LIARDETECTOR: A LINGUISTIC-BASED APPROACH FOR IDENTIFYING FAKE NEWS

THAIS GOMES DE ALMEIDA

LIARDETECTOR: A LINGUISTIC-BASED APPROACH FOR IDENTIFYING FAKE NEWS

Dissertação apresentada ao Programa de Pós-Graduação em Informática do Instituto de Computação da Universidade Federal do Amazonas como requisito parcial para a obtenção do grau de Mestre em Informática.

Orientador: Eduardo Freire Nakamura
Coorientadora: Fabíola Guerra Nakamura

Manaus
Abril de 2019

THAIS GOMES DE ALMEIDA

LIARDETECTOR: A LINGUISTIC-BASED APPROACH FOR IDENTIFYING FAKE NEWS

Dissertation presented to the Post Graduation Program in Informatics of the Federal University of Amazonas in partial fulfillment of the requirements for the degree of Master in Informatics.

Advisor: Eduardo Freire Nakamura
Co-Advisor: Fabíola Guerra Nakamura

Manaus
April 2019

Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

Almeida, Thais Gomes de
A447l LIARDetector: A linguistic-based approach for identifying fake news / Thais Gomes de Almeida. 2019
86 f.: il. color; 31 cm.

Orientador: Eduardo Freire Nakamura
Coorientadora: Fabíola Guerra Nakamura
Dissertação (Mestrado em Informática) - Universidade Federal do Amazonas.

1. Notícias Falsas. 2. Classificação. 3. Representação de Dados. 4. Aprendizagem Supervisionada. I. Nakamura, Eduardo Freire II. Universidade Federal do Amazonas III. Título

PODER EXECUTIVO
MINISTÉRIO DA EDUCAÇÃO
INSTITUTO DE COMPUTAÇÃO

PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

UFAM

FOLHA DE APROVAÇÃO

"LIARDETECTOR: A LINGUISTIC-BASED APPROACH FOR IDENTIFYING FAKE NEWS"

THAIS GOMES DE ALMEIDA

Dissertação de Mestrado defendida e aprovada pela banca examinadora constituída pelos Professores:

Prof. Eduardo Freire Nakamura - PRESIDENTE

Prof. Altigran Soares da Silva - MEMBRO INTERNO

Profa. Juliana Freire - MEMBRO EXTERNO

Manaus, 18 de Abril de 2019

Dedico esta dissertação a minha amada mãe Zioneia Gomes de Almeida, uma mulher simples e feliz, que sempre tem os melhores conselhos para oferecer e possui a capacidade de deixar a vida mais leve com sua risada.

Acknowledgments

I wish to start by thanking my parents who have always made efforts to ensure that I had access to education; that even without understanding exactly what I do, they have always supported me emotionally and given me advices. Special thanks to my mother Zioneia Gomes, who always makes me smile and for her unconditional love.

I would like to thank my sisters, Anne Almeida and Lahis Almeida, who have encouraged me, offered emotional support and academic tips; and for being present in my life, even though we were in different countries. To my nephews, Gustavo Brito and Miguel Brito, who illuminate the lives of those around them with a simple smile.

I would like to thank my advisors, Eduardo Nakamura and Fabíola Nakamura, and Professor Altigran Soares, for the opportunities and initial confidence they had showed me. Professor Juliana Freire for all the learning I gathered during the research internship at New York University. Thanks also to CAPES for their support through the Master's scholarship.

I would like to thank my friends Maísa Brenda, Erick Frota, Alice Adativa, Bruno Ábia, Mariana Tonin, Victória Aires, and Raoni Lourenço for all the emotional support and encouragement during these two years of research; and because they are inspiring human beings.

Resumo

Devido à infraestrutura da Web existente e à popularidade das plataformas de mídia sociais, é fácil compartilhar informações de forma massiva. Embora esse cenário online traga benefícios para a sociedade, ele também favorece que grupos maliciosos propaguem desinformação (notícias falsas) na Web, causando danos que vão desde afetar a reputação de entidades públicas (empresas, celebridades) a interferir em processos políticos. Neste trabalho, propomos uma nova abordagem de classificação baseada em padrões linguísticos para identificar notícias falsas. Tal abordagem reduz a dimensionalidade do espaço de características ao codificar distribuições de probabilidade de *tokens* (por exemplo, palavras) como valores de divergência e entropia. Nós descrevemos resultados experimentais, usando vários conjuntos de dados, que mostram que nossa abordagem é uma solução que melhora tanto a eficácia, quanto eficiência de modelos de aprendizagem. Em comparação com o *baseline*, nossa abordagem usa quatro ordens de magnitude menos atributos e obtém um ganho de até 74,3% de eficácia (Medida-F).

Palavras-chave: Notícias Falsas, Classificação, Representação de Dados, Aprendizagem Supervisionada.

Abstract

Due to the existing Web infrastructure and the popularity of social media platforms, it is easy to share information in large scale. Although this online scenario brings benefits to the society, it also favors malicious groups that propagate misinformation (e.g., alternative facts, fake news) on the Web, causing damages that range from affecting the reputation of public entities (companies, celebrities) to interfering on political process. In this work, we propose a novel classification approach based on linguistic patterns for identifying fake news. Our approach reduces the dimensionality of the feature space by encoding probability distributions of tokens (e.g., words) as Shannon entropy and Jensen-Shannon divergence values. We report experimental results using multiple data sets, which show that our approach is a win-win solution that improves efficacy and efficiency. Compared to the baseline, our approach uses four orders of magnitude less features, and achieve a gain up to 74.3% of F1-score.

