact that since its publishing in 2014, there was a shift in paradigm regarding text heavy tasks, with the type of approach becoming less frequent when compared to machine learning approaches, although both are of value.

3.2.8 Non Text Features

In the various works analyzed, it was verified that every solution contained features that were not extracted from the text produced by the user, albeit with varied degrees of importance. This type of feature does not have a consistent definition across the studies, and is here defined as all features not directly related or extracted from a user's messages. This definition covers a large number of features of several kinds:

- **Activity**: Some studies examine the user's behavior [21, 57, 89] and design features around it. Those include Time of comment; Day of the week; Number of posts per day; Number of posts per thread; Largest number of posts in a day; Time since creation of account.

- **Effect**: Since all studies focus on platforms where messages can be shared with audiences, this type of attributes are based on the community's response to the user's messages. These features rely on the premise that users react differently to troll messages than they do to normal ones [33, 57, 29]. Examples of such features are: Number of votes; Karma (ratio of positive to negative votes); Length of Deepest Comment Tree Branch; Total Number of Replies; Comment children count; Number of times a user is accused of being a troll; etc.

- **Chaos Theory** : Specific features derived from the chaos theory approach [32], such as Belief function score; Conflict measurement of messages; Total user conflict.

- **Clustering**: Features extracted applying clustering approaches (Joining users of same attributes in groups) [32, 50, 59], such as Number of clusters; Distance of the trolls to each other; Distance to other users.

- **Others**: Features with a high degree of specificity that don't have a clear category [33, 68, 43]. To this group belong features such as the centrality measures of the users' nodes; Geo-position; Type of client used.

In spite of the higher specificity of some of the features (there are community features which are not applicable to Twitter, for example), studies that use them display on average good results and the feature analysis efforts performed after the model is built demonstrate that they are relevant, in some cases even more than certain text features [33]. One example of this is Dollberg's research, which used a model with more non text features than text features. Beyond the positive results obtained (Section 3.2.12) It was verifiable that features such as the compression ratio of the messages had a better accuracy and f1-score (0.69 and 0.67) than the Word Count, a text feature, that had a score of 0.67 and 0.63 respectively.

3.2.9 Text Features

Text features play a fundamental role in the majority of studies gathered for troll detection. As opposed to non-textual ones, text features are all types of features that are extracted from text. Of the features gathered, they can be divided into three types of subfeatures: Superficial Text Features, Learned Semantic Features and Engineered Semantic Features.

- **Superficial Text Features**: This type of features comes from the extraction of knowledge from the text, but the type of information gathered is mostly about the attributes of the text, such as counters of certain types of words and ratios of those values. This type of features is the most prevalent type across the studies due to its range of possibilities [33, 21, 57, 29, 43, 88]. Such features include: Word count; Swear word count; Bad word to normal word ratio (how many swear words are in a user's messages compared to normal words); All capital word count; Bag of words - words and their frequencies; Number of stop-words, etc.

- **Learned Semantic Features**: A type of features that are obtained when using NLP (Section 2.4), these are extracted when applying functions specific for text categorization, such as creating embeddings from documents or obtaining sentiment scores. In these cases, new information is learned about the text and/or the relationships from the words in it [123, 89, 20, 144, 30]. This information relies on the algorithm/embedding/function used on the text, such as: Sentiment scores; Word2Vec distance between certain words; Readability Metrics (How easy to read is the text); Similarity to the news article being shared; BERT Model Outputs.

- **Engineered Semantic Features**: Features that are created specifically for the problem being addressed, which were created solely for the resolution of the researchers' problem. Such features were found in Seah et al.'s approach, where an embedding function was adapted to perform a sentiment analysis task on a specific, colloquial, language domain [123], or Dollberg's custom function to find the specific n-grams present in the text (BoW feature) [33].

The majority of studies use some text features, thus the representation of text is an important issue. The representations vary across platforms, with Word2Vec and BoW being used in more than half of all studies [123, 89, 20, 57]. The remaining approaches either used less popular representations, such as Tf-idf [29, 88] and Vector Space Model (a less common representation method with an algebraic background) [43, 30], or they used the most recent ones, such as BERT [144]. With new language representations appearing, and embeddings prevailing in NLP problems, it is expected that almost all future troll detection works that employ text features will contain modern text representation methods. This will create an imbalance in approaches with newer representations possessing the majority. From the works that employ these newer methods, it appears that these provide new ways to uncover undiscovered dimensions in the data, that were at this point not possible to find.

Since Troll Detection is an NLP problem, it is also expected that text features will always be prevalent in these types of studies.

3.2.10 Machine Learning Algorithms Used

Of all the studies found, the vast majority experiments with several algorithms, and chooses the one best suited for their specific problem, or the one that displays the best results. Since troll detection is viewed generally as a binary classification problem, algorithms suited best for this kind of task tend to be favored (Section 2.2.1). Aside from the algorithms detailed previously, the solutions using clustering favor the K-Means algorithm, one of the simplest, most popular clustering algorithms [66]. The output of this algorithm consists of K clusters (hence the name), grouped by one or more similar characteristics. This output format may be too simple to give the final verdict on the users, but may aid in the researchers' decision making process. This is because since K-Means is an unsupervised learning algorithm, it may group the users according to unrelated attributes, or may be too generic and have an unacceptable error margin.

Table 3.3: Algorithms Used Across the Troll Detection Studies

Algorithm	***Used in***
K-Means	[32] [50]
K-Nearest Neighbours	[59] [50]
Random Forest	[21] [43] [57] [144] [42]
Logistic Regression	[89] [21] [57]
Support Vector Machine	[123] [33] [29] [88] [30] [42]

Random Forest and SVM are the most popular algorithms, as is visible in Table 3.3. This is an explainable phenomenon, as both Random Forest and Support Vector Machines are documented to perform well for binary classification tasks (Section 2.2.5, Section 2.2.2). Many of the approaches for troll detection use a large number of features (the importance of each feature is later assessed (Section 3.2.12). To have a large number of features is preferred to an approach with a smaller number, since the studies use algorithms with mechanisms against overfitting and curse of dimensionality. The studies that employ this type of algorithms and with large numbers of features display good results, although on unrelated datasets [57, 29, 33, 88]. Thus, the us of the KNN algorithm is not advised, as it is sensible to overfitting and excessive features (Section 2.2.6). Empirical results confirm this hypothesis [29, 30, 59].

3.2.11 Decision

Once the dataset, features and algorithms are defined, there must be given a *verdict* on whether a user is a troll or not. Some approaches delegate that decision to the algorithm itself, which outputs one of two results and that is viewed as final [33, 89, 21, 57, 144, 30, 88]. Other solutions, however, set up their models and functions so that the verdict is decided by a percentage ("Trollness Percentage") [43] or a score (accounts with a score higher than X are considered trolls)[32].

In the message analysis classifier by Seah et al. [123], the prediction is given by a combination of the model outputs for Message, User and Thread Scores (same model): if two thirds of the scores are in the troll score interval, the message is marked as a troll. Another complex verdict method can be found in the study by Haque [50], where an anomaly score (distance to nearest neighbor and the sum of the distances to nearest neighbors) is calculated, and any data point with an anomaly score bigger than 40 was considered troll.

All the different approaches of the studies gathered (with the exception of Jansson et al. [59]) produced better-than-random results, which can be viewed as a positive trait, due to the fact that even the simplest methods can provide interesting and relevant results which aids in deterring trolls and their activity. One reservation that can arise from this data is that it verifies that there is no general strategy on how to approach this subject. This can be interpreted as a sign that more work in the field is required to reach a conclusion on what is the most effective method.

3.2.12 Feature Importance

It is still unclear what is the feature group that better identifies trolling, and if it is consistent across social networks or depends on the specific context. Due to that fact, in many of the existing studies, researchers start by getting the largest amount of features deemed as likely to be relevant and assess their relevance after. In some approaches, there are several models, and the features used in them range across almost all dimensions defined in the feature subsections (Sections 3.2.8 and 3.2.9).

Such a large and inclusive feature set is likely to contain irrelevant features. This approach is common across studies, and they display different methods of dealing with this adversity. Some cases, like Seah et al., experiment with various types of embeddings and language domain adaptations (the domain on which the sentiment analysis classifier is trained) and opt for the one with the best results[123].

In approaches that use SVM's, such as in Dollberg et al., knowledge extraction is done by running the algorithm with a single feature at a time, and its accuracy and F1-score compared with the same process for the remaining features, as well as the whole feature set [33]. The knowledge extraction process of this experiment allowed to assess the overall importance and relevance of each feature, having discovered that some features were irrelevant to the classification process. The knowledge extraction on the features proved the three most important ones were the **Users' Up/Down Ratio** (the study is applied on Reddit, where this value is called *Karma*), **Message Compression Ratio**, and**Message Word Count**, with 69%, 68% and 67% of precision, respectively. When using just these three features, compared to using all of them, the loss was just of 10 pp, which is a relevant finding. A critique for this work can be the lack of documentation of the classifier's results after removing the harmful features, as it is expected to have improved, and the fact that these findings base themselves on only one dataset (which may hamper generalization). Having these concerns in mind, potential useful knowledge on what to focus and what to ignore in future works was provided.

A more common approach across the studies, independently of the algorithm chosen, is to apply a logistic regression algorithm on the features and check the model's precision, accuracy, F1-score and AUC using just one of the features or groups of features. In the study of Im et al. [57], the precision obtained with the two best scoring algorithms tested, Logistic Regression and Decision trees, was of 94.2%, accuracy was 91.2% and AUC was of 0.93. Once the results were obtained, Logistic Regression was used not on each individual feature (since they were too many), but grouped by types, with each group's relevancy assessed: **Profile**, **Behavior**, **Language**, **Stop Words**, **BoW**. The task showed highly disparate results for every feature group, with several of them having a low relevance percentage when compared to others. Groups like the BoW features had a value of 58% of precision when used on their own, a value higher than random classification, but far from the 94% precision value obtained using the Decision Tree algorithm with all the feature groups together. The remaining features had an even lower score: Behavior features ranked 24% precision, Language ranked 15%, Stop Words 14% and the features extracted from the user's profile ranked just 7%. This allows to extract further knowledge for future work guidelines: Firstly, the features as a whole worked considerably better than the groups alone, showing that if an algorithm resistant to the curse of dimensionality is used, the excess of features, as long as they contain information, can be beneficial to the troll detection exercise. Another relevant conclusion to take from this exercise is the fact that the **Stop Word** group of features had some degree of relevance. This group of features was assembled based on a linguistics theory stating non-native English speakers use a higher number of stop-words [58, 94], and since most trolls are claimed to be from Russia (Section 3.1.2), this was thought as a possible relevant feature. The results prove that this feature was in fact relevant, and therefore the step of stop-word removal (Section 2.4.3) may possibly need to be rethinked for troll detection works.

In Mihaylov and Nakov's approach [89], every feature was subjected to Logistic Regression. All the features proved to be relevant, with the lowest scoring one being the measure of the **Bad Word Count** of each message, ranking 62.27% precision, still better than the baseline of 50%. In another study performed by Mihaylov [88], again all the features used proved to be relevant, being this author's work the only case where this is verified.

A different type of knowledge evaluation is performed on clustering approaches, where different clusters and clustering attributes are experimented [29, 50]. This includes testing the algorithms and distance among user observations with several distance measures, such as **Euclidean**, **Manhattan** and **Cosine** distance, with the Euclidean distance measure being the best for the researchers' problem.

Given the variety of features for this type of problems, and the fact that there is no rule regarding which of them should or shouldn't be used, an approach of gathering the largest amount of features possible and trim this number through knowledge extraction processes may be beneficial for future works in the field, as long as there is a concern regarding dimensionality and overfitting.

3.2.13 Evaluation Measure

All the models outputted in the various studies need to be subject to evaluation measures in order to evaluate their predictive performance (Section 2.3). For troll detection studies, the evaluation measures used have few variations across the studies, with Precision, Recall and F1-Score being predominant, with a fewer percentage of studies considering accuracy. This is justifiable for this type of studies, for accuracy is not the adequate metric when working with imbalanced datasets.

For the cases where the model is applied in a live environment, precision is not applicable, as there is no labelling done *a priori*, so the only existing verdict for the new data is the model's. Some authors deal with this adversity by using a wider variety of ways to evaluate their model. For Haque's [50] troll detection in the Twitch platform, test *datapoints* were inserted in his platform scanning tool. If the users' activities triggered the datapoint sensors, they were labelled as trolls and subject to measurements, such as the percentage of users flagged in comparison to normal users. The results were interpreted as successful due to that distribution being close to the projected ratio for trolls to normal users, which given the limitations of his environment, stands as a suitable measure.

Some studies opted for a mixed approach between human and machine: after the classifier gave the verdict on whether a user was a troll or not, three researches conducted a human analysis on the flagged accounts. Each of them voted whether the accounts the model analyzed were trolls or not. The human verdict was considered the majority vote for each user. With both verdicts given, the authors posteriorly analyzed the conflicting verdicts between human and model verification, in order to study the characteristics of the accounts which were not consensual [57]. Another way of evaluating the model is present in Dlala et al.'s [32] Chaos Theory approach. This time, since the solution is based on mathematical principles, the researchers tried to attest the validity of their solution through practical examples and demonstrations. One of these demonstrations was showing how the model distinguished what were normal users that shared troll messages and actual trolls that produced and/or shared troll messages, which can be interpreted as accuracy for this specific problem. On a different level, results that claim to be successes whose validation is based on specific cases, or do it without numerical measures, are more subject to critiques when reviewing them or searching for approaches. For future works, the Precision, Recall and F1-score values should be presented whenever possible, as the model/tool's effectiveness is easier to test and its results to replicate by researchers wishing to compare with their own results, or even to begin where they stopped.

3.2.14 Experimental Setup

There are several ways in which the result obtained from models can be assessed (Section 2.3.8). There were various types of measures used across the different studies.

Of the studies found, all but one claimed to be a success. To believe the majority of attempts tried are successes is a case of Survivorship Bias [125], as it is uncommon to find published papers regarding failed attempts in this specific field. The ranges of precision values differed across the studies, from Kumar et al.'s 51.04% to Im et al.'s 94.2%.

From the studies that use traditional evaluation measures (precision, recall and F1-score), K-fold cross validation is the predominant measure, with the main difference being the number of *K*. The studies with the smallest datasets [123, 33, 29, 43] are the ones where K is the lowest, in this case, k=5, 5-fold cross validation.

On the remaining studies that used K-fold cross validation [20, 57], both contain datasets comprised of millions of observations, and 10-fold cross validation was the preferred method, as is common in various studies of classification and other types. Their results were 80% and 94% precision accordingly.

Cross validation was observed to be one of the most used measures in the multiple studies, despite the different datasets, features, volumes and results, which may be considered a recommendation for future works of the like.

The remaining studies were each a specific case. All of them performed evaluation measures, but had to be adapted to their domain. The one that relied on mathematical expressions had them validated in a more human-centered way [32] through manual verification and assessment. The detection exercise that resorted to an algorithm relied on a mathematical validation for the hypothesis presented, and software testing techniques to assess the quality of the algorithm. In the case of Weller and Woo [144], their main performance measure was the Area Under the Curve-Receiver Operating Characteristics (AUC-ROC). Their dataset was also small enough that they could perform an additional, manual inspection of the model's output, which allowed them to pinpoint specific weaknesses of their approach.

The exception case is the only found study that explicitly claimed to had failed [59], and did not perform any type of cross validation. This happened due to after performing the plot of the users according to the K Nearest Neighbours algorithm, it was verified that the model showed no possible way of distinguishing trolls automatically. These results either affirm that trolls thrive because they mask their presence among other users, that their analysis had not been well conducted, or both.

The clustering approaches were also subjected to cross validation, in what proves that independently of the approach taken, cross validation is a popular evaluation measure amongst the different studies of this field. A characteristic of the studies that perform unique evaluation measures and experimental setups is that they are too specific for certain problems and are harder to reproduce. Some even omit details regarding the testing and implementation processes, leading to the conclusion that the solutions that deviate exceedingly make it harder to learn for other application areas.

3.3 Conclusions

In sum, although trolls keep trying to influence major events throughout the world, including in Portugal, it became clear that the scientific community and the population in general have noted their presence. A common set of behaviors were noted and documented over the years, which served not only as a common denominator for troll detection studies, but also made the users of the platforms aware on who can be a troll. This chapter surveys the state of the art of the studies performed on troll detection throughout the years and throughout the world, structuring it according to several dimensions. From the analysis of the several studies, the conclusion was reached that there are multiple types of approaches. Since the target platform of most studies are forums and social networks reliant on user interaction (e.g. Twitter, Reddit), it is normal for most approaches to converge. Such is verifiable in the types of approaches, where not only the users' text is used as features for the classifier, but also the metadata they generate in their profiles, through the reaction to their messages, etc. Once the data is collected, several models are trained, with the chosen algorithm belonging to the list of algorithms here presented (Section 2.2.1). The vast majority of these approaches obtain good results and set a baseline for where to start when performing a troll detection study. Besides these solutions, there were different types of detectors, with some favoring clustering, others mathematics, and even troll identity uncovering, all of them with positive results, proving there is more than one way of successfully detecting trolls. Some limitations can be appointed to this survey, such as the keywords used for the gathering of studies, which may missed similar works due to not having troll-related keywords, such as works describing misinformation campaigns in online platforms, which are not directly related to trolls but may be of relevance to the area. Another limitation of this work is the absence of studies of this kind adapted to the Portuguese context, which poses as the main motivation for the classifier to be developed, since no other tool of the kind was found, and there are evidences of the presence of trolls in the Portuguese Twitter base. Knowing the limitations of this state of the art, and the motivations for the Portuguese Troll classification study, the empirical study of this dissertation was then performed with the aim of being a baseline for Troll detection works in Portugal. With the project goals in mind, the background and core concepts regarding troll detection were now understood.

The next phase of the project would require a specification of the context and the environment in which the work would be performed, in order to combine and apply the acquired knowledge.

Chapter 4

TrollBus

This chapter details the extension created for SocialBus that allows it to perform Troll detection – **TrollBus**. Here are described not only the steps for the creation of the module itself, but the objectives that required to be achieved in order to develop it. These steps are the data collection process, the features engineered and the decision method utilized to label an account as troll or not. The description of these is fundamental in order to better describe the created platform and the work performed.

4.1 Implementation

The experiments performed in this dissertation and the knowledge acquired from them gave origin to models which were idealized not to be used in a static environment, but to persist and be applied to new data in a continuous manner. In order to fulfill this objective, a new module for SocialBus was designed, with its function being to label accounts as the data was streamed.