Keywords: Fake news, Classification, Data Representation, Supervised Learning.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

Although the spread of misinformation by media outlets is not a new phenomenon (Marcus, 1992), it has gained substantial attention in the last years. Due to wide circulation in online platforms, contemporary fake news have a large coverage and spread faster among news consumers. Hence, they have the potential of causing damages that range from affecting the reputations of public entities (companies, celebrities) to interfering on political processes (Blake, 2018; Tardáguila et al., 2018).

Unlike traditional media print that has processes of fact-checking and editorial judgment, users of online media can write news (without no significant third-part filtering) and reach as many readers as famous mainstreams sites like Fox News and the New York Times (Allcott and Gentzkow, 2017). Behind the main motivations that leads on the fabrication of fake stories, we highlight the financial and political gains. While web sites can circulate sensationalist content designed to attract online advertising revenue, they also can publish fabricated content that supports political propaganda in favor of some party or candidate (Bakir and McStay, 2018; Braun and Eklund, 2019). Given that 62 percent of adults in the US consume news from social media (Gottfried and Shearer, 2016) and many who see fake news stories report that they believe them (Silverman and Singer-Vine, 2016), these platforms have also become

a target for information operations (Narayanan et al., 2018; Lazer et al., 2018).

While fake news have attracted substantial attention, the problem is not well understood (Lazer et al., 2018). Moreover, even humans can have difficulty discerning between fake and real news (Domonoske, 2016; Edkins, 2016). The challenges of fact checking are both qualitative and quantitative. In addition to the difficulty inherent in parsing and cross-referencing conflicting sources and claims, the ease of publishing content on the Web has led to orders of magnitude increase in the volume of news sites and content. Automated methods that identify potential fake news and unreliable news sources can aid manual fact checking by providing contextual information and limiting the volume of content that the human fact-checker needs to consider. Such methods can also help us better understand the ecosystem of fake news: where they start, how they propagate, and how to counter their effects.

This works aims to provide an automated method to identify fake news in a timely manner in order to minimize its effects on society, and to help preserving the society trust on facts.

1.1 Problem

In this work, we study the problem of detecting fake news published on online platforms. Given a news document with textual metadata (headline and body text) N, our goal is to determine whether N is likely to contain fake or real news.

Previous works have attempted to explore linguistic patterns on fake news by representing them primarily through n-grams. This feature representation method leads to a high dimensional sparse matrix that can be restricted by memory limits and contribute in the increase of training and testing times of machine learning algorithms. To address these limitations, we propose a novel classification approach that encodes n-grams into entropy and divergence values. Besides, this dimensionality reduction aspect, our approach also incorporates distinct linguistic set of features – morphological,

psycholinguistic and readability patterns.

1.2 Research Hypotheses

In this work, we define the following research hypotheses:

- A classification approach that considers distinct sets of linguistic-based features (morphological, readability, psychological and stylometric) can lead to accurate prediction values.
- Applying information theory quantifiers to encode high dimensional term-matrix can improve both effectiveness and efficacy results of learning models.
- Building learning models with features extracted from news articles headlines can lead to competitive effectiveness results, when compared with models build on news body text.

1.3 Objectives

The main objective of this work is to propose and demonstrate the effectiveness and efficiency of a new approach, that incorporates distinct linguistic-based features set, in the task of identifying news reliability.

The specific objectives include:

1. Assessing the feasibility of linguistic-based patterns to represent news documents.
2. Proposing a classification approach that combines relevant linguistic patterns for identifying news reliability.
3. Demonstrating through study cases the effectiveness and efficiency of the proposed approach when compared to the state of the art.

1.4 Dissertation Structure

Chapter 2 describes the concepts needed to understand the general aspects upon the proposed approach is build, as well as the baselines. Chapter 3 presents a brief review of the recent literature on the detection of false news. In addition, it presents the databases publicly available for this task. Chapter 4 details our approach. Chapter 5 presents the experimental evaluation of our approach, as well as the results obtained. Finally, Chapter 6 presents our final conclusions, limitations, and future directions.

Chapter 2

Theoretical Background

In this chapter, we present the conceptual foundations needed for the understanding of our work. We start by discussing fake news definitions used in the literature, and then we describe several representation methods that are often applied to map text documents to numerical values. We also discuss supervised machine learning algorithms that we employ in our experiments (see Chapter 5), as well as evaluation measures.

2.1 Fake News

In recent literature, there is no broadly accepted formal definition for the term (*fake news*). The papers which describe this concept adopt the following definition (Allcott and Gentzkow, 2017; Shu et al., 2017a):

Definition 1 *: Fake news are news articles whose content is intentionally and verifiably false.*

Considering the Definition 1, the following categories of news are not considered to be false: (i) satirical news that has no intention of deceiving its readers and that are mistakenly perceived as factual; (ii) rumors that do not originate through news from

events; (iii) conspiracy theories, since they are difficult to verify as false or true; and (iv) misinformation that is created involuntarily.