Achieving this objective would fulfill one of the motivations of the work of this dissertation, which involves creating the necessary tools for similar works in the area. As a solution, the extension developed adapted the models to SocialBus' streaming data, in order to perform real-time detection of accounts as their tweets were collected. This module was named **Trollbus**– *an extension of SocialBus for troll detection*. This tool gathers the data as it is being collected and classifies it in real time.

For the development and deployment of this application, a development process needed to be followed, along with the documentation and implementation of the program's scope, requirements and use cases.

4.1.1 System Requirements

Before any specific system design was performed, the first stage of implementation was to elaborate a set of high level requirements. These detailed what was expected of the software, ranging from its functionalities, connection to other applications and expected output.

Every requirement details a fundamental need for the program, in order to not only apply the learnt knowledge, but to make it available in the most versatile way possible given the experiment's characteristics and project goals [148]. A total of 4 key requirements were identified, and are present in Table 4.1. They detail what is expected of the TrollBus program, how it connects to SocialBus, how the classification of accounts is performed and how it presents the discovered knowledge.

Table 4.1: Trollbus System Requirements

Entity	***Description***
Accounts	• Based on the collected Tweets, the program should apply the model and classify the accounts' membership: Troll or Non Troll. • The program should create a user representation. If the user already exists in the program, his representation should be updated when a new Tweet from him is collected
SocialBus	• The program should connect to SocialBus, and classify each tweet and tweet data as its being outputted.
Output	• If an account's membership changes from Non Troll to Troll, the program should notify the user.

With the requirements for the program defined, the next stage was designing the use cases for it, based on these specifications.

4.1.2 Use Cases

At this stage of the project, the functionalities were established and documented. The next step of the process involved detailing the necessary series of interactions with the system in order to allow the fulfillment of the requirements explicit beforehand. In this case, in order to fulfill all the system's requirements, only one use case was needed – **Classify SocialBus Tweet**.

With this use case (Table 4.2), the total of the program's requirements are fulfilled. To aid in better exemplifying the program's behavior, an activity diagram was also elaborated, and is visible in Figure 4.1.

As seen in the figure, for a user's n tweets collected, he is evaluated n times. If more than one tweet of his is collected, his representation is updated $n-1$ times given a time t. The longer the

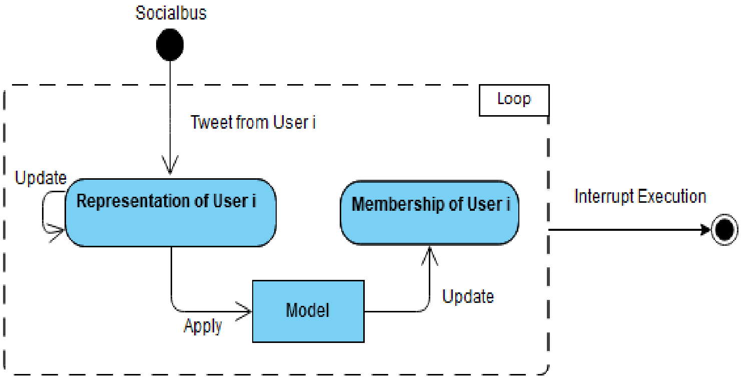


Figure 4.1: Activity Diagram of Trollbus' user labelling

execution time, the better the confidence in the labelling of the user, as the more the tweets of a user are gathered, the more accurate his representation is expected to be.

When a users' representation already exists and a new tweet from him is collected, his representation is calculated by the sum of each of the features, and the division by 2, updating the mean of each feature and updating the representation.

This formula to adapt the representation can be discussed. Using this calculation method, the user representation is assured to be based on the most recent tweets collected, implying changes in an account's behavior will be better represented in the account's membership. However, the usage of this method can be argued to emphasize the newly collected tweets too much. For future work, it is suggested comparing experiments using more than one representation update formula.

The TrollBus module was also added to the previously referred architecture diagram, visible in Figure 2.6. The updated architecture, along with the new component, is visible in Figure 4.2. This new diagram features the various processes of Trollbus including labelling a user, creating a user representation, and updating it as new data is collected.

4.1.3 Methodology

In order to deliver the Trollbus tool according to the time frames and functionalities described, an adoption of a methodology was needed to assure the final product would meet expectations. Due to the relative small size of the project, the work was focused on the developed use case. The implementation of TrollBus required a single sprint, where both the system and the functions required for the feature extraction were developed, along with the respective quality assurance processes (unit testing). As well as adding to the features and the functions to extract those features, a refactoring and improvement process was followed, in order to ameliorate both the readability of the code, as well as program performance. Since the labelling of accounts is performed in a live environment, the time required to label an account is a non discardable issue.

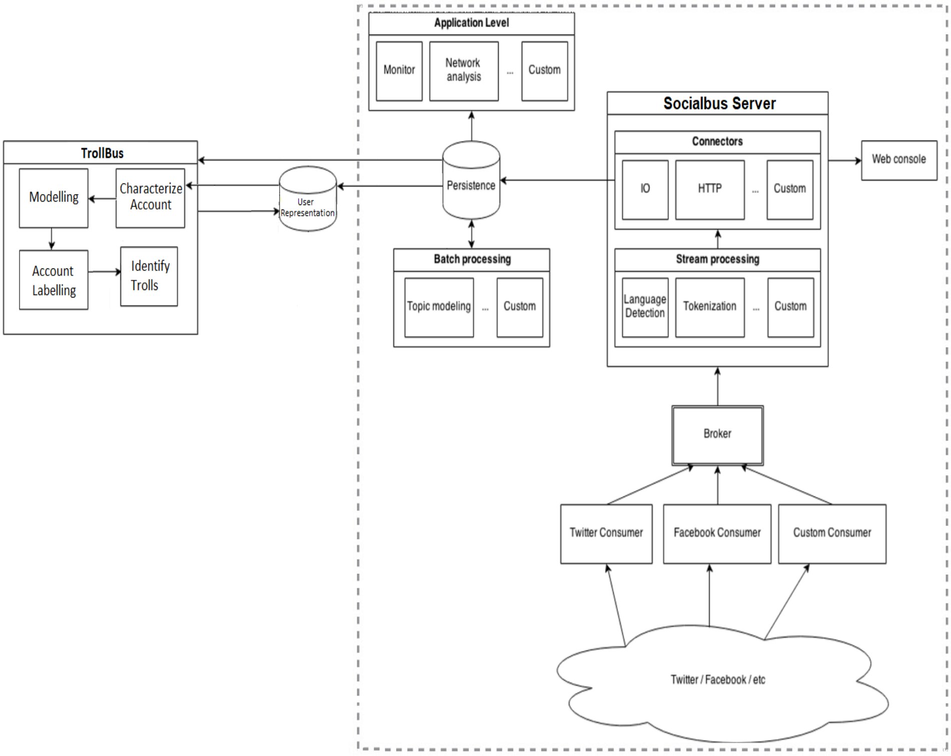


Figure 4.2: SocialBus architecture, updated with TrollBus

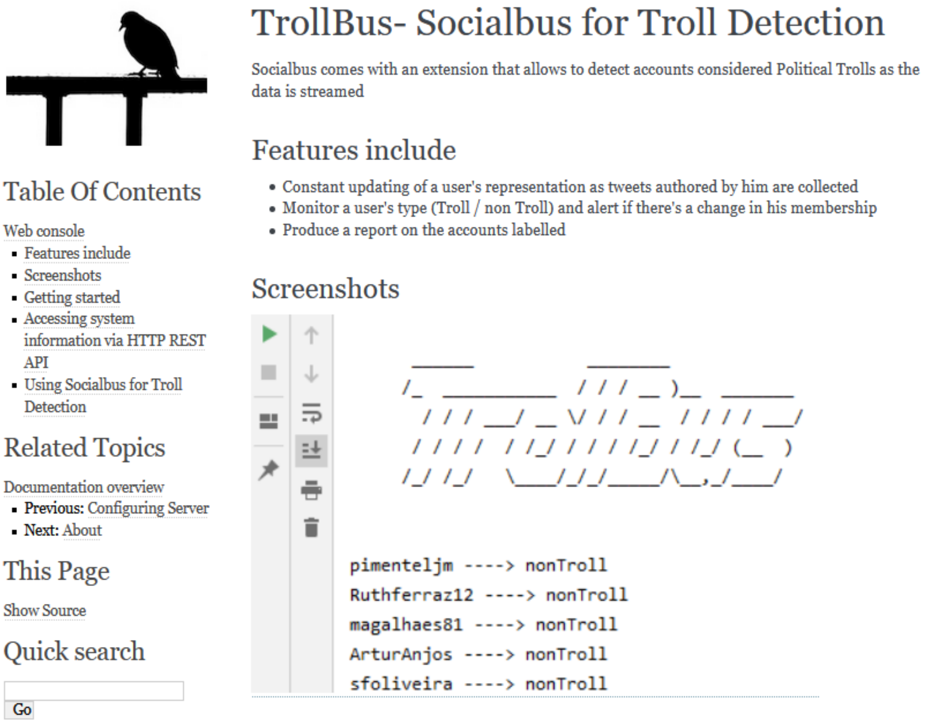
By following these processes, the quality of the final program was not only easier to assess, but any eventual improvements or changes were themselves possible without a significant learning curve.

4.1.4 Documentation

Once the development, deployment and quality assurance of the software tool was completed, a user manual was created and added to SocialBus' documentation. This application's guide consists of an HTML document, referring the features, options, usage guide and SocialBus history. A new page was added to this manual (Figure 4.3), describing TrollBus' features, how to install it, the extension's functionalities and expected output. This updated version of the tool was made available online and is available for download. This effort is considered to fulfill one of the motivations of this dissertation, which involves making available the knowledge gathered from this project, as well as incentivizing future works on the area.

4.2 Labelling Troll Accounts

For an application of not only TrollBus but the studied features and methods for troll detection in the Portuguese reality, several obstacles were present. Firstly, there was no data from the Portuguese Twitter with labels discriminated between trolls and non trolls, which meant that one



TrollBus- Socialbus for Troll Detection

Socialbus comes with an extension that allows to detect accounts considered Political Trolls as the data is streamed

Features include

- Constant updating of a user's representation as tweets authored by him are collected
- Monitor a user's type (Troll / non Troll) and alert if there's a change in his membership
- Produce a report on the accounts labelled

Screenshots

```
  ______           ____
 /_  __/________  / / / __ )__  _______
  / / / ___/ __ \/ / / __  / / / / ___/
 / / / /  / /_/ / / / /_/ / /_/ (__  )
/_/ /_/   \____/_/_/_____/\__,_/____/


pimenteljm ----> nonTroll
Ruthferraz12 ----> nonTroll
magalhaes81 ----> nonTroll
ArturAnjos ----> nonTroll
sfoliveira ----> nonTroll
```

Table Of Contents

Web console
- Features include
- Screenshots
- Getting started
- Accessing system information via HTTP REST API
- Using Socialbus for Troll Detection

Related Topics

Documentation overview
- **Previous:** Configuring Server
- **Next:** About

This Page

Show Source

Quick search

Figure 4.3: Section of the SocialBus manual page describing the Trollbus extension and how to use it

would have to be created. This solution, however, gave origin to another obstacle, which was that there was no objective definition of what constituted a troll for this specific segment of Twitter.

The proposed approach for the definition of trolls was generated based on a set of criteria used in the literature. These are mostly subjective, so in order to reduce the subjectivity in the labelling, a two-phase approach was defined to better characterize and determine the membership of these users. For the first step, each account would have a score computed for it, based on the criteria that was possible to be quantified. The higher the score, the more troll criteria a user fulfilled. After the score was calculated, the filtered accounts would be analyzed manually.

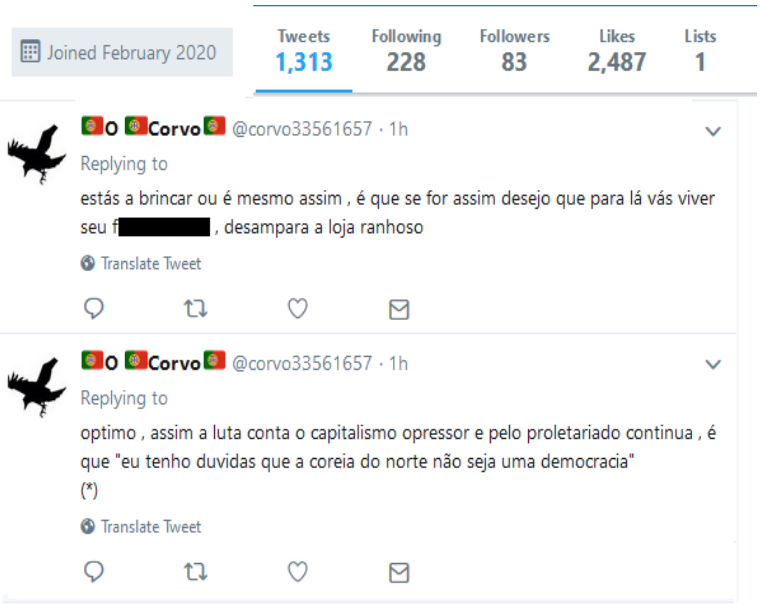
A total of 9 rules were established for a manual verification of an accounts' membership. These rules were created based on both an exploratory analysis of the suspected troll accounts found, as well as the existing literature describing trolls and troll behavior. When analyzing an account based on these guidelines, if a profile fulfilled at least 5 of them (more than half), he was considered a Troll. This approach was taken in order to reduce the subjectivity in the measuring, and also to minimize eventual false positives:

- **Exclusively Talking About Politics**: In order to be considered a troll, an account's tweets would have to restrain exclusively to politics. If in between discussing politics, the account under scrutiny discussed non political matters (e.g. sports, movies), it would not fulfill this requirement.

- **Usage of Insults**: A Troll, in order to provoke emotional responses, tends to insult other users, making this type of vocabulary prevalent in his tweets.
- **Sharing Fake News**: It is an observed fact that trolls may share news from disreputed or fraudulent news sources in order to reinforce or justify their opinion [124]. If a user shares information from such sources at least once, he is considered more likely to be a troll.
- **Following to Follower Ratio**: Malicious users have a tendency to follow a higher number of accounts, while having a much lower number of followers.
- **Account Creation Date**: Twitter Trolls in Portugal are a relatively recent phenomenon. Accounts created in the months preceding or following the Portuguese elections are more likely to be of trolls than the ones that are several years old.
- **Numbers in Handle**: A Twitter handle is the username that appears at the end of each users' unique Twitter URL (this is different from the name in profile). Exploratory data analysis revealed a possible trend in Trolls' handles, having a higher count of digits than the ones from normal accounts.
- **Name in Profile**: A user's profile name can be different from the handle, and unlike the latter, it does not need to be unique. Some users considered Trolls were observed to have names referring to politics, or non proper noun names in general.
- **Avatar Picture**: The majority of Trolls discovered through exploratory data analysis didn't have faces in their avatar picture. These ranged from cartoons, political symbols or political words. The most common avatar picture for Trolls was in a first analysis observed to be the Portuguese flag.
- **No References To Private Life or Other Social Media**: A Troll tends to take advantage of its anonymity. If a suspected Troll user has in his profile links to other (personal) social media or regularly posts photos of himself, he is significantly more likely to be considered a legitimate account.

In Figure 4.4, the user fulfills more than 5 of the labels in order to be considered a troll: He exclusively talks about politics, insults other users, his following count is almost triple of his followers, his profile name is not a proper noun, his handle displays 8 digits, his account is months old (at the time the analysis was conducted), and his avatar does not contain a face.

Not all accounts are as clear, however. Figure 4.5 displays a user that fulfilled several of the guidelines to be considered a troll: Frequent insults, a much higher following than follower count, the account is months old, speaks mostly about politics with a strong language and even retweets accounts that can be considered problematic (not in the picture). However, this same account has a normal name, appears in his handle picture and posts about his regular life, even appearing in those posts. This account was labelled as a non troll, as were all accounts of similar characteristics in order to restrict the positive cases to the ones where there was certainty about their membership.



Joined February 2020

Tweets	Following	Followers	Likes	Lists
1,313	228	83	2,487	1

🇵🇹O 🇵🇹Corvo🇵🇹 @corvo33561657 · 1h

Replying to

estás a brincar ou é mesmo assim , é que se for assim desejo que para lá vás viver seu f[redacted] , desampara a loja ranhoso

Translate Tweet

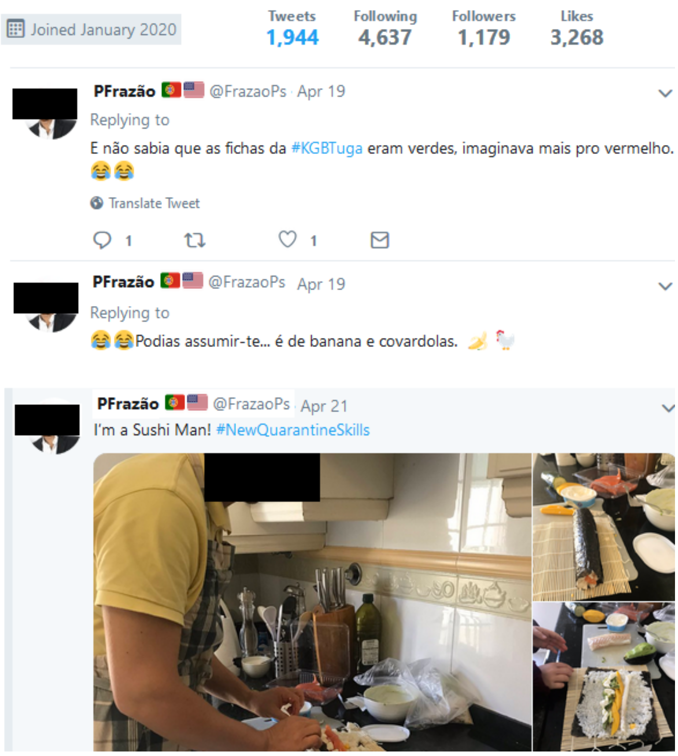
🇵🇹O 🇵🇹Corvo🇵🇹 @corvo33561657 · 1h

Replying to

optimo , assim a luta conta o capitalismo opressor e pelo proletariado continua , é que "eu tenho duvidas que a coreia do norte não seja uma democracia"
(*)

Translate Tweet

Figure 4.4: An example of a user that was labelled as a troll. First tweet: "Are you kidding me or is it really like this, because if it is so, I hope you go live there you ******, get off my face". Second Tweet: "Good, that way the fight against the oppressing capitalism and for the proletariat continues, because ”i have doubts that North Korea isn't a democracy”"



Joined January 2020

Tweets	Following	Followers	Likes
1,944	4,637	1,179	3,268

PFrazão 🇵🇹🇺🇸 @FrazaoPs Apr 19

Replying to

E não sabia que as fichas da #KGBTuga eram verdes, imaginava mais pro vermelho.
😂😂

Translate Tweet

1 1

PFrazão 🇵🇹🇺🇸 @FrazaoPs Apr 19

Replying to

😂😂Podias assumir-te... é de banana e covardolas. 🍌 🐔

PFrazão 🇵🇹🇺🇸 @FrazaoPs Apr 21

I'm a Sushi Man! #NewQuarantineSkills

Figure 4.5: An example of a user that was labelled as non Troll. First Tweet:"And I didn't know that the chips of the #KGBTUGA (*portuguese KGB*) were green, i'd imagine them more on the reddish side. Second Tweet: You could come forward... *(to not do so*) is of a banana (*idiot*) and a chicken"

This type of approach can be discussed in the sense that it is possible for trolls to adjust in order to not be detected, a known tactic employed by these users [5]. Additionally, since the dataset will be publicly available, labelling errors will be detected in the future.

4.2.1 Scoring Function

Based on the rules for considering a user a Troll (Section 4.2), a set of features were designed to be applied on the SocialBus data. The rules were translated into functions, where the compliance to each rule would assign him one point. Not all rules were translated into functions due to the inapplicability of translating some of them to code form in an objective way (such as the sharing personal information rule). If the final score of a user was above a certain threshold, he would then be flagged as suspect, and a manual verification would be performed, where the account would be evaluated based on all the rules. This step eliminates the set of accounts most unlikely to be trolls, and allows for a focus on the accounts that display the characteristics which make them more probable of being trolls.

A total of 6 functions were developed for the scoring function, visible in Annex A.2. A user's potential maximum score was of 6. This was highly improbable, however, so the minimum score required to signal an account for further investigation was set at 4.

Of the functions that generate an account's score, the verification on whether its profile picture contains a face can be debatable. This approach can be prone to false negatives, for example, through the usage of deepfakes (computer generated human faces) or even using photographs found online. Despite that, since there is a record of trolls presenting their profiles as entities and not users (e.g. right wing news, liberal activists) [5], and these not having faces in their avatars, this feature was considered relevant. The function **has_normal_name** verified whether a user had an identifiable first name in his profile. If a user didn't match any name on a list of Proper Noun names(as the ones found in a country's Civil Registry), the function would add 1 to his score, as it was more likely that his profile either wasn't a personal one, used a fictitious name, or was a page. This function used in itself would not allow to take many conclusions, as it was prone to nicknames, misspells or foreign names. It was, however deemed as possibly relevant when used together with the remaining functions.

4.3 Data Collection

As previously referred, the data was collected using SocialBus (Section 2.1.3). This tool allows for two types of methods for the collection of tweets:

- **Filtering by Keywords**: SocialBus takes as parameter a text file, in which a group of keywords is specified. While the program is running, it returns all the tweets produced containing the specified keywords, created in the time frame the program is active.

- **Filtering by Users**: SocialBus takes as parameter a text file, in which a group of user ids is specified. While the program is running, it returns all the tweets produced by the users with the specified id's, and the tweets of all the users who interacted (retweeted, replied) with the specified accounts in the time frame the program is active.

There are advantages and disadvantages associated with both options.

Filtering by keywords allows for a more specific tweet search, where if the keyword base is broad enough, it returns information highly probable to be about national politics. However, either the search is restricted to highly specific terms, such as *António Costa* (Portuguese prime minister), *Marcelo Rebelo de Sousa* (Portuguese President), etc., or the results will not be relevant, since the search for broader terms associated with politics, such as *Governo* (Government), *Oposição* (Opposition), *Presidente da República* (President), etc. returns a much greater proportion of tweets from Brazil, referring Brazilian politics. Another downside in searching by keywords comes from the volatile nature of Twitter itself, where every week there are different topics trending, some of them political, and if those topics are not in the keyword list, there is the risk of possibly relevant information being missed.

When filtering by users, the search can be more robust to the different trends and topics that arise over time, and by having a userbase consisting only of Portuguese users, the Brazilian users' tweets would not overwhelm the Portuguese ones. The adversities that come with this approach regard the *representativity* of the sample: What group of users would constitute a representative sample of the Portuguese Twittersphere? Also, in order to find positive cases (tweets from trolls), it was necessary a group of users with whom the trolls regularly interacted, with sufficient dimension to be considered representative of the Portuguese Twitter.

The approach chosen was the filtering by Users, where the user sample would consist of the political journalists in the biggest newspapers and television networks in the country, as well as every national user they followed. This decision was based on two key assumptions: Previous analysis of the trolls' behavior show they are prone to comment on popular journalists' tweets in order to reach a broader audience [57], and the second assumption relates to the study performed by Ghosh et al., where it is observed that a sample gathered from expert accounts on a subject is not only richer in information content, but is also more thrustworthy, more diverse and capture breaking news stories earlier than random sampling [47].

4.4 Account Seed

In the context of this project, a seed is the base set of accounts from which all other user ids for the data collection are gathered – the core set of accounts from which the bigger set (to be used in SocialBus) will be generated. Firstly, as the data collection was decided to be based on accounts, a search (mainly subjective) was performed regarding trustworthy entities, the people believed to be of relevance for the project.

The entity search was performed through an identification of the major news outlets in the country, along with a collection of the political journalists found in those organs.

Applied to this project, the media organs to search were limited to the most sold Portuguese newspapers [37, 75], as well as the primary news agencies and channels (Table A.1).

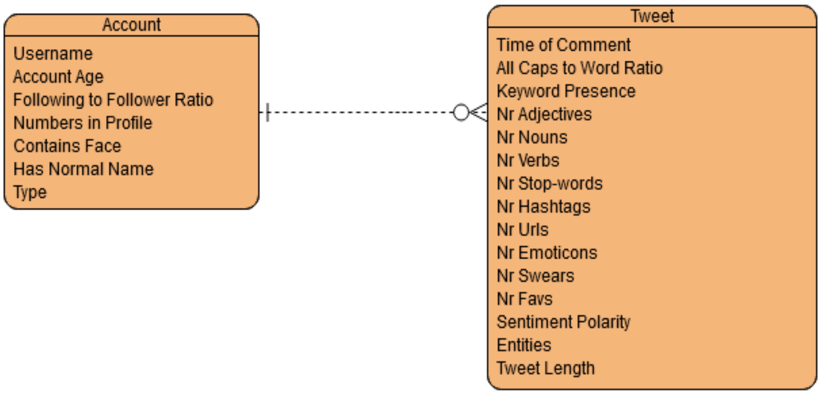


Figure 4.6: Database schema for the data collected from Socialbus, adapted for TrollBus

Once all the names were collected, these were searched on Twitter.

The set of accounts identified from this seed is then filtered automatically based on application specific constrains (e.g. language).

4.5 Data Preparation

The information outputted by this SocialBus encompasses several types of information regarding both the tweets and the account which published them. In order to better perform a troll detection exercise, however, SocialBus' output had to be treated in order to create the necessary features which would be used for classification. As such, a new extension was developed for SocialBus, which supported the data collection for TrollBus.

The schema for the data can be seen in Figure 4.6. For every collected tweet of a given account, the data is divided into account and post.

Using the works from the state of the art as reference, the gathered information was divided into 3 main feature sets: **Account Features**, **Post Features** and **Embedding Features**. The features in the sets were designed from a combination of the features and approaches found in the existing studies (described in Sections 3.2.8 and 3.2.9), with some original features that aroused from the study and observation of Portuguese accounts considered suspicious. These sets will be detailed in the subsections below.

4.5.1 Account Features

This feature set is comprised of the account metadata. They do not focus on the content (tweets) produced by a user but the data gathered from its profile. The data for these features was obtained from the information relative to the accounts returned by SocialBus. This data was treated and turned into features, with some original features added also, inspired from an exploratory study of collected data, believed to better suit the Portuguese Twittersphere. A total of 7 features were designed, present in Table 4.3 along with the studies they were adapted from.

This set has the particularity of not containing any features relative to the tweets' text. Along with the features adapted from the literature, two original ones were created: the verification of whether a user's profile name is present in the Portuguese Civil Registry and if his avatar contains a face. These were designed based on the analysis of accounts gathered through an initial study of the Portuguese Twitter communities. The features **Contains Face** and **Has Normal Name** are binary, which are either 1 or 0. In both cases, the negative example (doesn't contain face or normal name) is equal to 1, as the features were extracted from the scoring function. After considering whether the account's age should be the date of creation or an integer of the number of years passed since the date of its creation, it was decided that the latter would be more advantageous, as it would not lose pertinence over time (e.g. A model that asserted that accounts created in 2020 were more probable to be trolls would lose pertinence over the years).

Contrary to the post feature set, the features in this set are not expected to change from tweet to tweet. This condition could (although less likely) change over time, however, so for every tweet of the user found, its account features would be the mode of these attributes.

4.5.2 Post Features

The features in this set refer to the attributes of the tweets' text that can be quantified by integer or real values. These features were adapted or inspired by several works in the literature, with each feature and its inspiration present in Table 4.4. In order to use the features in this group to classify each account, the attributes of Username and Type were added to the set, in order to identify each user.

The high amount of features demonstrates that this specific group of features is one of the most recurring ones (also because of how broad this group is). Not all features mentioned in the state of the art were included due to SocialBus and data collection limitations. Such discarded features include the measurement of how many replies were received by a tweet or how many times a user is referred in those replies, as the data is collected in the moment the tweets are published. There were also features relative to tweet metadata, which didn't suit any of the defined feature sets and were added to the post features set due to, along with the remaining features, describing the tweet itself.

The total number of features generated for this set was 18, with each one of them detailed in Table 4.4.

Of the features in the table, Time of comment, Number of hashtags, Mentions, Emojis, Urls, Favorites and Tweet Length were extracted directly from SocialBus, not requiring any kind of treatment. Of the features engineered, All Caps to Word Ratio and Keyword Presence required standard Python list operations.

In the feature Nr Stopwords, the list of stop-words and the function to list them was imported from the library *NLTK* [95], one of the most popular Python libraries for works with natural language data, possessing several tools for the Portuguese language.

The functions that counted the number of Adjectives, Nouns, Verbs and Entities were imported from *spaCy* [127]: a Python library designed for natural language processing, offering several types of tools including named entity recognition, pre-trained text models and text classifiers, all of which claimed to have good results. SpaCy's language model, which was used to label the words according to their type, was trained on a combination of news articles from *Público* and *Folha de São Paulo* [128]. It contains 500 thousand keys and 20 thousand word vectors, with 300 different dimensions. A language model of such volume and diversity was not possible to create in the scope of this dissertation, therefore, this model was considered of quality for the work to perform.

The Number of Swears feature was calculated by checking if any word was equal to the elements of a previously generated list of swearwords. In the case of this project, the list was adapted from the work of Gustavo Laboreiro [69], which contained a list of Portuguese profanity. The list used was adapted from the original one, limiting it to the ones signaled by the author as the ones with the strongest connotation, originating a list of 25 swearwords.

The terms in the feature Keyword Presence (Section A.1) were created from a collection of political insults. For the case of this project, the list consisted of political insults in the Portuguese language. This insult list was created from an adaptation of the work of Winberg [146], describing the insults used in the United States' politics, a set of different Portuguese articles referencing polarizing political speech and social media insults in Portugal [9, 26, 39], and the found synonyms of those insults in the Portuguese Priberam Dictionary [107]. Some of the insults on the list were not exclusive to politics but are noted to be used with a political connotation. Some of the words in the list are not considered offensive outside a political context, but are documented to be used in an insulting matter in a political speech.

Finally, the Sentiment Polarity scores refer to the sentiment analysis scores of a user's tweets. In the case of this dissertation, the scores were calculated by using the *SentiLex-PT* project, a sentiment lexicon designed for the extraction of sentiment and opinion about human entities in Portuguese texts [17].

All these features were applied to each collected tweet. This implied that as each tweet was a row in the feature set, an account could have several rows. In order to limit each account to one row, every user had one entry in the feature set, with every feature being the mean value of attributes for all the tweets gathered of that account.

4.5.3 Embedding Features

The embedding feature set is comprised of two columns: the tweet's text and the tweet's Type.

One of the most important steps at this stage was the choice of representation to use, in order to create the group of features that was learned from the text itself. Several options were considered, such as BoW, Tf-Idf or Word Embeddings (These representation are detailed in Section 2.4.6).

Both BoW and Tf-Idf had a record of being used in other troll detection approaches, where they not only displayed positive results, but also existing cases where they are by far the feature group with the best results in terms of precision and recall [57].

The usage of word embeddings, despite having less expression in the existing troll detection studies, is predominant in NLP works in recent years, and has also been appearing in troll detection works as well (e.g. by the work of Weller [144]). Although the works using this type of representation display positive results in the metrics of AUC, precision and recall, the approaches using BoW remain as the ones that display the best results for these metrics.

It was decided that the text would be represented using word embeddings (Section 2.4.9). This choice was made due to the literature in both troll detection [123, 144] and other NLP works [65] demonstrating that the usage of this kind of representation allows to uncover information which was not obtainable using other methods. The potential for uncovering useful relationships in the users' text in a way that could not be done manually was a factor deemed important. Following the progress being done in the NLP field of research, the usage of word embeddings was therefore considered pertinent.

Once this was established, there were several options regarding the way the representations would be used, and how a text model would be created: train a text model consisting solely on the tweets' vocabulary (created from the tweets) or create the representation using an existing language model. A limitation to consider when selecting from the two was the fact that there are fewer language models for the Portuguese language than for English, for example, and even less applied to European Portuguese. After a research effort on the existing language models for Portuguese, the model offered by *spaCy*, a library already used in the post features set, was considered the best one given the circumstances. These models allowed to predict named entities, part-of-speech tags and syntactic dependencies, but also contained a representation based on word embeddings with pretrained weights, designed to be versatile and suitable for existing models, in order to achieve better accuracy [127]. The final approach chosen was to experiment with spaCy's language model. The data volume was considered not voluminous enough in order to create a custom representation, as well the fact that there were limited computational resources. The usage of spacy also allowed the application of these features along with the numerical data, as the library used (scikit-learn) required the word vectors to be adapted: each tweet would be treated as a *document*, and a document embedding exercise was performed. With this technique, every tweet was turned into an array of 5 numeric values. The length of the array was chosen as this is the default length of the library used. An increase in the vector size, due to the number of tweets, reached calculation times which were not feasible given the existing computational resources, making the vector size one of the limitations of this work. For future work, if the resources are available, it is recommended to experiment with higher vector sizes. These values were split into 5 columns and added to the dataset.

4.5.4 Features Based on Multiple Posts

Once the representation and model to use were selected, there were still several options regarding the way the accounts could be categorized regarding their tweets. Due to the relational nature of the database where the tweets were stored (Figure 4.6), one account could possess several tweets associated to it. As an account only has one verdict (Troll/ non Troll), a mechanism to deal with this condition was required in order to classify an account based on the collected text. Two separate ways for categorization were created, both of them using different processes:

- **Classification Tweet by Tweet**: Each tweet of the account is analyzed separately, with the account having several verdicts, one for each tweet.
- **Classification of All Tweets as One**: All of an account's tweets are joined into one bigger tweet, and the analysis is performed just on that tweet.

Both approaches would have to be considered, although the implications for the usage of each of the sets varied greatly: The classification of an account's type based on each of its tweets requires each tweet to be labelled according to its user, meaning an account's membership is determined several times, with the number of verdicts for that account equal to the number of collected tweets authored by it. By using this method, there could be implications with smaller tweets, as simple answers (e.g. a tweet only consisting of an ”OK”) by trolls could create inaccuracies in the model. The total of the verdicts for that user would be aggregated, and the final decision would require an additional process to calculate the verdict from the given group of predictions.

Classifying accounts once, based on all their tweets joined together as one bigger tweet, would be a more direct classification task, in which a user would be classified once. This would bring a different set of adversities however, such as an unpredictable result on shorter tweets.

The definition of all feature groups allows to answer the second research question, although it is evident that there are several other possible answers. In the case of this dissertation, however, the features were adapted to the data collection platform, as well as the Portuguese language, therefore this answer is the one most dependent on the data collection process.

4.5.5 Estimating Embedding Features performance

When analyzing the data collected and the possible methods of classifying the accounts' text, two candidate ways of using text classification were then asserted:

- Joining all of the user's tweets together into one condensed tweet.
- Classifying each of the user's tweets individually and asserting his membership through those classifications

Using all the tweets collected, their text is condensed into a single (bigger) tweet and from the classification of that set of texts, a verdict regarding the user's membership is outputted. This approach does not require additional methods, but may lead to difficulties in the model when analyzing new, smaller, tweets.

The alternative to this method classifies each of the user's collected tweets, making the account have several verdicts regarding its membership. This second study requires an additional method to determine the type of account. By joining all the tweets together, it was considered the possibility of the model performing worse when classifying smaller tweets. In this second approach, where the models learn patterns of troll behavior based on individual tweets, that possibility is mitigated. Two aggregation rules were proposed. These reflect the strategies devised to transform the set of predictions for the tweets of an account into a single aggregated prediction:

- **Classify each tweet of an account** – if the mode of the account's tweets had a score above the best threshold and were considered troll, the user would be labelled as such.
- **Calculate the mean of each account's scores** – All the account's scores would be summed, if their mean was higher or equal than the best threshold, it would be labelled as troll.

In the case of an account possessing the same amount of troll and non troll messages, he would be considered a troll, as false positives are not considered as important as false negatives for this work.

With both the SocialBus extension and the processes needed to execute it defined, an empirical study was then performed.

Table 4.2: Specification of Use Case Classify SocialBus Tweet

Use Case Name	**Classify SocialBus Tweets**
Actor	System (Trollbus)
Description	The System receives the SocialBus output, creates a representation of the user, applies the model and classifies the user as Troll or Non Troll
Pre-Condition	SocialBus Twitter consumer is running
Typical Course of Events	1. SocialBus collects a tweet and saves it 2. The system reads the file where the tweet is saved 3. The system reads the tweet and tweet data 4. The system creates a representation for the user who wrote the tweet 5. The system applies the model on the user's representation 6. The system saves the user's representation and verdict 7. The system classifies users until SocialBus' execution is stopped 8. The system generates a report on which accounts were marked as Trolls and which accounts were classified in total
Alternate Courses	1. SocialBus collects a tweet and saves it 2. The system reads the file where the tweet is saved 3. The system reads the tweet json 4. The system verifies that there is already a representation created for that user 5. The system updates the user's representation 6. The system applies the model on the updated representation
Post Condition	The system saved a summary file in the program's directory
Implementation Constraints and Specifications	If a user's representation changes from Non Troll to Troll, the system should display a console message warning of such.
Assumptions	The System shall run indefinitely as long as SocialBus is executing.