Although such categories are not included in the previous definition, some of them promote information that has the potential to cause harm to society. An example is the conspiracy theory in which the global anti-vaccination movement is based. Supporters of this movement do not vaccinate their children because they claim, among other reasons, that the side effects of vaccines are worse than the disease they prevent. As a result, countries such as Italy, Romania, and Germany have recently had disease outbreaks (e.g., measles) that had been long considered eradicated (Pains, 2018).

Based on the lack of coverage of definition 1, we elaborate a broader definition in terms of news categories:

Definition 2 *: Fake news items are news articles whose content promotes misinformation regardless of the implicit intent of their publisher, and is not verifiably true.*

Definition 2 covers the misinformation categories *ii*, *iii*, *iv*, since it considers the author's intent and verifiability of the news. That is, the news is false whether it was intentionally created or not; unless it is clear that the published news is fabricated. In the latter case, the person responsible for misinformation is the reader or consumer of the news. Regarding the verifiability, we assume as fake the news where it is not possible to verify reliability.

Therefore, we consider as fake those news that promote misinformation, except for news coming from satirical sites. This exception is motivated by the fact that satiric sites make clear in their *homepages* that the contents conveyed by them have the purpose of amusing their readers through irony, sarcasm and other figures of language.

2.2 Data representation

Machine learning algorithms take as input a set of data (e.g., images, audio, text) that has been maped into feature vectors in a a multidimensional space. In this section, we will discuss approaches commonly used in the literature for performing this mapping of textual data into numerical vectors. In each subsection, we will present the correspondent feature vectors of the documents in Figure 2.1, according to the data representation technique described.

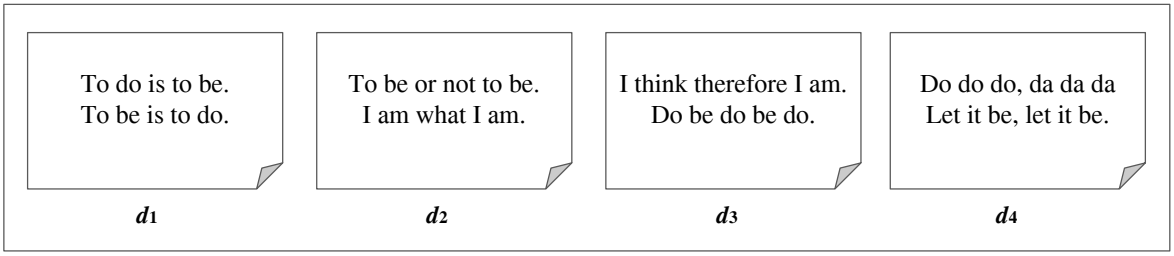


Figure 2.1: Example of a data collection (adapted from Baeza-Yates and Ribeiro-Neto (2013)).

2.2.1 Bag of Words (N-grams)

In bag of words representation, documents are mapped to feature vectors belonging to a high-dimensional space, which is determined by the vocabulary size considered. Each element of these vectors consists of the word frequency w_i in document dj (see Table 2.1).

Table 2.1: Feature vectors of documents in Figure 2.1 obtained through *bag of words*. The numerical values correspond to word frequencies.

Doc.	**Vocabulary Words**												
	am	be	da	do	is	it	let	not	or	to	what	think	therefore
d_1	0	2	0	2	2	0	0	0	0	4	0	0	0
d_2	2	2	0	0	0	0	0	1	1	2	1	0	0
d_3	1	2	0	3	0	0	0	0	0	0	0	1	1
d_4	0	2	3	3	0	2	2	0	0	0	0	0	0

If the elements of feature vectors represents the frequency of only one term, it is said that the bag of words is based on unigrams; if they represent two terms, bigramas and n-grams. This data representation loses information about the order of terms in the documents, that is, spatial information about the relationship between terms are not captured.

2.2.2 Term Frequency and Inverse Document Frequency

Term Frequency (TF) and Inverse Document Frequency (IDF) are the key concepts of the most popular weighting technique in the field of Information Retrieval, called TF-IDF (Baeza-Yates and Ribeiro-Neto, 2013). Since in a text some terms have a greater importance than others, the TF-IDF weights help defining terms that are relevant in a document.

Analogously to the BOW representation, TF-IDF projects documents in a mulidimensional space proportional to the vocabulary size. However, in TF-IDF representation documents are mapped into feature vectors, in which the elements are the multiplication product of TF by IDF. Thus, each document d corresponds to a vector c_1,...c_m, where c_1 is the weighted frequency of a term i in document d, normalized by the frequency of term i in the data set

$$\mathbf{c_i} = \begin{cases} (1 + \log f_{i,d}) \times \log \frac{N}{n_i} & , \text{ se } f_{i,d} > 0 \\ 0 & , \text{ se } f_{i,d} \leq 0 \end{cases}. \tag{2.1}$$

In the above equation, the term frequency (first element of the product) is in the logarithmic expression because it makes the weights directly comparable to the weights of the IDF measure. Table 2.2 shows examples of feature vectors that were maped by TF-IDF weighting scheme.

Table 2.2: Feature vectors of documents in Figure 2.1 obtained through *bag of words*. The numerical values correspond to TF-IDF weights.