Table 4.3: Account Features

Feature	***Definition***
Username	User's handle
Account Age	Years since the account's creation date [57, 30]
Following to follower Ratio	Following ÷ Followers [57]
Numbers in Profile	Number of digits in the handle [42, 29]
Contains Face	Whether the avatar picture contains a face
Has Normal Name	Whether the user's name is present in the civil registry name list
Type	Troll or non Troll

Table 4.4: post features

Feature	***Definition***
Username	User's handle
Time of Comment	Timestamp with format HH:MM:SS [57, 42, 43, 88]
All Caps to Word Ratio	Number of words in all caps divided by total number of words in the tweet [33]
Keyword Presence	Number of political words and insults in the tweets [42]
Nr Adjectives	Adjective count [42, 29, 30]
Nr Nouns	Noun count (proper and common) [42, 29, 30]
Nr Verbs	Verb count [42, 29, 30]
Nr Stop-words	Portuguese stop-word count [57]
Nr Hashtags	Hashtag count [57]
Nr Mentions	Mention count [89]
Nr Urls	External link count [57]
Nr Emoticons	Emoticon count [89]
Nr Swears	Portuguese swear count [89, 33]
Nr Favs	Number of times a tweet was favorited at the time of collection [57, 88]
Sentiment Polarity	Portuguese sentiment analysis score [89, 123, 59, 29]
Entities	Number of Portuguese entities referred [89]
Tweet Length	Number of characters in the tweet [33]
Type	Troll or non Troll

Chapter 5

Empirical Study

This chapter presents the conditions used in the implementation of the experiments performed, as well as the evaluation of their results. The data collection and troll annotations are described, along with the experimental setup and exploratory data analysis findings. Of the experiments here detailed, the algorithm parameters and the different combinations tested during the experiments are specified. Finally, the resources and technologies used for them are explained.

5.1 Data

The main goals of this project are to determine the most adequate features to detect trolls and to verify if the methods used for other languages also apply to the Portuguese universe. To fulfill them, the required data would consist of tweets from the Portuguese Twittersphere, referring to politics. In order to find suspicious accounts amongst the recovered data, it has to be extracted from the segment of Twitter where these accounts proliferate. Collecting this type of data is not a trivial matter, as the filters used for the collection not only need to limit the data to political themed tweets, but do it in a way that is representative of the reality of the Portuguese universe, and *only* to it. Both objectives present as challenges: There is the risk of a non representative data collection– this can happen either by missing important elements of what is being discussed through a faulty collection process, or by having too much noise in the data. The noise risk occurs as the data collection is prone to non relevant information, since there is the risk of the collected data to be overrun by tweets from the Brazilian Twittersphere, which is the biggest Portuguese speaking community on Twitter [24].

As referred in Section 4.3, a method for achieving this type of representativity in data was created for this experiment. In this particular case, the expert accounts on Portuguese politics were assumed to be the political journalists.

5.1.1 Account Base

As referred in Section 4.4, the rules for the base set of accounts which would be the basis for entire user list were defined, and were limited to the political journalists of high profile newspapers in

Portugal.

Since Twitter doesn't possess an office in Portugal [138], the majority of high profile users doesn't have verified accounts. Knowing this limitation, an alternative method to check if the account was legitimate was followed: If the profile matched the journalist's name and was followed by other journalists (either verified or assumed to be real through this method), it was assumed to be the real account. Other factors were also indicative that the account was real, such as a high follower number.

Table 5.1: News Media and Journalist accounts found

Media Organizations	***Nº of Journalists Found***
SIC	14
Expresso	14
Observador	10
Público	8
Jornal de Notícias	5
TVI	5
Agência Lusa	4
Diário de Notícias	3
Correio da Manhã	1

A total of 67 relevant accounts were found. An interesting aspect of this account collection is the fact that *Correio da Manhã*, which in 2019 was the most sold newspaper in Portugal [75], has few identifiable editors or political journalists on Twitter.

The sum of all unique accounts followed by the gathered journalists was of 32 153 users.

The set of accounts identified is then automatically filtered, based on application specific constraints (e.g. language).

5.1.2 Filtering Portuguese Users

Once all the unique users were obtained, a process was needed in order to separate the non Portuguese speaking users from the Portuguese ones. This decision was made through the usage of two python libraries: **GetOldTweets3** and **Langdetect**.

Get Old Tweets, currently in its third release, is a library that allows to retrieve a specified users' past tweets, which, according to the author, bypasses some limitations of the Twitter API [108]. Langdetect is a library that given a text or document, determines the language it is written in [110].

The approach taken to determine if a user was of interest to the seed consisted in analyzing his 10 latest tweets. If a minimum of 3 of them were detected as Portuguese, the user would be added to the relevant accounts list.

The past tweets number was limited to 10 due to computational reasons – the higher the number of tweets, the more time it would take to retrieve all the tweets from that user. The minimum number of tweets required to be in Portuguese was limited to three for two main reasons:

- Library Limitations: Due to the informal nature of Twitter, the tweets' text contain a vast number of colloquial expressions, such as interjections and onomatopoeias. These can cause the library to misinterpret these words and assume the text is in a different language than the one it is.

- Portuguese Twitter Characteristics: It was observed that the majority of Portuguese users tweeted in more than one language, with the predominant secondary language being English. While it is common for Portuguese users to tweet in Portuguese and English, the same does not apply to the opposite case. Therefore, if the number of tweets in Portuguese was at least 3, the user was considered relevant for the seed. If the user was in fact Portuguese, but did not meet the 3 tweet minimum criteria, he was considered a non relevant user for this task.

Once all the irrelevant users were filtered, the total number of accounts selected was of 9022 users.

On the filtered list, a manual verification exercise was then performed, aiming to assess whether the sample contained every account in the seed set and if the process had allowed too many false positives.

The initial method used to analyze the distribution of tweets included searching for the accounts recovered on Twitter, in an effort to understand how representative had been the collection exercise. It was verified that of the accounts marked as relevant, all major political figures (government and opposition members, parties' official accounts) were present, as well the total of the journalists used for the seed. Of the non journalist and non politician accounts, profiles with a high degree of influence over the Portuguese Twitterspehere (high follower count, many replies and comments on every tweet) were also present. These accounts' behavior was observed to consist on commenting on daily events which included, but were non exclusive to, political matters. By having these accounts in the collection, the presence of tweets written in a more informal language was assured. This was considered beneficial, in order to not restrict tweets regarded as non troll to the ones with a more formal language, present in most politicians and parties' tweets.

Despite the filtering techniques, there were still Portuguese speaking accounts considered irrelevant in the obtained list. A manual filtering of the accounts was performed based on the tweet location, removing the ones outside of Portugal. This was based on the assumption that removing users by this method would reduce irrelevant users in a higher proportion than potentially accounts account of interest would be discarded. The users that were removed consisted mainly of Brazilian accounts, although there were also Galician users (Galician is a language with similar origins to Portuguese [102] and was marked as so by langdetect). Some Portuguese accounts were also removed, as these were dedicated exclusively to sports and sporting news, and were considered non-relevant for the exercise. The final user list was composed of 7723 ids.

Table 5.2: Score distribution and analysis

Score	***Total Found***	***Trolls Found** (n=35)*	***Percentage***
4	35	18	51%
3	937	9	26%
2	6 428	2	6%
1	13 463	2	6%
0	7 376	1	3%

Once the list of ids for SocialBus was defined, the program was executed, with the filtering returning tweets from the selected accounts and the ones that interacted with them. The tweets outputted from this method were of a mean of 6500 tweets per hour in the busiest hours, which were discovered to be from 11-14 and from 18-23.

5.2 Troll Annotation

As this dissertation regards techniques for detecting trolls, there needs to be a method for labelling a user. As the definition on what constitutes a Portuguese troll has already been defined (Section 4.2) and a function has been developed to aid in finding users that fulfill this definition (Section 4.2.1), the annotation method will be described in the subsections below. A formal annotation method of this type has never been performed in Portugal, and as such is hoped to be a potential baseline for future works in the area (stemming or not from this project).

5.2.1 Anotating Portuguese Trolls

With the score function's results deemed representative enough to use as an account signaling tool (Section 6.1), the same exercise was performed on a sample of data collected through SocialBus. The sample consisted of 28 239 accounts and the function was applied to all of them and their respective scores were annotated. If the score function was appropriate for the Portuguese accounts, it would mean that the higher the score, the higher the probability of that user being a troll. Of all the accounts, 35 of them had a maximum score of 4. In order to keep the uniformity of the samples, 35 accounts were selected at random for each score, and were evaluated (Table 5.2).

The first conclusions taken upon inspecting the score distribution was that the number of accounts with a score of 3 or higher was considerably lower than the remaining ones. Of the accounts found to have the highest score, none of them displayed the two highest scores possible, and of all the accounts, only 0.12% of them had a score of 4. Of the accounts with the highest score, more than half of them were considered trolls. An analysis of the remaining 49% not considered troll revealed the scoring function was sensitive to personal accounts created from 2020 onwards, which in this case had, for example, nicknames (no known name) and numbers in their profile (e.g. if a user has his birth year in the handle, he was guaranteed at least a score of 1). Even from a relatively small sample, there were strong evidences that the higher the score, the higher the probability of a user being a troll. It was also demonstrated that a single function could not dictate the users' type, but together they were important in signaling suspicious accounts. With positive results on both Russian and Portuguese troll experiments, the score function was considered suitable for the notation of positive cases. These results indicate that the Portuguese trolls share similarities with the trolls documented in the state of the art.

5.2.2 Class Distribution

With the method to retrieve and label accounts already defined (4), the data collection was done in two steps: The first step served essentially to test the approach and tune its parameters as necessary, while the second approach was more extensive and allowed to retrieve a bigger portion of the data.

At the end of this second collection, a total of 268 accounts were labelled as trolls. An analysis on the first 30 accounts flagged, 2 months after their first labelling, shows that 7 of them (23%) had been banned. According to previous studies, bans are one of the best signs that an account is in fact a troll [33], although this feature can only be asserted *a posteriori* and is not applicable in exercises of classification in real time. With the collected tweets, a dataset was assembled consisting of 99 727 tweets from regular users and 4016 troll tweets. After grouping the tweets by account, the dataset was comprised of Tweets from 20 489 different normal users and 268 different trolls, which consisted of a proportion of approximately 1 troll account to 76 normal accounts, and 1 troll tweet to 24 normal tweets.

5.3 Experimental Setup

In this section, the steps taken for the machine learning approach for troll detection are detailed. At this stage, the data has been collected in an acceptable amount, and the manual labelling has been performed. Here is described the processes applied to the data, whether in terms of imbalance treatment, or algorithms applied. Finally, the estimation methodology of the applied algorithms is also detailed.

5.3.1 Validating the Scoring Function

A validation method for the scoring function judged to be suited for this type of task would be to use a *Gold Standard*– the diagnostic test or benchmark that is the best available under reasonable conditions [142]. In this case, the most appropriate Gold Standard would be to use the scoring function on a list of accounts, created by a certified entity, containing accounts of Portuguese users considered to be trolls. Due to factors such as trolls being a recent phenomenon in Portugal, and the lack of studies on this subject, such a list does not exist. There are, however, several lists of confirmed Russian Trolls published by Twitter and other renowned entities, all containing users known to have interfered in the 2016 US General elections (Section 3.1.2). The approach chosen was to use the scoring function on a dataset of troll users and tweets released by NBC [104], chosen as it had the closest feature structure to the data gathered with SocialBus. This dataset contained a total 267 368 tweets belonging to 453 users.

The expected outcome would be for all accounts to have high scores, or at least a score higher then 0. The better the results of this experiment, the more confidence there was on the scoring function's abilities to find trolls.

To perform this type of validation, the hyperparameters specified in Section 4.2.1 were not the same as the ones used for the Portuguese language: The date for the function **is_account_created_after** was set at 08 august 2016. This date predates the day of the American General Elections by 3 months. In the function **has_normal_name**, instead of Portuguese names, the list used was of every name in the English language – To aid in this process, the python library **name-dataset** was used. This library searches for a given name in a set of approximatelly 160 000 English first names and 100 000 last names [101], which was a number deemed acceptable for this exercise. Finally, the function **contains_keywords** had its array of keywords modified to the most used insults and political words used in the 2016 US elections [5], left wing and right wing. Due to the posterior deletion of the troll accounts by Twitter, their avatar picture was no longer available. This lead to a reduction in the final possible score to 5, and the minimum score for suspicion to 3.

5.3.2 Dealing With Class Imbalance

As this project possessed a dataset with class imbalance (Section 2.3.7), the measure selected for dealing with this issue was a mix of undersampling with SMOTE, an approach that is referred in this technique's original paper [72], and would be applied to account and post features.

The previous methods used are independent of the algorithm applied. There are methods specific to some algorithms, as is the case with the embedding features. The algorithms used for this feature group (Neural Networks) have different mechanisms to deal with class imbalance. An example of this type of mechanisms is the update of the model weights in proportion to the error of the model calculation [143, 67].

The challenge of choosing the ratio is the amount of potentially relevant information that would be lost in case of undersampling the data too much, as it was seen through the statistical analysis that the dataset contained various types of outliers (Section 5.4.2). In this case, it was selected a 1:2 ratio of troll to non troll accounts in order to not reduce too much the non troll accounts, which would narrow the account diversity (which was shown to be non discardable). The presence of the non troll accounts with different behaviors was considered a positive measure in reducing potential overfitting. SMOTE was then applied on the training dataset.

A sample of 80% of the data was selected for the application of SMOTE. Of the dataset of 20 758 accounts and their tweets, 80% of them were selected creating a training set of 16 391 normal accounts and 217 trolls, and a test set of 4 098 users, 52 troll and 1 995 normal accounts. All the accounts present in the train set were distinct from the ones in the test set and vice versa.

After dividing the data, the SMOTE was executed through the usage of the package *imblearn* of the library **scikit-learn**[48]. These new Post and User feature sets, consisted of 3278 normal users and 1639 troll users. Figure 5.1 presents an illustrative example of the data distribution based on the features nrNouns (Y axis) and NrVerbs (X axis). On the left, the non treated data was plotted according to these variables for trolls (orange) and non trolls (blue). It is visible how

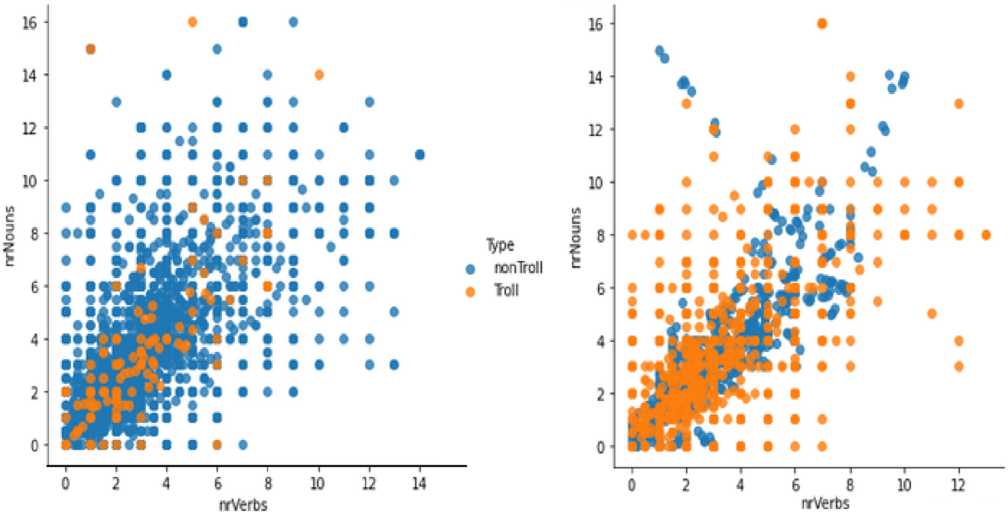


Figure 5.1: Scatter Plot of two exemplar features of the Post Features dataset before (left) and after (right) applying SMOTE and Undersampling

overwhelming the negative examples are both in terms of quantity and distribution, as the trolls are distributed in the same areas as the non trolls.

After applying SMOTE and undersampling of the majority class, in Figure 5.1 on the right, the trolls are still in minority, but are better distributed across the referential. Although the application of SMOTE increased the overlap between the two types of accounts, the generated models were expected to distinguish between both classes.

With this new distribution, it was expected that the classification algorithms would perform better than if applied on the data without prior treatment.

5.3.3 Performance Estimation Methodology

To measure and evaluate the results of the classification experiments, different metrics are calculated and analyzed. It is important to measure the models' performance on a multitude of metrics, as well as evaluating which are the best metrics both for evaluation and for comparison with other troll detection approaches.

With a balanced train set, the test set used was left purposely imbalanced, in order to have a closer distribution of accounts to that found on Twitter. This distribution would affect how performance was measured. As with other studies referenced in the state of the art that had imbalanced datasets (Section 3.2.4), metrics such as accuracy cannot be considered a reliable measure, as it provides an overoptimistic estimation of the classifier ability on the majority class [2]. Another popular measure used in this type of exercise is the precision metric, although in this case precision was not considered as important, as it was considered better for a model to correctly predict which users were in fact trolls (recall) than to predict trolls most of the times it labels a user a troll (precision). Assuming this premise, false positives were considered less harmful for this work than false negatives, as labelling a troll user as a non troll is considered worse than considering a non troll user a troll – This is due to the fact by labelling a non troll user a troll, he may display a behavior or language which makes him problematic, but to label a troll user a non troll, however, implies the model is not detecting the characteristics that distinguish him from a regular user.

Although less important, precision must not be discarded completely, as a high recall does not necessarily mean the model has good performance (e.g. labelling all users as troll gives a positive class recall of 100%). An investigation regarding the evaluation measures used in the state of the art (Section 3.2.13) revealed the ones that were given the most importance were mostly precision and recall. For the studies with the more imbalanced datasets, however, different measures were prioritized: the metric of AUC-ROC curve was used and considered important for these tasks, as there are studies that verify that this metric is mostly unaffected by skewed distributions [60]. Considering this information, how this project's models were to be evaluated were then inspired on these specific studies. The metric considered more important for the project, on which the model's performances would be asserted, was chosen to be the AUC-ROC Curve (Section 2.3.5).

Regarding the model evaluation, the preferred method was the Holdout (Section 2.3.9) method.

5.3.4 Tuning the Threshold

In order to compare the different experiments, various thresholds were experimented, with the tuning of this value allowing to obtain different metrics of precision and recall. A threshold was needed in order to compare the metrics of precision and recall with other models and other studies. For this project, the metric of the Matthews Correlation Coefficient (Section 2.3.6) was selected, meaning the best threshold for each model was considered the one with the highest MCC value.

The main measure to assert the models' performance in this dissertation would continue to be the AUC, even for a better comparison with other studies, but metrics such as precision and recall would be calculated from the threshold considered best by this method.

5.3.5 Algorithms Used For Numeric Feature Sets

There were several algorithms suitable for application on this work, based on the multiple approaches found in the state of the art (Section 3.2.10). In order to assert which ones displayed the best results, the algorithms experimented were limited to the standard ones used in other approaches, with the exception of when the tweets' text was used directly, where the Neural Network algorithm was used, which is currently the state of the art to deal with this kind of data.

For the sets limited to numeric features (Users, post and the combinations of all the feature sets), two algorithms were selected: **Support Vector Machine** and **Random Forest**. The parameters used in these algorithms is present in Table 5.3. All algorithms were implemented through the *scikit-learn* library [121]. The values of trees for the random forest were selected in accordance to the typical number of trees used in works in the area [40], as it is verified that above the 1000 trees, a limit was achieved where no significant improvement was verified [97].

Table 5.3: Algorithms Used and their Parameteres

Algorithm	Implementation
SVM	Linear Kernel
	Polynomial Kernel
	Radial Basis Function Kernel
Random Forest	50 Trees
	100 Trees
	500 Trees
	1000 Trees

5.3.6 Algorithms Used For Text Features

For the features consisting of the users' texts, the preferred algorithm chosen was the Neural Network algorithm. This is due to it being effective for high dimensionality problems, as is the case with text and documents with a vast vocabulary, as well as having the ability to learn and model non-linear and complex relationships, as is the case with human speech [80]. Once the algorithm was chosen, the type of neural network had to be selected. For works in the area of text classification, there has been an observed tendency for both Recurrent Neural Networks (**RNN**) and Convolutional Neural Networks (**CNN**). Of this observation, it was verified that for tasks where the length of the text was important, such as question-answering, translation, etc., there was a more frequent usage of RNNs. For tasks where feature detection in the text was more important, such as searching for angry terms, sadness or political speech, CNNs were the most used [65]. Taking from not only what is done in the state of the art in troll detection but also on text classification, a CNN was chosen. A CNN was created using spaCy's Text Categorizer function, which allows to create and tune a convolutional neural network according to the desired parameters and the desired labels, which in this case were **Troll** and **Non Troll**. In order to categorize the data, it is firstly passed through several NLP associated processes such as parsing (determining the syntactic structure the text), stopword removal, tokenization and vectorization (converting the text into word vectors). In spaCy's architecture and NLP in general, these processes are done in a sequential manner, in what is called a *pipeline*. Once all the processes explicited are done, the data is then passed on to spaCy's CNN [54].

This network works, according to its author, in a four step strategy - **Embed**, **Encode**, **Attend**, **Predict**, which correspond to the network's 4 layer architecture, adapted from the work of Yang et al. [147].

The Embedding process maps the words and characters into one-dimensional numeric vectors, as referred in (Section 2.4.9).

The Encoding stage works by taking the sequence of word vectors and computing a representation called sentence matrix, where each row represents the meaning of each token in the context of the rest of the sentence. There are several methods to compute this process – spaCy uses a RNN, which creates an intermediate representation before sending the data to the CNN.

After this stage, the Attending procedure calculates the words' *attention*. This process reduces

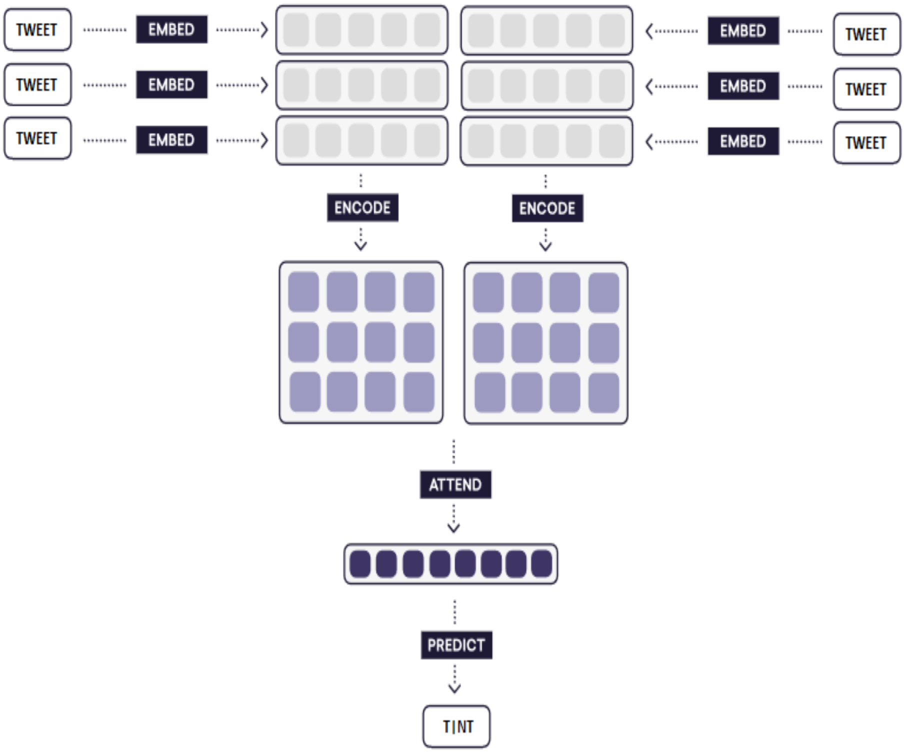


Figure 5.2: Representation of spaCy's Text Categorizer Layers, adapted from [55]

the input to only parts of it, where the most relevant information is concentrated instead of an entire sentence. In spaCy, the attention layer reduces the matrix to a vector, creating a "summary" vector.

Finally, the Predict stage, where once the text or pair of texts has been reduced into a single vector, the target representation is learnt and in this case the probability of the text's membership for both classes is outputted.

This process, applied in Troll detection, is visible in Figure 5.2.

With the understanding of how the network functioned, the parameters had to be chosen. In this case, spaCy allows the tuning of several parameters for the categorizer: the data to use, number of iterations, number of texts and the type of vectorizer (in case of wanting to import a different vectorizer from the one the library offers). For the training process itself, spaCy defines an *optimizer* – a function used during the training of the model to hold intermediate results between updates of the model weights [141]. For this, spaCy uses the *compounding* function, taking three parameters: the starting value, the maximum value and the compounding factor. This function allows to regulate the size of the batches to use while training the model, beginning in the starting value parameter and calculating the next iteration by multiplying the previous value by the compound rate specified as the third parameter, without exceeding the maximum value specified as second parameter. The data used was 80% of all the tweets gathered: 82,238 (78 923 troll and 3315 non troll) tweets, without class imbalance treatment (as referred in Section 5.3.2). For the number of iterations, the number chosen was of 25 as it was verified that above this number there was no significant improvement on the model's performance. The number of texts was the total of the tweets gathered, separated into train set and test set. As spaCy's vectorizer was deemed as adequate for the task, no external one was imported, but created during the execution. For the model training, the parameters were chosen based on a combination of the existing literature [141] and a trial and error approach would be necessary in order to determine the best results for the optimizer function. The best compounding value was found by a trial and error approach, consisting of comparing the results as the parameters of this function were changed, until reaching the values of 37.0 for starting value, 150.0 for maximum value and a compounding factor of 1.001. These values were inspired in spaCy's own documentation [129] as well as the work of Vasilev [141]. They were then applied to both text feature sets. To evaluate the model from the aggregation of the account's verdicts, the test set, which had the tweets joined together for each user, would have to be separated into individual tweets, changing the number of testing samples from 4150 users to 21707 messages.

With the implementation details aligned, the next stage would be to analyze the results given by the processes explained in this chapter, with the metrics detailed, and draw conclusions from them.

5.4 Exploratory Data Analysis

This section refers another experiment performed on the data, where simple statistical and visualization methods were applied, to better understand the data and the problem. An assessment of the feature correlations of the various groups was also performed.

5.4.1 Most Popular Words

With the tweets collected and their type assigned, a superficial analysis on the texts' most used words was considered beneficial. This type of visualization could reveal trends, a general theme in the tweets or a type of word repeated in one of the groups that could be used later for further investigation. With the objectives defined, firstly, for an efficient information retrieval process, a list of stop-words was compiled, containing the standard stop-words found in the NLTK library [95], as well as some Twitter specific vocabulary (e.g. RT) and were removed from the tweets in order for them to not overwhelm the representation. As the amount of non troll tweets were in a higher number than the troll ones, it was expected of the distribution in the former to be more sparse in terms of thematics, as these follow trending topics and trending tweets. The most popular words were displayed using the library **WordCloud** [109].

The most popular words for the non Troll accounts reflect the date when they were collected. Specifically, the tweets were gathered from the beginning to the end of march 2020, a time where the *Covid-19* virus was spreading in Portugal, where messages appealing to staying home and practicing social distancing quickly became predominant [15]. One of the messages most repeated in Portuguese media, *"Vai ficar tudo bem"* (everything will be ok), heavily influences the most used words list, as is possible to verify in Figure 5.3, along with various variations of words

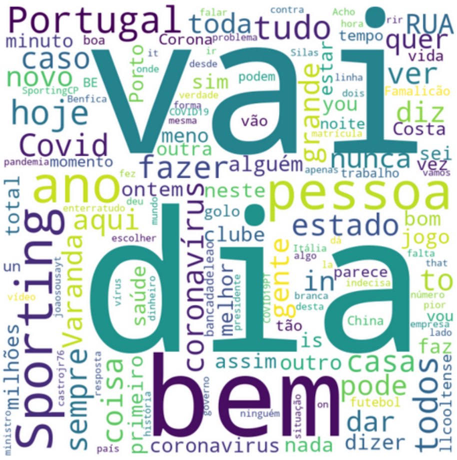


Figure 5.3: Most Popular Words of non Troll Users

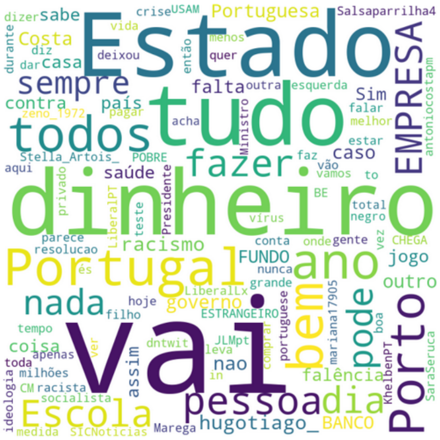


Figure 5.4: Most Popular Words of Troll Users

referencing the virus. One aspect of the collected text, verifiable upon this visualization, is that although it contains politically themed words (e.g. *"Estado","Portugal","BE",etc.*), it is also sensible to trends that transcend politics – One of the most frequent words is the word "Sporting", which is linked to sports. This is considered a consequence of using popular accounts in the data collection: The information obtained better reflects the reality of what is being discussed at a given time. However, in a community like the Portuguese Twittersphere, this creates a trade off where other kinds of trends are also collected, such as sports, a vastly spoken about topic in Portugal. This implies that there will be noise present in the data, where in this case *noise* describes tweets by relevant and non relevant accounts on themes that are not of interest for the project.

The most popular words for Troll users, present in Figure 5.4 reflect the collection methods and rules used. The vast majority of words are linked to politics and economics, as the users where selected partly because they fulfilled the requirement of only talking about these matters. Of this collection, words in all caps are present, which may be an indicator of trolls and their usage. Another particularity of this group of words is the presence of three Portuguese parties: CHEGA, LiberalPT (*Iniciativa Liberal*) and BE (*Bloco de Esquerda*), parties which have been accused by the community of having trolls on their behalf, and are frequently criticized and referenced by trolls of the opposing political spectrum. Finally, there are several user handles in the most popular words list (e.g. Salsaparrilha4), the majority of which displayed high scores and were considered trolls in the annotation stage. This can be an indicator that accounts that frequently interact with trolls may also be trolls themselves.

5.4.2 Statistical Analysis of Features

The statistical analysis was done by splitting the created feature sets, creating sets consisting of just trolls and just non trolls. From the analysis of Dollberg and Im et al. [33, 57], trolls that write in English contain in their speech more swearwords, words in all caps and stop-words. If that

Table 5.4: Statistical Analysis Results of Account Features

		Trolls	**Non Trolls**
Account Age	Mean	4.0185	6.3007
	Median	2.0	6.0
Following to Follower Ratio	Mean	3.4019	6.3007
	Median	1.6023	0.9286
Numbers in Profile	Mean	1.3909	0.8483
	Median	0.0	0.0
Contains Face	Mode	0.0	0.0
Has Normal Name	Mode	1.0	0.0

case was verified in the sample of trolls recovered, it would be a positive indicator that the same condition was verified for Portuguese Trolls.

The features groups' mean and median were calculated. For the binary variables, the mode was the preferred metric. Only the numeric values were analyzed, so variables containing strings were not studied in this manner. A sample consisting of all the data (99 727 normal tweets and 4016 Troll tweets from 20 758 accounts) was analyzed, with the results present in Tables 5.4 and 5.5.

Regarding the post Features, it was verifiable that for this case, the Trolls' tweets had a lower median length than their normal counterpart, and contained a mean of more 0.65 stop-words. Based on the collected sample, the findings of Im et al. relative to stop-words apply to Portuguese trolls, although not as overwhelmingly as in Russian/American Trolls. Relative to swearwords and words in all caps, both types of accounts had close values regarding swearword mean and words in all caps. Using the data of the sample, the results were not significant enough to make assumptions on these matters. Of the remaining features, trolls were observed to write with more verbs, nouns and adjectives. Regarding political insults, trolls showed a higher mean and higher median.

For the case of account features (Table 5.4), the mean account age for trolls was of 4 years. Some of the found trolls had older accounts, which a subsequent analysis on them showed that the majority had their tweets starting in a more recent point in time, such as the months before elections. This type of profiles are part of a known tactic employed by trolls, in which they appropriate older, unused accounts in order to avoid detection [5]. Using the median, however, which is more resistant to outliers, demonstrates the median account age of trolls being 2 years, which is one third of the median value for normal accounts. Troll accounts had a higher following to follower ratio both in terms of mean and median. The same applied to the numbers in profile measure, where the mean showed at least one number in a user's profile handle. For the binary variables, the majority of both troll and normal users had recognizable faces in their avatars. This did not mean that the pictures were of themselves, but the relevance of this feature would have

Table 5.5: Statistical Analysis Results of Post Features

		Trolls	**Non Trolls**
Sentiment Polarity	Mean	-0.1216	-0.0916
	Median	0.0	0.0
Tweet Length	Mean	104.1484	101.9208
	Median	78.0	91.0
Nr of Stopwords	Mean	5.2553	4.6050
	Median	3.0	3.0
Nr of Hashtags	Mean	0.0370	0.0
	Median	0.0713	0.0
Nr of Mentions	Mean	0.9943	1.0570
	Median	1.0	1.0
Nr of Urls	Mean	0.2454	0.2756
	Median	0.0	0.0
Nr of Emoticons	Mean	0.5492	0.3366
	Median	0.0	0.0
Nr of Swearwords	Mean	0.0428	0.0396
	Median	0.0	0.0
Nr of Favorites	Mean	0.0	0.0
	Median	0.0	0.0
Nr of Verbs	Mean	2.3452	2.2210
	Median	1.0	2.0
Nr of Nouns	Mean	2.8203	2.6076
	Median	2.0	2.0
Nr of Adjectives	Mean	0.7632	0.7444
	Median	0.0	0.0
Nr of Entities	Mean	1.1612	1.2999
	Median	1.0	1.0
Nr of Political Keywords	Mean	0.6134	0.0941
	Median	1.0	0.0
All Caps to Normal Word Ratio	Mean	0.0407	0.0498
	Median	0.0	0.0

Table 5.6: Features displaying the highest correlation in the Post Feature set

Features	Correlation
Nr Verbs, Nr Stopwords	0.83002
Tweet Length, Nr Stopwords	0.81715
Nr Stopwords, Nr Nouns	0.81034
Tweet Length, Nr Nouns	0.79694
Nr Verbs, Nr Nouns	0.66057

to be assessed. Regarding the names, troll users had a majority of non proper noun names, as opposed to their normal counterpart.

The statistical analysis allowed to verify that some hypothesis proven in the state of the art for trolls in other countries had positive indicators of being true for the Portuguese context. This could indicate that when applying classification techniques, the methods and features used for those cases would be interchangeable to the Portuguese reality. This first analysis allowed to verify that some variables have a high disparity among both types of users, but others are close together. Most of the variables with higher discrepancies are present in the account features dataset, while the results for the post feature set are closer together. With the results in mind, it would have to be asserted if the feature set with the highest discrepancies originated the best model.

5.4.3 Feature Correlation

With the feature distributions analyzed and compared, a feature correlation was then performed. To assert the similarity between features, the pandas library's *correlation* function was used, and the value outputted between all features was documented. The minimum correlation value decided to discard a feature at this stage was 0.9, a value considered high, but deemed as sufficient for this work, as each feature was considered to add information to a user's representation, and the features would be assessed again in a feature importance exercise (Section 6.3.1).

The feature correlation for the **Post Features** set was calculated for every feature, with the ones that displayed the highest correlation visible in Table 5.6. The feature with the highest correlation were the number of verbs and the number of stop-words, which implied the more the verbs present in a tweet, the more probable it was to find more stop-words in it. The higher correlations between grammatical categories (words, nouns, adjectives) and a tweet's length are understandable, as the longer the tweet, the more words it contains. Despite some variables displaying higher correlations between them, none of them displayed a value considered sufficient enough to eliminate any of the features.

Of the features in the **Account Features** set, no feature displayed a correlation value higher than 0.1.

Chapter 6

Results

In this chapter, the results of a validation exercise on the scoring function are analyzed and its pertinency for this work is discussed. The validation of the score function was considered an important step, as there was needed a way to assert the veracity of the taken assumption that accounts with higher scores were more likely to be trolls.

With the scoring function validated, this chapter then presents the results obtained and compares them between each other and other works in the area. This analysis allows to not only answer the remaining research questions, but to assert if the majority of the project's goals have been fulfilled.

6.1 Asserting Validity of the Scoring Function

The scoring function implements a heuristic approach, based on existing literature and some information collected in the exploratory data analysis exercise (Section 5.4). The goal of this exercise is to test how appropriate the developed heuristic is.

With the dataset and expected outcome defined in the Experimental Setup (Section 5.3.1), the experiment results are visible in Table 6.1.