Doc.	**Vocabulary Words**												
	am	be	da	do	is	it	let	not	or	to	what	think	therefore
d_1	0.00	0.25	0.00	0.31	0.48	0.00	0.00	0.00	0.00	0.77	0.00	0.00	0.00
d_2	0.52	0.34	0.00	0.00	0.00	0.00	0.00	0.33	0.33	0.52	0.33	0.00	0.00
d_3	0.29	0.38	0.00	0.70	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.36	0.36
d_4	0.00	0.22	0.64	0.41	0.00	0.42	0.42	0.00	0.00	0.00	0.00	0.00	0.00

2.2.3 Part-of-Speech Tagging

A part of speech is a homogeneous category of words that presents similar properties in gramatical sentences (Fisicaro and Gauvin, 2018). The process of assigning a part of speech category (POSTAG) to a word is called *part-of-speech tagging*. This assignment considers both the word definition and context (relationship between adjascent words in a sentence). In POSTAG representation, each document d_j is mapped to a feature vector in which the elements are frequencies of a particular part of speech tag (see Table 2.3).

Table 2.3: Feature vectors of documents in Figure 2.1 obtained through *bag of words*. The numerical values correspond to POSTAG frequency in the documents.

Doc.	**Part of speech category**									
	TO	VB	VBZ	CC	PRP	RB	VBP	WP	NN	VBN
d_1	4	4	2	0	0	0	0	0	0	0
d_2	2	2	0	1	2	1	2	1	0	0
d_3	0	4	0	0	2	2	2	0	0	0
d_4	0	8	0	0	2	0	0	0	1	1

In Table 2.3, TO corresponds to the part of speech category *"To" as preposition/infinitive, superlative*; VB to *verb, base form*; VBZ to *verb, present tense, 3rd singular* ; CC as *conjunction, coordinating*; PRP as *personal pronoun*; RB as *adverb*; VBP as *verb, present tense, not 3rd sing*; WP as *wh-pronoun*; NN as *noun, common, singular or mass*; and VBN as *verb, past participle*.

2.3 Machine Learning

Machine Learning (ML) is defined as the set of computational methods that use experience (past input information) to improve the effectiveness of predictions (Mohri et al., 2012). In *unsupervised* learning, the ML models have as input a set of unlabeled data and they must perform predictions considering all the data set. There is no distinction between training and test data. The goal of these models is to reduce the dimensionality of the ML problem or clustering documents based on similar patterns (Shalev-Shwartz and Ben-David, 2014). On the other hand, *supervised* learning models take an annotated set of training samples and they must perform inferences about samples with unknown labels. This type of learning is commonly associated with classification, regression, and ranking problems (Mohri et al., 2012). In this work, we consider the automatic fake-news detection as a supervised learning problem, more specifically, a classification problem (the output is discrete). In what follows, we describe the following supervised ML algorithms: K-Nearest Neighbor, Gaussian Naive Bayes, Support Vector Machine and Random Forest.

2.3.1 *k-Nearest Neighbor*

The k-Nearest Neighbor (kNN) is a classification algorithm on-demand or lazy (Baeza-Yates and Ribeiro-Neto, 2013). Lazy learning algorithms do not build a classification model *a priori*, thus the inference is performed only when a new document d_j is submitted to the algorithm. To classify a class-unknown document d_j, the kNN classifier algorithm ranks the document's neighbors among the training document vectors, and uses the class labels of the k most similar neighbors to predict the class of the new document. The classes of these neighbors are weighted using the similarity of each neighbor to d_j. Commonly, the Euclidean or Manhattan distance is used as similarity measure (Liao and Vemuri, 2002).

Figure 2.2 illustrate an example of a kNN classification decision, when a number

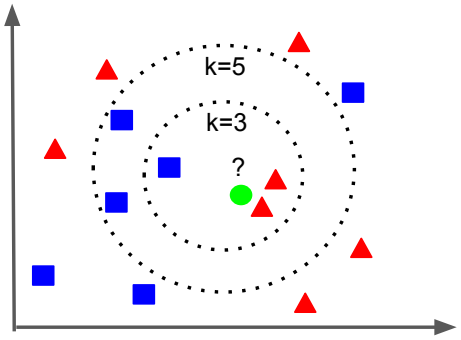


Figure 2.2: kNN classification example.

of $k = 3$ and $k = 5$ neighbors are considered. Given a new document d_j (green point) and the number of neighbours to be ranked $k = 3$, the classifier will assign the red class to d_j, since two of the 3 nearest points are red. If it considers $k = 5$, kNN will assign the blue class.

2.3.2 *Gaussian Naive Bayes*

The Naive Bayes classifier is a simple bayesian network with one root node that represents the class and n leaf nodes that represent the attributes. The probabilistic model of naive Bayes classifiers is based on Bayes' theorem, and the adjective naive comes from the assumption that the features in a dataset are mutually independent. In practice, the independence assumption is often violated, but naive Bayes classifiers still tend to perform very well (Langley et al., 1992). Especially for small sample sizes, naive Bayes classifiers can outperform the more powerful alternatives (Lewis, 1998).