Table 6.1: Scoring Function Results

Score	***Number of Users***	***Percentage***
3+	208	46%
1-2	163	36%
0	82	18%

Given the obtained results, the scoring function was therefore considered relevant for the signaling of positive accounts, as it assigned high scores to a significant percentage of users. That percentage, however, was not significant enough to limit the troll detection efforts to this function and not pursue a machine learning approach. Due to the scoring function's results, however, its functions were used as features, and are present in the account features group.

6.2 Experiment Results and Analysis

This section details the several experiments performed during the duration of the project, compares them and discusses their results. In the subsection below a comparison on the result of all the models is explicit, with each feature group and their combinations discussed on the following subsections.

6.2.1 All model results

In Table 6.2, the best obtained results for all the different groups of features are detailed. All results were obtained with the Random Forest algorithm, with 1000 trees. Along the results, in bold are highlighted the best value obtained of a given metric in all the experiments performed.

Table 6.2: Metrics and Results obtained with Random Forest with 1000 trees

Feature Set	*AUC*	*Best MCC*	*Best Threshold*	*Precision*	*Recall*	*F1-Score*
Post F	0.85	0.16	0.455	0.08	0.44	0.14
Account F	0.70	0.10	0.512	0.06	0.25	0.10
Embedding F (all together)	0.71	**0.46**	0.167	0.57	0.38	**0.46**
Embedding F (mode of tweets)	0.81	0.33	0.147	0.22	0.50	0.31
Embedding F (mean of tweets)	0.81	0.33	0.147	**0.81**	0.25	0.38
Account F + Post F	**0.90**	0.27	0.437	0.15	0.54	0.24
Account F + Embedding F	0.78	0.16	0.423	0.10	0.33	0.15
Post F + Embedding F	0.86	0.18	0.469	0.09	0.44	0.16
Account F + Post F + Embedding F	0.89	0.25	0.398	0.13	**0.56**	0.21

6.2.2 Comparison Between Feature Groups

The experiments performed on the account and post feature groups were similar, and consisted on the application of the Random Forest and Support Vector Machine algorithms. The results of these experiments are visible on Table 6.3. The ROC curves of the individual feature groups can be seen in Figure 6.2.2, with the account model in Purple and the post model in Greem.

The best threshold was asserted for both feature groups. This value was obtained through the calculation of the Matthews Correlation Coefficient (MCC). As referred in Section 5.3.4, the threshold which displayed the highest value on this metric was the one considered to calculate precision, recall and F1 values. These values would then be used for comparison with the other models and other studies.

Table 6.3: Account and Post Features Algorithm Result Comparisons

Algorithm	Specification	Account Features AUC	Post Features AUC
Random Forest	50 Trees	0.68	0.82
	100 Trees	0.69	0.84
	500 Trees	0.7038	0.8532
	1000 Trees	0.7039	0.8542
SVM	Linear Kernel	0.69	0.79
	Polynomial Kernel	0.60	0.59
	RBF Kernel	0.66	0.55

The model which displayed the best results for both account and post features was the one generated with the Random Forest algorithm, with 1000 trees, displaying varying discrepancies when compared to the results of the SVM. For both feature groups, the results between the two algorithms are close enough to each other to consider these differences hardly statistically significant. Both of the most used algorithms found in troll detection literature display positive results on both feature groups. Despite this observation, the best performing SVM (linear kernel) displayed a worse result than the RF with the lowest amount of trees for all examples. While some experiments in the state of the art had SVM's outperforming RF's, this is not the case for this work.

In the case of the account model, its AUC (0.70) was considerably lower than the one obtained with the post features (0.85), which can in part be related to the fact that a user's profile contains limited information regarding his shared content and intentions. This implies that it is not the most accurate method to categorize Twitter accounts solely by their profile metadata. This can be considered normal, as a trolls' primary mean of destabilizing is through speech, and in the case of this social network, through their tweets. Its MCC is also lower, proving that the information contained in this feature set is sufficient to detect trolls, but, based on an analysis of the models, not enough to surpass features regarding an account's text (Table 6.2).

After calculating the MCC for both the best account and post models, the best coefficient obtained was of 0.16 for post features and 0.10 for account features. According to Chicco and Jurman [22], these values are not considered positive for the experiments, as they display a high amount of false positives. For this problem in particular, however, it was considered better to have a false positive than a false negative. When increasing the threshold, the models' precision is lower but its recall significantly increases, allowing to detect most of the troll accounts even with a higher number of false positives. This can be seen through the post features model ROC curve, which shows that a low threshold can, for example, achieve a true positive rate close to 80% while having a false positive rate close to 20%. For these reasons, the model was deemed satisfactory.

The Post features model displays a higher AUC metric (0.85). The analysis of this model's results of show that it is possible to detect Portuguese trolls just from the superficial aspects of their text, better than, inclusively, their profile information.

A conclusion that is possible to take from the results of both of these feature groups is that based on the numerical features of the language used in a certain user's tweet, it is possible to

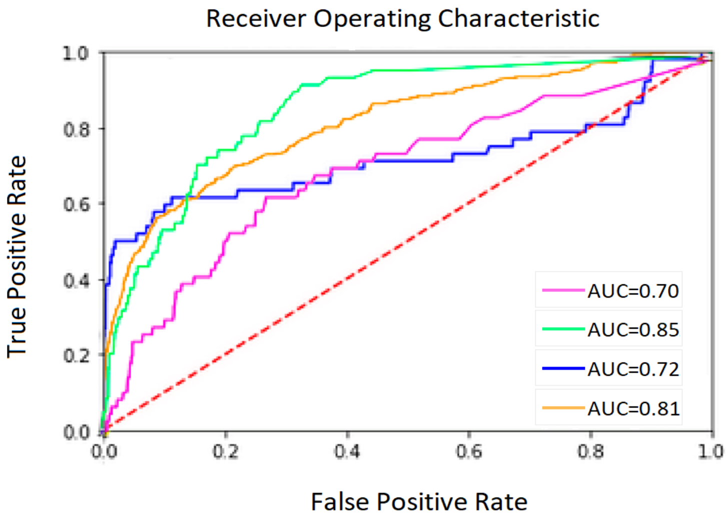


Figure 6.1: ROC curve plots of Account (Green), Post (Purple), Embedding features with all the tweets condensed (Blue) and the aggregation of classifications of each user (Orange)

verify whether that user is malicious or not. It is also possible to assert a user's membership based on his profile metadata, although with worse results. One aspect that can be discussed is that these feature groups are prone to having false positives, which, when analyzing other existing studies, appears to be a common trend.

6.2.3 Embedding Features Results

As referred in Section 4.5.3, two types of approaches for text classification were proposed. In the first case, the observations are the accounts. In the second, the problem is solved indirectly by changing the granularity to the level of the post and then aggregating all the predictions concerning one account. Both approaches are solved using spaCy's *Text Categorizer*.

The text categorizer's parameter for number iterations was set at 20 as above this number, no significant gain was registered in terms of AUC.

As is visible in Figure 6.2.2, in blue, the AUC obtained from classifying all the tweets together – 0.72, is not considerably high, and its threshold, according to the ROC curve plot, has a high variation, even reaching a point where it is worse than random sampling. It displays, however, the fastest curve rising of all the models. Despite these values, when analyzed its best threshold – 0.167 (Table 6.2), this model displays the highest MCC, with 0.46 as well as the highest F1 value for all models calculated, with a value of 0.46 as well. This implies that according to the papers suggesting the usage of the MCC as the best metric to assert performance for imbalanced datasets, the text model with all the tweets together is considered the best model, implying the values of precision and f1-score influence the MCC value in a larger proportion. Considering AUC, this model was not considered overwhelmingly positive, but still displays better results than the account feature group.

The aggregation of classifications for each user, as visible in Figure 6.2.2,in orange , has an AUC metric of 0.81, which is considerably higher than the same metric for the approach using all tweets together, although possessing a lower MCC, with 0.33, and therefore a worse threshold. Although this model may display difficulties and a lower AUC than the post feature set, these difficulties appear to be lower than the ones displayed by the approach using the condensed tweet, considering their AUC values.

To assert the model's performance the MCC and the highest threshold values were calculated, giving values of 0.33 and 0.147 respectively. The next stage was to perform the experiments described in Section 4.5.5 using that threshold (0.15).

Of both experiments (Table 6.8), the results are varied, with the method of classifying users by the mode of their classification scores achieving a higher recall, with 0.50 and a lower precision, with 0.22, while the method of calculating the users by the mean of their classification scores method achieving a higher precision, with 0.81 and a lower recall, with 0.25. Using the mean method, however, yields the highest precision value obtained of all the experiments (0.81), with, however, one of the lowest recall values (0.25). Both models using just embedding features output the highest MCCs and highest precision values of all experiments.

Comparing the results from the two different models, these approaches yield positive results, with one approach outputting the highest MCC and F1 score, and the other outputting the highest precision value. The embedding feature group is not the best model when performing comparisons by AUC, as is the case of this dissertation, but displays the highest precision or MCC values of all experiments, depending on the type of approach used with the embedding feature set. While this group performs worse in terms of recall, it does not display the problem of being prone to false positives as greatly as the post and account feature sets. The results obtained allow to reach the conclusion that it is possible to detect trolls based solely on features learnt from their text, achieving greater results than the model generated from the metadata in their accounts, but not as well as the one created from the numerical aspects of their text.

6.2.4 Comparison Between Combinations of Different Sets of Features

As with the approaches in the state of the art which divide their features into groups, in this project there was a following exercise where the groups of features were joined together, and their models assessed. Grouping the features together was expected to yield better results than using the sets individually, as the feature groups together provided a theoretically more complete representation of the user under evaluation. Aligned with what was performed previously, the Account (**Acc**), Post and Embedding (**Emb**) features were combined, and the Random Forest and Support Vector Machine algorithms were applied. Their results are visible in Table 6.4.

Of the combinations of the feature groups, the results increased for all algorithms, achieving the highest obtained AUC of all the experiments, with a value of 0.90. The ROC curves of the combined models are visible in Figure 6.2. All but one of the combination models generate AUC values above 0.80, which indicate that a grouping approach is preferable to single feature groups.

Table 6.4: Models Generated From Geature Group Combinations AUC Result Comparisons

Algorithm	Specification	Acc + Post	Post + Emb	Acc + Emb	Acc + Post+ Emb
Random Forest	50 Trees	0.86	0.83	0.77	0.86
	100 Trees	0.87	0.85	0.77	0.86
	500 Trees	0.89	0.86	0.78	0.89
	1000 Trees	0.90	0.86	0.78	0.89
SVM	Linear Kernel	0.85	0.65	0.71	0.85
	Polynomial Kernel	0.70	0.59	0.72	0.73
	RBF Kernel	0.74	0.59	0.71	0.70

Of these results (Table 6.2), the model with the highest AUC is not the one containing all the feature groups, but the one which combines only the numerical aspects of the text and the account metadata (although the difference between that model and the one with all the features is hardly significant). However, in terms of recall, the model using all the features displays the highest recall value for its best threshold – 0.56. The model considered the best of the dissertation was, therefore, the model combining account and post features, and would be the one compared with the literature in terms of best results.

A conclusion to take from these results is that regarding AUC, for the collected sample, the users' membership is asserted the best when using features related to their numerical aspects of text, together with their profile metadata. This conclusion allows to answer one of the core questions of this dissertation. The features that lead to better results in detecting trolls in the Portuguese Twitter are a combination of the accounts' profile metadata along with the numerical aspects of their text.

6.2.5 Discussion

After a comparison exercise with the different models generated in this work, the results considered best needed to be evaluated in context, meaning they required to be compared to the several studies on troll detection from which the feature sets were removed. For this exercise, the solutions that prioritized the metric of AUC were given more importance, as these allow for a more direct comparison. While some researchers stated their precision and/or recall values, others opted for just AUC. Given that all the studies use different datasets, the results are not directly comparable. However, we can check if the conclusions drawn based on the obtained results are in line or not with with the ones obtained in other studies. The studies that had balanced datasets used as main metric the models' accuracy. This metric is not directly comparable, as the accuracy measure with imbalanced datasets (as this the case of this project) is usually high, and does not detail relevant information regarding the detection of the minority class. The comparison of approaches for post features is present in Table 6.5.

Using the metrics from the threshold with the highest MCC shows that the model's results are not only in line with other similar approaches with imbalanced datasets, but given the different conditions, display higher results in some metrics. As is the case with the work of this dissertation,

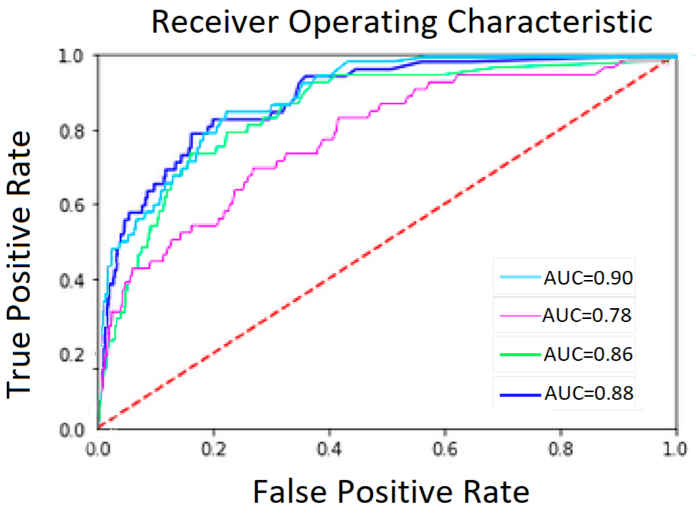


Figure 6.2: ROC Curves of the Random Forest Classifier (1000 trees) for all feature combination models: Account+Post (Light Blue), Account+Embedding (Purple), Post+Embedding (Green), Account+Post+Embedding (Dark Blue)

features of the post feature group perform well on average. Using the work of Im et al. as reference [57], the precision metric for this type of features is low, but in line when compared to studies (the ones that reveal the value for this metric).

Comparing the account features' results with the state of the art (Table 6.6), it is visible that there are few papers that used or distinguished feature sets that could be classified in this group of features. The ones that did, however, show an unparalleled higher AUC for the work of Fornacciari et al. [42], although the authors only state the recall value obtained. For the Portuguese Trolls and the work of Im et al. [57], these are close in terms of AUC metric.

Despite having a lower AUC, when compared to the study closest in terms of this metric, the work of this dissertation has a slight difference in terms of precision, but a considerably higher recall. The lower number of studies that use these kinds of features, and the overall lower results obtained from this and other approaches indicate that although it is possible to detect malicious accounts based solely on their account metadata, this strategy is prone to have lower results than the ones involving text.

The embedding feature set was the most inclusive, containing works solely relying on this type of features. When comparing the best results of the text features with the state of the art (Table 6.7), it is visible that both approaches used present an AUC result which is in line with what is done in other studies of similar kind.

Both approaches have even a higher precision metric than the remaining studies, although with a lower recall (compared to the studies that present this metric). A note that the highest AUC obtained (0.98) uses a BoW approach, while the one which uses embeddings [144] has the second highest AUC when compared to the remaining studies which use older types of representations. As this study displays a higher AUC than some approaches for the tweet classification, it can be affirmed that more modern representations as the ones generated from embeddings provide

Table 6.5: Post Features Results

Paper	*AUC*	*Precision*	*Recall*	*Accuracy*
A holistic system for troll detection on twitter [42]	0.87	–	0.79	–
Still Out There : Modeling and Identifying Russian Troll Accounts on Twitter [57]	0.86	0.15	0.21	–
TrollBus	**0.85**	**0.08**	**0.44**	–
Antisocial Behaviours in Online Discussion Communities [20]	0.82	–	–	–
Filtering Trolling Comments through Collective Classification [30]	0.67	–	–	–
Hunting For Troll Comments in News community Forums [89]	–	–	–	73.85

positive results for troll detection, but don't yet surpass results obtained from older representations.

As with this dissertation, the approaches in the state of the art experimented the various combinations of their feature sets, and documented the one with the best results as the best obtained model. Assuming the best model in this work was the one with the highest AUC value, the comparison of best results is present in Table 6.8.

It is visible that, given the different conditions, the best results of this project display a higher AUC than some of the similar works of the area (Table 6.8). These works may not be directly comparable due to different feature sets and experimental setups, but knowing there were similar conditions (e.g. imbalanced dataset) and the features were derived from the work performed on these same studies, it is possible to claim the results of this paper are as good as the ones in the state of the art, performing better than some. These results seem to indicate that using all the features together creates a quality model, even when the features from which they were adapted are used with other languages and communities.

6.2.6 Conclusions

For this work and the various feature sets developed, all the models created showed positive results regarding AUC, with the values of precision and recall in line with the works performed that had similar setups.

One limitation asserted from the analysis of the results is the few studies that are suitable for comparison, as there are more studies that use balanced datasets, and measure their performance with metrics according to that distribution, which are not adequate for this concrete approach (e.g. accuracy). Along with this, the existing studies do not make available their datasets for comparison. Despite these conditions, it was possible to compare the model with the studies that did use comparable metrics such as AUC, where it is possible to assert the model not only has results in the range of the existing studies but also surpasses some of them on this metric.

Table 6.6: Account Features Results

Paper	*AUC*	*Precision*	*Recall*	*Accuracy*
A holistic system for troll detection on twitter [42]	0.87	–	0.79	–
Still Out There : Modeling and Identifying Russian Troll Accounts on Twitter [57]	0.74	0.07	0.01	–
TrollBus	**0.70**	**0.06**	**0.25**	–
The Metadata Troll Detector [33]	–	–	–	0.69

The choice of using an imbalanced dataset approach was chosen, as the models would be used in a live environment with real world data (Section 3.1), and it was considered beneficial for them to be trained on a data that simulated the reality of Twitter.

From these results, it is possible to answer the first research question: While there is no feature set from which it is not possible to detect trolls, the features regarding their speech, either through numerical aspects of the text or the text itself, pose as the better options for a troll detection work. Generalizing the results obtained from this work, the combination of various groups of features regarding the text and the account metadata has the potential to generate better results than using the different feature groups separated.

6.3 Feature Analysis

With the models calculated and their results analyzed, a study of the features used, their importance, and their correlation was performed. The study of the features and their effects on the models would not only allow to discover eventual features harming their performance, but also to understand what were the features considered most important, thus answering one of the raised research questions. This study would reveal what were the features most relevant to find trolls, while indirectly allowing to better describe the Portuguese trolls and how to find them.