Let x_i be the feature vector of a sample i, $i \in \{1, 2, .., n\}$, c_j be the notation of a class j, $j \in \{1, 2, .., m\}$, and $P(x_i|c_j)$ be the probability of an observing sample x_i belongs to class c_j. The objective function in the naive bayes probability is to maximize the posterior probability over the training data in order to formulate the decision rule

$$NaiveBayes(x_i) = \arg\max P(c_j|x_i), \tag{2.2}$$

in which the posterior probability is defined as

$$P\left(c_j|x_i\right) = \frac{P\left(x_i|c_j\right) \cdot P\left(c_j\right)}{P\left(x_i\right)}. \tag{2.3}$$

One typical way to handle continuous attributes in the Naive Bayes classification is to use Gaussian distributions to represent the likelihoods of the features conditioned on the classes. Thus each attribute is defined by a Gaussian probability density function (PDF) given by:

$$a_i \sim N\left(\mu, \sigma^2\right) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{a_i - \mu}{2\sigma^2}}. \tag{2.4}$$

2.3.3 *Support Vector Machine*

The Support Vector Machine (SVM) projects each document in a vector space, where marginal vectors are used to determine the separation space between classes. The basic idea behind the training procedure is to find a hyperplane, represented by vector $\overrightarrow{w}$, that not only separates the document vectors in one class from those in the other, but for which the separation, or margin, is as large as possible (Pang et al., 2002). This search corresponds to a constrained optimization problem. Let c_j be the correct class of document d_j, the solution can be written as

$$\overrightarrow{w} = \sum_j \alpha_j c_j \overrightarrow{d_j} \quad \alpha_j \geq 0, \tag{2.5}$$

where α_j's are obtained by solving a dual optimization problem. Those $\overrightarrow{d}_j$ such that α_j is greater than zero are called support vectors, since they are the only document vectors contributing to $\overrightarrow{w}$. Classification of test instances consists of determining which side of $\overrightarrow{w}$'s hyperplane they fall on.

2.3.4 *Random Forest*

Random Forest (RNF) is an ensemble learning algorithm, i.e., methods that apply some randomness heuristic to generate many learning models and aggregate their results. In addition to constructing each tree using a different bootstrap sample of the data (bagging heuristic), RNF changes how the classification or regression trees are constructed: each node is split using the best among a subset of predictors randomly chosen at that node.

Let $(X, Y), (X_1, Y_1), ..., (X_n, Y_n)$ be i.i.d. pairs of random variables such that X (feature vector) takes its values in $\mathbb{R}^d$, while Y (the label) is a binary (0,1)-valued random variable; The collection $(X_1, Y_1), ..., (X_n, Y_n)$ is called the training data, and is denoted by D_n. A RNF classifier is defined by (Biau et al., 2008) as

$$RNF(X, Z, D_n) = \begin{cases} 1, \text{ if } \frac{1}{m}\sum_{j=1}^{m} g_n(X, Zj, D_n) \geq \frac{1}{2} \\ 0, \text{otherwise} \end{cases}, \tag{2.6}$$

where n corresponds to the number of base classifiers, m to the number of features to split each node, Z to a randomized set of feature space; and g_n to a base predictor.

2.4 Feature Selection

In a complex classification domain, some features may be irrelevant and others may be redundant (Chandrashekar and Sahin, 2014). These extra features can increase computational time and impact on the system accuracy (Bolon-Canedo et al., 2011). Therefore, selecting important features from input data leads to the simplification of a problem, and faster and more accurate detection rates (Zainal et al., 2006). In what follows, we describe a feature selection model called *Information Gain*.

Information gain is frequently employed as term-goodness criterion in the field of machine learning (Mitchell et al., 1990; Quinlan, 1986). The information gain of a

feature t is defined as

$$IG(t) = -\sum_{i=1}^{|C|} P(c_i)\, logP(c_i) + P(t) \sum_{i=1}^{|C|} P(c_i|t)\, logP(c_i|t) + P(t') \sum_{i=1}^{|C|} P(c_i|t')\, logP(c_i|t'), \tag{2.7}$$

in which c_i represents the i-th category; $P(c_i)$ is the probability of the i-th category, $P(t)$ and $P(t')$ are the probabilities that the feature t appears or not in the documents, respectively; $P(c_i|t)$ is the conditional probability of the i-th category given that feature t appeared; and $P(c_i|t')$ is the conditional probability of the i-th category given that feature t does not appeared. The information gain algorithm produces as output a ranking of features in decreasing order.

2.5 Information Theory Quantifiers

Quantifiers associated with first-order word statistics and other linguistic elements have been employed to quantify the size, coherence, and distribution of vocabularies in language samples of various types (Rosso et al., 2009). Following we describe two information theory quantifiers.

2.5.1 Shannon Entropy

The Shannon entropy can be defined as a measure to quantify the uncertainty of a p distribution Lesne (2014). Let x be a random variable, with values belonging to a finite set χ, the normalized entropy of x is formulated as

$$H(P) = S(P)/S_{max} = \left(-\sum_{p_i \epsilon P} p_i \log(p_i) \right) \Big/ S_{max}, \tag{2.8}$$

where $P = \{p_i; i = 1, ..., N\}$, $S(P)$ denotes the Shannon's entropy and $S_{max} = \log(N)$ corresponds to the maximum entropy.

2.5.2 Jensen-Shannon Divergence

This divergence is defined as a measure of distance between two probability distributions (Mimno et al., 2009). Let P and Q be two probability distributions, and S be the Shannon entropy, then the Jensen-Shannon divergence is calculated as follows:

$$JSD(P, Q) = S\left(\frac{P+Q}{2}\right) - \frac{S(P)+S(Q)}{2}. \tag{2.9}$$

2.6 Model validation techniques

To assess the effectiveness of classification models over unseen data, i.e., their generalization ability, we must use a model validation technique. These techniques help to avoid two types of situations: *overfitting* and *underfitting*. The first occurs when the predictive model fits the training data too well, and the second occurs when the predictive model is not able to capture any trend patterns over the training data. Both situations leads to poor effectiveness when applied to new data. In this section, we will describe two model validation techniques: leave-one-out and k-fold cross-validation.

2.6.1 K-fold cross validation

In k-fold cross-validation, sometimes called rotation estimation, the dataset D is randomly split into k mutually exclusive subsets (the folds) D_1, $D_2 ... D_k$ of approximately equal size (Kohavi et al., 1995). The predictive model is trained and tested k times; each time $t \in 1, 2, ..., k$, it is trained on $D \setminus D_t$ and tested on D_t. In stratified cross-validation, the folds are stratified so that they contain approximately the same proportions of labels as the original dataset.

When the amount of data is large, k-fold cross validation should be employed to estimate the accuracy of the model induced from a classification algorithm, because the accuracy resulting from the training data of the model is generally too optimistic (Witten et al., 2016).

2.6.2 Leave-one-out cross validation

Leave-one-out cross validation (LOOCV) is a special case of k-fold cross validation, in which the number of folds equals the number of instances (Wong, 2015). LOOCV is normally restricted to applications where the amount of training data available is severely limited, such that even a small perturbation of the training data is likely to result in a substantial change in the fitted model (Cawley and Talbot, 2003). Due to be computationally expensive, LOOCV is rarely adopted in large-scale applications.

2.7 Effectiveness measures

There are several ways of evaluating the effectiveness of learning algorithms. Measures of the quality of classification are built from a confusion matrix which records correctly and incorrectly recognized examples for each class (Sokolova et al., 2006).

Table 2.4 presents a confusion matrix for binary classification, where TP are true positives, FP – false positives, FN – false negatives, and TN – true negatives. In this section, we describe three measures commonly use in literature to evaluate the effectiveness of predictive models: precision, recall and f-measure scores.

Table 2.4: Confusion matrix for binary classification.

		Prediction	
		Positive	Negative
Real class	Positive	TP	FN
	Negative	FP	TN

2.7.1 Precision

This metric corresponds to the percentage of items classified as positive that actually are positive (Baeza-Yates and Ribeiro-Neto, 2013). It is given by the ratio between the number of true positives and the sum of true and false positives, i.e.,

$$PR = \frac{|TP|}{|TP| + |FP|}. \tag{2.10}$$

2.7.2 Recall

This metric consists of the percentage of positives that are classified as positive. It is formulated as the ration between the true positives and the sum of true positives and false negatives, i.e.,

$$RE = \frac{|TP|}{|TP| + |FN|}. \tag{2.11}$$

2.7.3 F-measure

F-measure is the harmonic mean of the precision and recall. It relates precision and recall metrics to obtain a quality measure that balances the relative importance of these two metrics (Baeza-Yates and Ribeiro-Neto, 2013), and it is defined as

$$F1 = \frac{2 \times (PR \times RE)}{PR + RE}. \tag{2.12}$$

The F-measure macro (F1-macro) corresponds to the average of all the F1 values of the categories considered.

2.8 Final Remarks

In this chapter, we presented the theoretical background for this work. Our approach to fake news detection, as well as the baselines used for comparison purposes, have

been defined using these concepts.

Chapter 3

Related Work

Fake news has primarily drawn recent attention in a political context but it also has been documented in topics such as vaccination, nutrition, and stock values (Lazer et al., 2018). In this chapter, we discuss techniques that have been proposed to detect fake news published in Web sites and shared in social media. We also discuss publicly-available datasets that have been used to evaluate these techniques.

3.1 Fake News Datasets

To date, there is a lack of large scale publicly-available fake news datasets in literature. Although online news can be massively collected from mainstream sites and social media, the challenges of create a corpus range from finding a properly definition for fake news (Tandoc Jr et al., 2018) to how determining its veracity at low cost and in a timely manner. We listed bellow some of the efforts to build a benchmark for fake news detection:

- **BS Detector** (Risdal, 2016). This dataset contains 12,999 news articles distributed over 244 unreliable web sites. Each news article is labeled by a Chrome extension (rather than human annotators) as belonging to one of the ten follow-

ing categories: *fake news*, *satire*, *extreme bias*, *conspiracy theory*, *rumor mil*, *state news*, *junk science*, *hate group*, *clickbait* and *proceed with caution*. The articles cover news from different domains such as politics, economy and health.

- **BuzzFeed-Webis** (Potthast et al., 2017): This dataset comprises 1,627 news shared on Facebook from nine news agencies over a week close to the 2016 U.S. election. The news were fact-checked claim-by-claim by Buzzfeed journalists, and then, they were rated as *mostly true*, *mixture of true and false*, *mostly false*, and *no factual content*.