6.3.1 Feature Selection and Importance

With the correlations between all the features asserted and their values deemed as not significant enough to discard (Section 5.4.3), this did not mean all the features were necessarily beneficial to the model results. In order to assert if there were features lowering the best possible results, a **Recursive Feature Elimination** exercise was performed. This exercise was performed through the usage of *sklearn*'s RFE function [120], which recursively eliminates features in the random forest model and asserts its performance variance over each iteration, according to a given metric. For this case, the metric of AUC was prioritized. After running the function with 25 iterations, the output for all feature models (Account, Post, Account+Post and Account+Post+Text) was that there was only one feature harming the models' AUC, which was the feature *contains_face*.

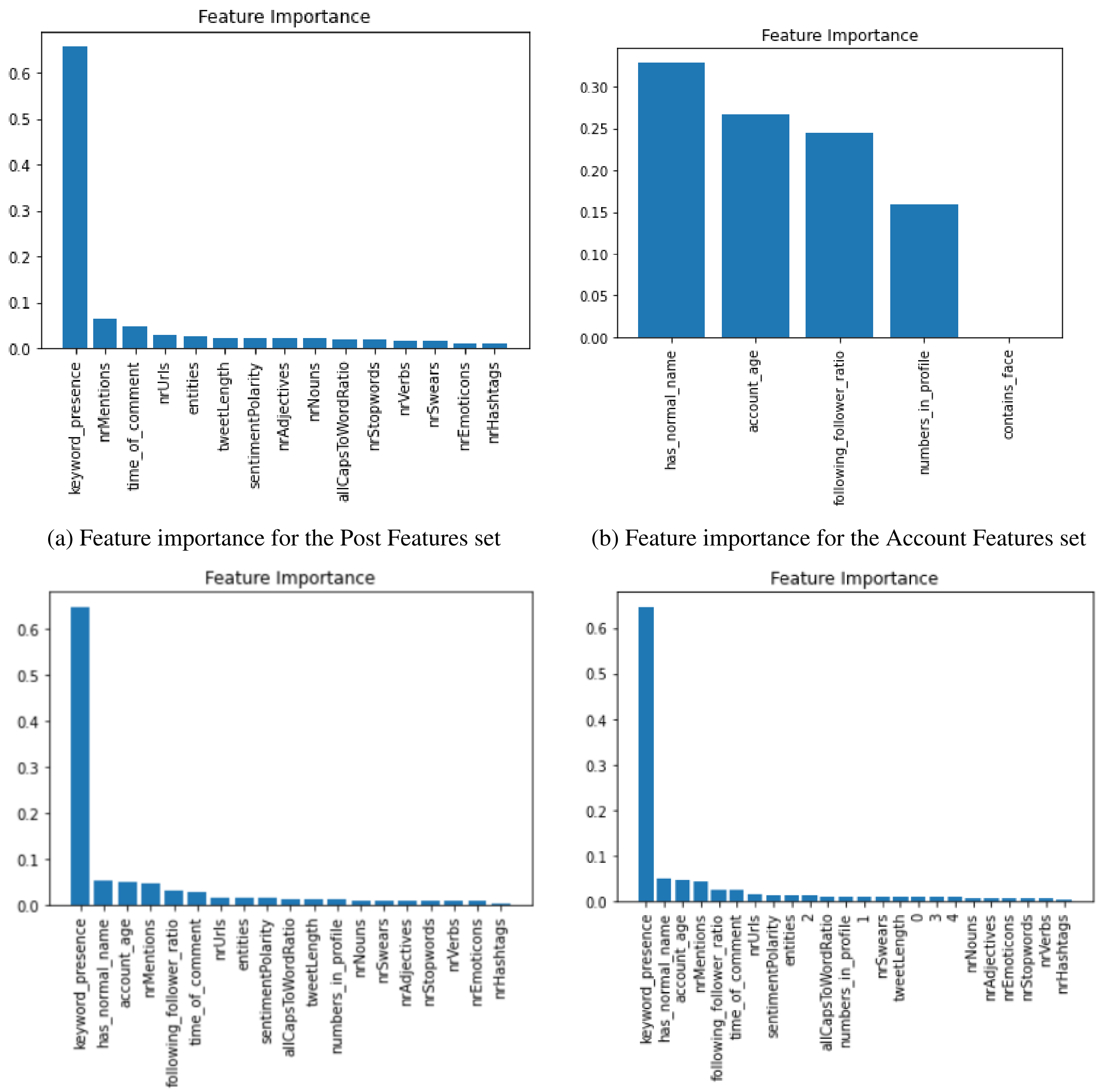
Table 6.7: Embedding Features Results

Paper	***AUC***	***Precision***	***Recall***	***Accuracy***
Still Out There : Modeling and Identifying Russian Troll Accounts on Twitter [57]	0.98	0.58	0.79	–
Identifying Russian Trolls on Reddit with Deep Learning and BERT Word Embeddings [144]	0.846	–	–	0.749
TrollBus – Classification Tweet by Tweet (Mean user classification)	**0.81**	**0.81**	**0.25**	–
A Holistic System For Troll Detection on Twitter[42]	0.74	0.70	–	–
TrollBus – Classification With All User Tweets Condensed	**0.72**	**0.57**	**0.38**	–
Antisocial Behaviours in Online Discussion Communities [20]	0.72	–	–	–
Troll Detection by Domain-Adapting Sentiment Analysis [124]	0.6	–	–	–
Hunting For Troll Comments in News community Forums [89]	–	–	–	76.46

Once the feature evaluation was performed, their influence on the models' decision process needed to be asserted. As all the models were generated through *sklearn*'s Random Forest algorithm, the function *feature importance* was used. This function displays the weight of each feature in the model's decision process. The models' feature importances are visible in figures 6.3a, 6.3b, 6.3c and 6.3d. In the Account Features model, it is visible how little impact the feature of whether a user contained a face had in the model decisions.

For all the feature groups, the presence of political insults is the most prevalent factor used by the models in labelling a user a troll or not. This is understandable, as the labelled trolls were selected on a basis of insulting other users and referring mainly to politics. By using these words more frequently, they actively incentivize speech polarization to achieve their goals. Secondly, the feature of whether a user contained a normal name or not was considered the second most important feature, showing that one of the features engineered specifically for the Portuguese reality was given importance by the model, as opposed to the other specific feature of the whether the user contained a face. A user's account age also proved important for the model, as well as the number of mentions, validating two of the hypothesis raised during the Exploratory Data Analysis phase of the project (Section 5.4), which linked trolls to a higher number of mentions (possibly to other trolls) and the fact that they display an accounts with a lower mean age.

The study of the feature importances allows to answer the research question of whether the features allow to extract complementary information: Accounts which use more words tied to politics (including political insults) have a higher probability of being trolls. It was also observed



(a) Feature importance for the Post Features set

(b) Feature importance for the Account Features set

(c) Feature importance for the Account and Post features

(d) Feature importance for the set containing Account, Post and Embedding Features

Figure 6.3: caption

Table 6.8: Results of the Best Models Obtained

Paper	***AUC***	***Precision***	***Recall***	***Accuracy***
Still Out There : Modeling and Identifying Russian Troll Accounts on Twitter [57]	0.98	0.58	0.79	–
A Holistic System For Troll Detection on Twitter[42]	0.96	–	0.96	–
TrollBus	**0.90**	**0.15**	**0.54**	–
Identifying Russian Trolls on Reddit with Deep Learning and BERT Word Embeddings [144]	0.846	–	–	.749
Antisocial Behaviours in Online Discussion Communities [20]	0.82	–	–	–

that features tied to a user's name, account's age and mentions to other users have a high degree of importance when labelling trolls in the Portuguese universe.

The most important feature are a mix of text (post) and account features, proving that all the models used have features considered important, proven in the results and following the tendency found in the state of the art of more inclusive feature sets having on average better results.

Chapter 7

Conclusion

This project aimed at analyzing political trolls on Twitter, the various efforts to detect them, and combining these efforts in a study to assert the best features to do so when applied to the Portuguese reality. Various feature sets were created, and in order to achieve these goals, an empirical study was performed on Portuguese Twitter data in order to analyze the different sets of features, and how they performed on this segment of Twitter. To aid in this study and to assure a way that the knowledge discovered could be applied, a software tool allowing troll detection in real time was developed, called Trollbus.

The analysis of the literature allowed to create three different feature sets regarding the Twitter accounts: Account, Post and Embedding Features. In the empirical study performed, several troll classification models are created, either using the groups by themselves or combined. The model considered best was created from a Random Forest algorithm with 1000 trees.

The data used for the project was collected using the SocialBus platform, and adapted for Trollbus. The labelling of accounts was done by asserting if an account displayed a list of pre-established characteristics, through a scoring function. The models obtained from the empirical study were then applied to TrollBus, creating an extension to SocialBus which classifies Twitter accounts as soon as they are collected, warning when a troll user is discovered, and updating a user's representation each time a new tweet from him is collected.

The final distribution of the collected data generated an imbalanced dataset, simulating the distribution and ratio of malicious to normal accounts found on Twitter. In order to train the models, the imbalance of the train set was reduced through the usage of SMOTE. In order to assert the performance of the models generated, due to the class distribution, the metric of AUC was prioritized. The other metrics calculated such as precision and recall were obtained from the threshold which yielded the highest MCC. The results obtained for the different feature sets (alone and combined) were not only in line with what was obtained in similar studies, but given the circumstances, yielded better results than some of them. The best result obtained was of 0.90 AUC, with a precision value of 0.15 and a recall value of 0.54, and was obtained from the combination of the Superficial Text and User features.

The analysis of the results proved it is possible to detect trolls in the Portuguese universe with any of the feature groups, meaning trolls can be detected from their account metadata, the numerical values of their text or an analysis of their language. The feature group with the best individual results was the Superficial Text Feature group, with 0.85 AUC, and the worst was the user features group, with 0.70 AUC. The analysis of these results prove that while it is possible to detect trolls without an analysis of their text, the absence of this type of features will result in worse results and a lesser quality user representation.

An analysis of the results and their features revealed that, by a large margin, the feature which influenced the models' decisions the most was the presence of political insults. This conclusion is considered logical, as trolls use an offensive and aggressive vocabulary, and it is of their interest to polarize discussions through the usage of such words to achieve their goals of sowing discord and normalizing tangencial discussion.

An observation regarding the models' results conclude that metrics as precision and recall did not achieve values as high as other studies found in the literature. For future work, it would be advantageous if the gathering of tweets was done through a longer period of time, larger and more balanced datasets, as well as, if possible, additional features.

One of the limitations of this work was the subjective labelling of accounts. Knowing this limitation *a priori*, only the accounts that displayed the majority of characteristics defined for a troll were labelled as positive, while accounts that were dubious were marked as negative. For future work, an in-depth study of the trolls in Portugal is suggested, as the labelling method used has the risk of having ignored a portion of the malicious userbase, such as fake personal profiles, and may not describe completely all the variants of the political trolls in Portugal. Also as future work, it is suggested that the models are re-tested after a significant time period in order to compare the results, as it is known the language and behavior of the users online changes over time.

Another limitation is the time required for TrollBus to analyze the accounts collected – SocialBus collects several tweets each second. Due to the computational weight of the models and representations, the TrollBus module takes a longer amount of time to classify each user, generating a queue of users to classify. The longer SocialBus is running, the bigger the queue and the longer it will take for TrollBus to classify each of the collected accounts, needing additional time to calculate the accounts' membership even after SocialBus is no longer running. The usage of a lighter, possibly less accurate model is suggested.

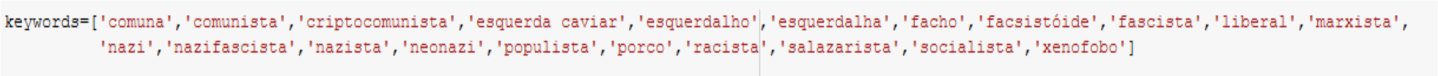
The research in this field has showed to be very recent, and when applied to Portugal, this study is the first of its kind. For this reason, this work expectes to promote future troll detection works in Portugal, giving a contribution of a real-time troll detection tool, several models, a dataset with its collection rules specified, the knowledge discovered about trolls in Portugal, and a survey on existing troll detection approaches.

With the conclusion of this dissertation, the study and experiment results were considered positive. The adaptation of established methods in troll detection, combined with a quality software development methodology and the usage of previous tools and projects by other researchers allowed to reach positive results in terms of AUC, effectively proving there are accounts considered trolls in Portugal, but they can be detected by machine learning approaches. From these results, a conclusion was reached that although it is possible to detect trolls, their definition is not always clear, therefore, the better the trolls are detailed, the better the results at detecting them are expected to be.

Appendix A

Annexes

A.1 Political Keywords



```
keywords=['comuna','comunista','criptocomunista','esquerda caviar','esquerdalho','esquerdalha','facho','facsistóide','fascista','liberal','marxista',
         'nazi','nazifascista','nazista','neonazi','populista','porco','racista','salazarista','socialista','xenofobo']
```

Figure A.1: Keywords used in the feature ”Contains Keywords”

A.2 Scoring function methods

Table A.1: Functions in class ScoreAssigner

Function	***Verifies If***
is_account_created _after(date)	An account is created after a date defined by the user
get_following_to_follower_ratio(value)	The ratio of accounts the user follows to the accounts that follow the user are over a value defined by the user
get_numbers_in_handle(value)	A user has the specified amount or greater number of digits in his handle
avatar_contains_face()	The user's avatar contains a recognizable face
has_normal_name()	The user's profile name is included in a specified list of names (e.g. Portuguese Civil Registry)
contains_keywords()	The user's tweets contain at least one of a series of words linked to insults and keywords (e.g. fascist)

Of the libraries required for the functions, all but one used the standard Python libraries [111]. The exception was the function **avatar_contains_face**, which used the Python library **Face Recognition** [1]. This external library uses a pre-trained Deep Learning model to identify if a face or faces are present in a picture, and if so, lists their coordinates in an array. Since SocialBus returns the link to every users' avatar picture, the face recognition function was applied to the image in that url, and if the array of coordinates for the face location was not empty, the function would return 0, otherwise it would return 1 and add that amount to the final score. This approach can be prone to false negatives, for example, through the usage of deepfakes (computer generated human faces) or even using photographs found online. Despite that, since there is a record of trolls presenting their profiles as entities and not users (e.g. right wing news, liberal activists) [5], and these not having faces in their avatars, this feature was considered relevant. The function **has_normal_name** verified whether a user had an identifiable first name in his profile. If the user didn't match any name on the list of all names specified, the function would return 1, as it was more likely that his profile either wasn't a personal one, used a fictitious name, or was a page. This function used in itself would not allow to take many conclusions, as it was prone to nicknames, misspells or foreign names. It was, however deemed as possibly relevant when used together with the remaining functions. The functions here detailed were later used for the Superficial Text and User Features sets (Section 4.5).

References

[1] Ageitgey. Face Recognition. *Github*, 2020.

[2] Josephine S Akosa. Predictive Accuracy : A Misleading Performance Measure for Highly Imbalanced Data. *SAS Global Forum*, 942:1–12, 2017.

[3] Saleh Alaliyat. Video -based Fall Detection in Elderly ' s Houses Video - based Fall Detection in Elderly ' s Houses Saleh Alaliyat. (January), 2015.

[4] Eijaz Allibhai. Hold-out vs. Cross-validation in Machine Learning. *Medium*, oct 2018.

[5] Adam Badawy, Aseel Addawood, Kristina Lerman, and Emilio Ferrara. Characterizing the 2016 Russian IRA influence campaign. *Social Network Analysis and Mining*, 9(1):1–11, 2019.

[6] Adam Badawy, Aseel Addawood, Kristina Lerman, and Emilio Ferrara. Characterizing the 2016 Russian IRA influence campaign. *Social Network Analysis and Mining*, 9(1):1–11, 2019.

[7] Yaman Barlas. Formal aspects of model validity and validation in system dynamics. *System Dynamics Review*, 12(3):183–210, 1996.

[8] Marco Baroni, Georgiana Dinu, and Germán Kruszewski. Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. *52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014 - Proceedings of the Conference*, 1:238–247, 2014.

[9] Alfredo Barroso. A direita aperta o cerco. *I*, jul 2019.

[10] Natasha Bertrand. Russia organized 2 sides of a Texas protest and encouraged 'both sides to battle in the streets'. *Business Insider*, nov 2017.

[11] Liliana Borges. Campanha ameaçada por multiplicação de trolls e bots. *Público*, sep 2019.

[12] Matko Bošnjak, Eduardo Oliveira, José Martins, Eduarda Mendes Rodrigues, and Luís Sarmento. TwitterEcho - A distributed focused crawler to support open research with twitter data. *WWW'12 - Proceedings of the 21st Annual Conference on World Wide Web Companion*, pages 1233–1239, 2012.

[13] C. E. Brodley and Mark A. Friedl. Decision tree classification of land cover from remotely sensed data. *Remote Sensing of Environment*, 61(3):399–409, 1997.

[14] Cody Buntain and Jennifer Golbeck. Automatically Identifying Fake News in Popular Twitter Threads. 2017.

[15] Diogo Camilo. Coronavírus: Governo aprova pedido de Marcelo para declarar estado de emergência. *Sabado*, mar 2020.

[16] Miguel Carvalho. Investigação: Os segredos do pregador Ventura. *VISÃO*, pages 1–10, may 2020.

[17] Paula Carvalho and E Mário J Silva. SentiLex-PT: Principais características e potencialidades. *Oslo Studies in Language*, 7(1), 2015.

[18] Cheng Chang, Yihong Zhang, Claudia Szabo, and Quan Z. Sheng. Extreme user and political rumor detection on twitter. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 10086 LNAI:751–763, 2016.

[19] Kai Chen, San Bruno, C A Us, Gregory S Corrado, San Francisco, Jeffrey A Dean, and Palo Alto. United States Patent -Computing numerical representations of words in a high dimensional space. 1(12), 2015.

[20] Justin Cheng, Cristian Danescu-Niculescu-Mizil, and Jure Leskovec. Antisocial behavior in online discussion communities. *Proceedings of the 9th International Conference on Web and Social Media, ICWSM 2015*, pages 61–70, 2015.

[21] Justin Cheng, Cristian Danescu-Niculescu-Mizil, and Jure Leskovec. Antisocial behavior in online discussion communities. *Proceedings of the 9th International Conference on Web and Social Media, ICWSM 2015*, pages 61–70, 2015.

[22] Davide Chicco and Giuseppe Jurman. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics*, 21(1):1–13, 2020.

[23] David A. Cieslak and Nitesh V. Chawla. Learning decision trees for unbalanced data. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5211 LNAI(PART 1):241–256, 2008.

[24] J. Clement. Leading countries based on number of Twitter users as of January 2020. *Statista*, jan 2020.

[25] Thomas Colson. Twitter threatens 'corrective action' against Boris Johnson's Conservatives party after it created a fake fact-checking service. *Business Insider*, nov 2019.