- **Celebrity** (Pérez-Rosas et al., 2018). This dataset covers news articles related to public figures of the entertainment industry (actors, singers, socialites). It was collected from online magazines and comprises a total of 500 news articles labeled as *fake* or *real*. The ground truth was obtained using gossip-checking sites.

- **CREDBANK** (Mitra and Gilbert, 2015): This dataset consists of 60 million tweets collected over 5 months, from October 2014 to February 2015. The tweets are grouped into 1049 events, and each event is annotated with a credibility score based on the assessment of 30 Amazon Mechanical Turk annotators. The credibility scores are: *certainly accurate*, *probably accurate*, *uncertain accurate*, *certainly inaccurate*, *probably inaccurate*, and *uncertain inaccurate*.

- **Emergent** (Silverman, 2015). The Emergent dataset focuses on news articles that report rumors about world, business and technology. It contains 1,600 articles collected from web sites during August to November of 2014. The ground truth of each article was given by journalists and follows a truthiness scale: given a rumor with a known label, the annotators assign whether news headlines/contents are *for*, *against* or merely *reporting* the rumor.

- **LIAR** (Wang, 2017): LIAR is a collection of 12,836 political short statements. This dataset was collected from the fact checking site PolitiFact. The statements

were sampled from a diverse set of sources (including TV, newspapers, official statements, campaign speeches), and each statement is labeled for truthfulness according the ratings: *pants on fire*, *false*, *barely true*, *half true*, *mostly true* and *true*.

- **NewsRealiabity** (Rashkin et al., 2017): This corpus includes 74,476 news articles about science, world and politics. The *trusted* articles were sampled from the Gigawords corpus. For the *fake* samples, the ground truth was given by a journalistc report[1] which categorizes web sites as containing satire, hoax and propaganda news.

- **TriFakeNews** (Shu et al., 2017b). TriFakeNews dataset comprises two sets of news articles which both contain metadata about news publishers, text pieces and social engagements. The first set has 182 news labeled by BuzzFeed journalists and the second has 240 news labeled by PolitiFact site. Both datasets have an even distribution of real and fake news articles and cover the political scenario.

- **US-Election2016 Set** (Allcott and Gentzkow, 2017): This dataset contains 948 fake news articles that circulated in the three months before the 2016 U.S. Presidential Elections. The news articles in this set were identified as fake by Snopes, PolitiFact, and BuzzFeed.

- **Fake.br** (Monteiro et al., 2018): Fake.br corpus comprises 7,000 Brazilian news articles collected from January 2016 to January 2018. The news articles are from seven online magazines and cover distinct categories (e.g, politics, entertainment and religion). This corpus also includes metadata information such as author, date of publication and numbers of shares and visualizations.

Table 3.1 shows a summary of the publicly-available datasets described above. We can note that only the BuzzFeed-Webis and TriFakeNews datasets include meta-

[1] www.usnews.com/news/national-news/articles/2016-11-14/avoid-these-fake-news-sites-at-all-costs

Table 3.1: Summary of publicly-available datasets.

Dataset	Source	Ground truth	#samples	Text metadata	Social metadata	Publishers metadata
BS Detector	Web	Chrome Plugin	12,999	✓	✓	
BuzzFeed-Webis	Web, Facebook	Journalists	1,627	✓	✓	✓
Celebrity	Web	Gossip checker	500	✓		
Credbank	Twitter	Crowdsourcing	1049	✓	✓	
Emergent	Web	Journalists	1,600	✓		
Liar	PolitiFact	PolitiFact	12,836	✓		
NewsReliability	Web	Journalists	74,476	✓		
FakeNewsNet	Web, Twitter	BuzzFeed, Politifact	422	✓	✓	✓
US-Election2016	Web, Facebook	Snopes, Politifact	948		✓	
Fake.br	Web	Authors	7,000	✓	✓	✓

data about news text (headline or body text), social interactions (shares, likes) and publishers info (web site bias). Although these two corpus present richness in terms of features, BuzzFeed-Webis does not present a well balanced distribution of samples between classes, and TriFakeNews' samples have their ground truth assessed at a web site level – each news of a given web site will inherit it is label. This labeling assumption does not fit sites that share both fake and real articles.

The others datasets also present limitations such as having few samples (Celebrity, Emergent and US-Election2016) or an unbalanced number of samples per class (Credbank); do not reflect news publishers speakers (Wang, 2017) nor have any news content metadata (Allcott and Gentzkow, 2017). We highlight that although *BS Detector* dataset has a huge number of records, only 8% of them are labeled. The rest of records contains missing labels. Another disadvantage of this corpus is that its ground truth comes from a plugin rather than manual or fact-checking labeling. Thus, any model trained on this dataset will learn indirectly the parameters of Chrome plugin (Shu et al., 2017b).

3.2 Fake News Detection

Researches related to fake news detection fall into two main approaches: content-based and social context-based analysis (Shu et al., 2017a). While the former is design

to captures writing styles on news articles, the second aggregates users behavior by exploring social engagements.