[26] Joana Gorjão Henriques; Vera Moutinho; Filipe Ribeiro; Dinis Correia. "Há uma montanha de discurso de ódio a erguer-se nos media e nas redes sociais". *Público*, mar 2019.

[27] Franklin Ph D. Logistic Regression : A Paradigm for Dichotomous Response Data. pages 1–5, 2014.

[28] José Mourão Da Costa. O Partido Nacional Renovador: A nova extrema-direita na democracia Portuguesa. *Analise Social*, 46(201):765–787, 2011.

[29] Jorge De-La-Pena-Sordo, Iker Pastor-Lopez, Xabier Ugarte-Pedrero, Igor Santos, and Pablo G. Bringas. Anomaly-based user comments detection in social news websites using troll user comments as normality representation. *Logic Journal of the IGPL*, 24(6):883–898, 2016.

[30] Jorge De-La-Peña-Sordo, Igor Santos, Iker Pastor-López, and Pablo G. Bringas. Filtering trolling comments through collective classification. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7873 LNCS:707–713, 2013.

[31] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. (Mlm), 2018.

[32] Imen Ouled Dlala, Dorra Attiaoui, Arnaud Martin, and Boutheina Ben Yaghlane. Trolls Identification within an Uncertain Framework. *Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI*, 2014-Decem(February):1011–1015, 2014.

[33] Stephan Dollberg, Tobias Langner, Jochen Seidel, and Roger Wattenhofer. The Metadata Troll Detector. page 14, 2015.

[34] Niklas Donges. The Logistic Regression Algorithm, 2018.

[35] Jocelyn D'Souza. An Introduction to Bag-of-Words in NLP, 2018.

[36] Fernando Esteves. André Ventura reconhece ao Polígrafo que promotor de desinformação lhe construiu o site do "Chega". *Polígrafo*, dec 2018.

[37] Expresso. Expresso é o jornal mais lido em Portugal. *Expresso*, jun 2019.

[38] Tom Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006.

[39] José Carlos Fernandes. Um mundo cheio de porcos fascistas? *Observador*, may 2017.

[40] Manuel Fernández-Delgado, Eva Cernadas, Senén Barro, and Dinani Amorim. Do we need hundreds of classifiers to solve real world classification problems? *Journal of Machine Learning Research*, 15:3133–3181, 2014.

[41] I V O Filipe and Valente Mota. Olhó-Passarinho: Uma Extensão do TweeProfiles Para Fotografias. 2014.

[42] Paolo Fornacciari, Monica Mordonini, Agostino Poggi, Laura Sani, and Michele Tomaiuolo. A holistic system for troll detection on Twitter. *Computers in Human Behavior*, 89:258–268, 2018.

[43] Patxi Galán-García, José Gaviria De La Puerta, Carlos Laorden Gómez, Igor Santos, and Pablo García Bringas. Supervised machine learning for the detection of troll profiles in twitter social network: Application to a real case of cyberbullying. *Logic Journal of the IGPL*, 24(1):42–53, 2014.

[44] João Gama, André Ponce de Leon Carvalho, Katti Facelli, Ana Carolina Lorena, and Márcia Oliveira. *Extração de Conhecimento de Dados Data Mining*. 2017.

[45] Vaishali Ganganwar. An overview of classification algorithms for imbalanced datasets. *Southeast Asian Journal of Tropical Medicine and Public Health*, 2(4):94–98, 2012.

[46] Genesis. Pros and Cons of K-Nearest Neighbors, 2018.

[47] Saptarshi Ghosh, Muhammad Bilal Zafar, Parantapa Bhattacharya, Naveen Sharma, Niloy Ganguly, and Krishna Gummadi. On sampling the wisdom of croGhosh, S., Zafar, M. B., Bhattacharya, P., Sharma, N., Ganguly, N., & Gummadi, K. (2013). On sampling the wisdom of crowds. 1739–1744. https://doi.org/10.1145/2505515.2505615wds. pages 1739–1744, 2013.

[48] Glemaitre. scikit-learn-contrib / imbalanced-learn. *Github*, 2020.

[49] Vairaprakash Gurusamy, Subbu Kannan, S Kannan, and K Nandhini. Performance Analysis: Stemming Algorithm for the English Language. *IJSRD-International Journal for Scientific Research & Development|*, 5(August):2321–0613, 2017.

[50] Albert Haque. Twitch Plays Pokemon, Machine Learns Twitch: Unsupervised Context-Aware Anomaly Detection for Identifying Trolls in Streaming Data. (May):1–9, 2019.

[51] João Henrique and Alves Pereira. TweeProfiles2 : real-time detection of spatio-temporal patterns in Twitter. 2014.

[52] Andrew Higgins. Effort to Expose Russia's 'Troll Army' Draws Vicious Retaliation. *New York Times*, may 2016.

[53] Tin Kam Ho. Random decision forests. *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, 1:278–282, 1995.

[54] Matthew Honnibal. Embed, encode, attend, predict: The new deep learning formula for state-of-the-art NLP models, 2016.

[55] Matthew Honnibal. Example 2: Hierarchical Attention Networks for Document Classification. *explosion.ai*, 2016.

[56] David a Hull and Gregory Grefenstette. A detailed analysis of English stemming algorithms. *Rank Xerox Research Centre*, 73(MLTT-023):381–386, 1996.

[57] Jane Im, Eshwar Chandrasekharan, Jackson Sargent, Paige Lighthammer, Taylor Denby, Ankit Bhargava, Libby Hemphill, David Jurgens, and Eric Gilbert. Still out there: Modeling and Identifying Russian Troll Accounts on Twitter. 2019.

[58] Tania Ionin, Maria Luisa Zubizarreta, and Salvador Bautista Maldonado. Sources of linguistic knowledge in the second language acquisition of English articles. *Lingua*, 118(4):554–576, 2008.

[59] Filip Jansson, Oskar Casselryd, and Filip Jansson. Troll detection with sentiment analysis and nearest neighbour. 2017.

[60] László A. Jeni, Jeffrey F. Cohn, and Fernando De La Torre. Facing imbalanced data - Recommendations for the use of performance metrics. *Proceedings - 2013 Humaine Association Conference on Affective Computing and Intelligent Interaction, ACII 2013*, (September):245–251, 2013.

[61] Beel Joeran, Corinna Breitinger, Bela Gipp, and Stefan Langer. Research-paper recommender systems: a literature survey. *International Journal on Digital Libraries*, 17(4):305–338, 2015.

[62] Jeremy Jordan. No Title. *Support vector machines*, 2017.

[63] Will Koehrsen. Random Forest Simple Explanation, 2017.

[64] Will Koehrsen. Neural Network Embeddings Explained. *Medium*, oct 2018.

[65] Kamran Kowsari, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura Barnes, and Donald Brown. Text classification algorithms: A survey. *Information (Switzerland)*, 10(4), 2019.

[66] Hans Peter Kriegel, Erich Schubert, and Arthur Zimek. The (black) art of runtime evaluation: Are we comparing algorithms or implementations? *Knowledge and Information Systems*, 52(2):341–378, 2017.

[67] M Kukar and Igor Kononenko. Cost-sensitive learning with neural networks. *13th European Conference on Artificial Intelligence*, pages 445–449, 1998.

[68] Srijan Kumar, Francesca Spezzano, and V. S. Subrahmanian. Accurately detecting trolls in Slashdot Zoo via decluttering. *ASONAM 2014 - Proceedings of the 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, (Asonam):188–195, 2014.

[69] Gustavo Alexandre Teixeira Laboreiro. Noise reduction and normalization of microblogging messages. 2018.

[70] Peter A. Lachenbruch and M. Ray Mickey. Estimation of Error Rates in Discriminant Analysis. *Technometrics*, 10(1):1–11, 1968.

[71] Nada Lavra??, Peter Flach, and Blaz Zupan. Rule evaluation measures: A unifying view. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 1634:174–185, 1999.

[72] W. Philip Kegelmeyer Nitesh V. Chawla Kevin W. Bowyer Lawrence O. Hall. snopes.com: Two-Striped Telamonia Spider. *Journal of Artificial Intelligence Research*, 2009(Sept. 28):321–357, 2006.

[73] Omer Levy and Yoav Goldberg. Neural word embedding as implicit matrix factorization. *Advances in Neural Information Processing Systems*, 3(January):2177–2185, 2014.

[74] Elizabeth D. Liddy. Natural Language Processing. In Encyclopedia of Library and Information Science. *Marcel Decker, Inc.*, pages 1–15, 2001.

[75] Rute Lourenço. CM lidera venda de jornais com 61,8% do mercado. *Correio da Manhã*, nov 2019.

[76] Agência Lusa. Extrema-direita entra pela primeira vez no parlamento português em democracia. *Observador*, oct 2019.

[77] Agência Lusa. Fake News: Director do Polígrafo diz que há dificuldade em distinguir a verdade da mentira. *Diário de Notícias*, oct 2019.

[78] Agência Lusa. Portugal continua a ser dos países onde menos se paga por notícias online. *Notícias ao Minuto*, jun 2020.

[79] Edward Ma. NLP Pipeline: Stop words (Part 5), 2018.

[80] Akshat Maheshwari. Report on Text Classification using CNN, RNN & HAN. *Medium*, jul 2018.

[81] Pedro Maia, André; Soares, Carlos; Abreu. TweeProfiles3 : visualização de padrões espacio-temporais no Twitter. 2015.

[82] Oded Maimon and Lior Rokach. *Data Mining and Knowledge Discovery*. 2nd edition, 2005.

[83] R. B. Marimont and M. B. Shapiro. Nearest neighbour searches and the curse of dimensionality. *IMA Journal of Applied Mathematics (Institute of Mathematics and Its Applications)*, 24(1):59–70, 1979.

[84] Negative-sampling Word-embedding Method, Yoav Goldberg, Omer Levy, Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. word2vec Explained : Deriving Mikolov et al .' s Negative Sampling Word-Embedding Method. (2):1–5, 2014.

[85] Michaelg2015. Logistic Regression, 2015.

[86] Bruno Miguel and Alves Vieira. Journalism 3 . 0 : Multidimensional Cluster Visualization and Labelling on Twitter Data for Data Journalism. 2016.

[87] Luís Miguel and Azevedo Pereira. TweeProfiles4 : a weighted multidimensional stream clustering algorithm. 2015.

[88] Todor Mihaylov, Georgi D. Georgiev, and Preslav Nakov. Finding opinion manipulation trolls in news community forums. *CoNLL 2015 - 19th Conference on Computational Natural Language Learning, Proceedings*, (September):310–314, 2015.

[89] Todor Mihaylov and Preslav Nakov. Hunting for Troll ' s Comments in News Community Forums. pages 1–5, 2014.

[90] Tomas Mikolov, Greg Corrado, Kai Chen, and Jeffrey Dean. Vector Space. pages 1–12.

[91] Monika Myers, Jolie and Evstatieva. Meet The Activist Who Uncovered The Russian Troll Factory Named In The Mueller Probe : Parallels : NPR. *Npr*, mar 2018.

[92] Marwa Naili, Anja Habacha, Ben Ghezala, and Ben Ghezala. Comparative word Comparative study study of of segmentation word embedding embedding methods methods in in topic topic Comparative study of segmentation word embedding methods in topic segmentation. *Procedia Computer Science*, 112:340–349, 2017.

[93] Sarang Narkhede. Understanding AUC - ROC Curve. *Medium*, jun 2018.

[94] Garrett Nicolai and Grzegorz Kondrak. Does the phonology of L1 show up in L2 texts? *52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014 - Proceedings of the Conference*, 2:854–859, 2014.

[95] Nltk. NLTK 3.5 documentation, 2020.

[96] David Opitz and Richard Maclin. Popular Ensemble Methods: An Empirical Study. 1.

[97] Thais Mayumi Oshiro and Pedro Santoro Perez. How Many Trees in a Random Forest ? How Many Trees in a Random Forest ? (January):154–168, 2016.

[98] B. P. Pande and H. S. Dhami. Application of Natural Language Processing Tools in Stemming. *International Journal of Computer Applications*, 27(6):14–19, 2011.

[99] Saurabh Pal. Implementing Word2Vec in Tensorflow. *Medium*, jun 2019.

[100] Paulo Pena. Site do partido de André Ventura foi criado por autor de desinformação ligado ao e-toupeira. *Diário de Notícias*, dec 2018.

[101] Phillipperemy. name-dataset. *Github*, 2020.

[102] Joseph-Maria Piel. Origens e estruturação histórica do léxico português. *Estudos de Linguística Histórica Galego-Portuguesa*, (1976):9–16, 1989.

[103] Joël Plisson, Nada Lavrac, and Dr. Dunja Mladenić. A rule based approach to word lemmatization. *Proceedings of the 7th International Multiconference Information Society (IS'04)*, pages 83–86, 2004.

[104] Ben Popken. Twitter deleted 200,000 Russian troll tweets. Read them here. *NBC*, feb 2018.

[105] Porto Editora. Palavra Do Ano. *Porto Editora, Infopédia*, dec 2011.

[106] David M W Powers. Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation David. (December), 2007.

[107] Priberam. Dicionário Priberam da Língua Portuguesa. *Dicionário Priberam da Língua Portuguesa*, 2020.

[108] Python. GetOldTweets3 PyPl. *PyPl*, 2019.

[109] Python. WordCloud PyPl, 2019.

[110] Python. langdetect Pypl. *PyPl*, 2020.

[111] Python. The Python Standard Library. *Python*, 2020.

[112] Miguel Sozinho Ramalho. *Towards Holistic Political Manipulation Detection on Twitter*. Masters thesis, Faculdade de Engenharia da Universidade do Porto, 2020.

[113] MIT Technology Review. Data Mining Has Revealed Previously Unknown Russian Twitter Troll Campaigns. *Medium*, nov 2018.

[114] Tomy Antunes Rodrigues. RetweetPatterns : detection of spatio- temporal patterns of retweets. (January), 2014.

[115] LateshMalik Rushi Longadge,Snehlata S. Dongre. Class Imbalance Problem in Data Mining: Review. *International Journal of Computer Science and Network (IJCSN)*, 2(1):e256, 2013.

[116] Gustavo Sampaio. UEFA sugeriu que atribuição de lugares nas competições europeias "seja feita pela classificação final em 2018/19"? *Polígrafo*, may 2020.

[117] Sapo. Portugueses continuam a gostar do Facebook e a ligar pouco ao Twitter. jun 2017.

[118] Yutaka Sasaki. The truth of the F-measure. *Teach Tutor mater*, pages 1–5, 2007.

[119] Data School. Simple guide to confusion matrix terminology, 2015.

[120] Scikit-learn. sklearn.feature_selection.RFE, 2019.

[121] Scikit-learn. scikit-learn. *scikit-learn*, 2020.

[122] Jim ; Sciutto and Mary Ilyushina. Exclusive: Putin's 'chef,' the man behind the troll factory, 2017.

[123] Chun Wei Seah, Hai Leong Chieu, Kian Ming A. Chai, Loo Nin Teow, and Lee Wei Yeong. Troll detection by domain-adapting sentiment analysis. *2015 18th International Conference on Information Fusion, Fusion 2015*, pages 792–799, 2015.

[124] Chun Wei Seah, Hai Leong Chieu, Kian Ming A. Chai, Loo Nin Teow, and Lee Wei Yeong. Troll detection by domain-adapting sentiment analysis. *2015 18th International Conference on Information Fusion, Fusion 2015*, pages 792–799, 2015.

[125] Michael Shermer. How the Survivor Bias Distorts Reality, 2014.

[126] Francisco Soares. André Ventura diz que se ganhar eleições "ofender polícias vai dar mesmo prisão". *Sol*, jun 2020.

[127] Spacy. spaCy. *Spacy.com*, 2020.

[128] Spacy. spaCy models Portuguese. *Spacy.com*, 2020.

[129] Spacy. TextCategorizer. *Spacy.com*, 2020.

[130] Tim Stevens. Twitch Plays Pokemon: In conversation with the phenomenon's creator (Q&A), 2014.

[131] Emily Stewart. How China used Facebook, Twitter, and YouTube to spread disinformation about the Hong Kong protests. *Vox*, aug 2019.

[132] Philip H. Swain and Hans Hauska. Decision Tree Classifier: Design and Potential. *IEEE Trans Geosci Electron*, GE-15(3):142–147, 1977.

[133] Nina Tahmasebi and Thomas Risse. On the uses of word sense change for research in the digital humanities. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 10450 LNCS:246–257, 2017.

[134] Techopedia. K-Nearest Neighbor (K-NN). *Techopedia.*

[135] Mohamad Ali Torkamani and Daniel Lowd. On robustness and regularization of structural support vector machines. *31st International Conference on Machine Learning, ICML 2014*, 3:1989–1999, 2014.

[136] Craig Trim. Language Processing, 2013.

[137] Joseph Turian, Lev Ratinov, and Yoshua Bengio. Word representations: A simple and general method for semi-supervised learning. *ACL 2010 - 48th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, (January 2010):384–394, 2010.

[138] Twitter. Carreers Twitter.

[139] Twitter. About - Twitter, 2019.

[140] Twitter. Rate Limiting, 2019.

[141] Yuli Vasiliev. *Natural Language Processing with Python and spaCy*. no starch press, San Francisco, 1st edition, 2020.

[142] E Versi. "Gold standard" is an appropriate term. *BMJ. 305 (6846): 187*, 1992.

[143] Shoujin Wang, Wei Liu, Jia Wu, Longbing Cao, Qinxue Meng, and Paul J. Kennedy. Training deep neural networks on imbalanced data sets. *Proceedings of the International Joint Conference on Neural Networks*, 2016-Octob:4368–4374, 2016.

[144] Henry Weller and Jeffrey Woo. Identifying Russian Trolls on Reddit with Deep Learning and BERT Word Embeddings. (9):1–11.

[145] Jeremy B White. Twitter warns 1.4 million users they interacted with accounts linked to Russian 'troll farm'. *Independent*, feb 2018.

[146] Oscar Winberg. Insult Politics: Donald Trump, Right-Wing Populism, and Incendiary Language. *European journal of American studies*, 12(2), 2017.

[147] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. *2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2016 - Proceedings of the Conference*, pages 1480–1489, 2016.

[148] Raymond T. Yeh and Pamela Zave. Specifying Software Requirements. *Proceedings of the IEEE*, 68(9):1077–1085, 1980.

[149] Tian Zhang, Raghu Ramakrishnan, and Miron Livny. BIRCH: An Efficient Data Clustering Method for Very Large Databases. 1:103–114.

[150] Yin Zhang and Rong Jin. Understanding Bag-of-Words Model : A Statistical Framework.