Content-based analysis is core to identifying fake news as the information being reported in news pieces are primarily textual (Kumar and Shah, 2018). Following this approach, Hosseinimotlagh and Papalexakis (2018) addressed the problem of fake news detection using a tensor-based model that clusters news articles into different fake news categories. The proposed model aimed to explore the potential of content by capturing latent relations between articles and terms, as well as spatial relations between terms. They achieved 0.80 of homogeneity (quality of clusters) per fake news category. However, they used a subset of 450 samples from *BS Detector* (Risdal, 2016) dataset. As we discussed previously, this corpus does not present reliable ground truths.

To identify linguistic characteristics of untrustworthy text, Rashkin et al. (2017) studied the feasibility of predicting the reliability of the news article into four categories: trusted, satire, hoax, or propaganda. Their Max-Entropy classifier with L2 regularization on TF-IDF feature vectors resulted on F1 score of 0.65.

Other works have represented news articles by a combination of writing styles attributes (Horne and Adali, 2017; Potthast et al., 2017; Pérez-Rosas et al., 2018). Pérez-Rosas et al. (2018) combined morphological (POS tags), syntactic (context free grammar productions), understadability (readability indexes), psychological (LIWC (Pennebaker et al., 2015)), and n-grams (enconded by TF-IDF) patterns to build a classification model. Their model achieved accuracy values up to 0.76 on *Celebrity* dataset. The authors highlight that legitimate news in tabloid and entertainment magazines seem to use more first person pronouns, talk about time, and use positive emotion words; while fake content has a predominant use of second person pronouns, negative emotion words and focus on the present.

Horne and Adali (2017) and Potthast et al. (2017) studied satire, fake and real news articles. To build a classifier for distinguish between these news categories, they used complexity (median depth of syntax tree, Type-Token Ratio, etc), stylistic (POS

tags) and psychological (LIWC) features. Horne and Adali (2017) findings include that the language used on fake news is more similar to satire than real and it is aimed to create mental associations between entities and claims. They reported accuracy values of 0.91 and 0.78 in the tasks of distinguishing real from satire news and fake from satire, respectively. Potthast et al. (2017) built two classification models to the task of differentiating between satire, fake, mainstream and hyperpartisan news articles. The first model is topic-based (standard *bag of words*) and the second is style-based (n-grams, readability scores, and the average number of words per paragraph). They found that style-based and topic-based classifiers are somewhat effective at differentiating hyperpartisan news from mainstream news (accuracy values up to 0.71 for both models). However, they were not effective at differentiating fake from real news (0.55 accuracy for style-based and 0.52 for topic-based).

Fairbanks et al. (2018) investigated whether credibility and bias can be assessed using content-based (n-grams encoded by TF-IDF values) and structure-based methods. The structure-based method constructs a reputation graph where each node represents a site, and the edges represent mutually linked sites, as well as shared CSS, JavaScript, and image files. They found that both methods achieved high AUC values in detecting bias, but only the structure-based model was able no attain high effectiviness in detecting credibility (AUC value of 0.35 for content-based and 0.88 for structure-based). As the authors emphasize, the AUC for the content model droped due to the unbalanced distributions of samples per class.

Social context-based approaches are also important due to the ability of classification models to extract users response and the spreading patterns of news stories (Castillo et al., 2011; Shu et al., 2017b; Ruchansky et al., 2017). Castillo et al. (2011) analyzed tweets related to"trending" topics and classified them as credible or not credible. They represented the tweets by features based on messages (e.g., length, punctuation), users (e.g., number of followers, registration age), topics (e.g., ratio of positive/negative sentiment, ratio of Urls) and propagations (e.g., depth of re-tweet

tree) characteristics. They reported results of precision and recall in the range of 0.70 and 0.80 in the credibility assignment task.

Following a similar idea, Shu et al. (2017b) proposed the TriFN framework, which captures the tri-relationship between news publishers, articles and users. TriFN combines latent matrix representations with a semi-supervised linear classifier to make predictions over *TriFakeNews* dataset. They found that content-related features are not effective on their own, but when combined with features related to the publishers (partisan bias) and the users' credibility, they can attain accuracy values up to 0.83. We note that in this work, although their dataset is rich in terms of social metadata, they had less than 125 news samples per class (see 3.1) which reflects directly in their findings about news content.

Ruchansky et al. (2017) proposed a framework, called CSI, which consists of a Recurrent Neural Network model to detect fake news based on the response of a given post received from users. The news articles are represented by features as the frequency of temporal spacing of user activity, and user propensity score to engage in a post thread. Their model achieved F1-macro values up 0.84 over a Twitter dataset and 0.93 over a Weibo dataset.

The picture that emerges from these approaches is that content and topic-related features, while effective for detection of bias and satire, often fall short for the task of detecting fake news. Castillo et al. (2011), Rashkin et al. (2017), Hosseinimotlagh and Papalexakis (2018) and Ruchansky et al. (2017) made weak assumptions about the data ground truth – the sample labels are assign based of an inheritance criteria, i.e, samples that were originated from sites or Twitter-topics (with a known reliability) inherit their ground truth. This assumptions does not fit well when sites share both fake and real articles, nor when a tweet related to a misleading topic reports that the topic is fake.

Although works like Horne and Adali (2017) and Shu et al. (2017b) have interesting insights, they have applied learning models in datasets that have less than 80

Table 3.2: Summary of words that addressed the fake new problem.

Work	Learning Approach	Features	Dataset
Castillo et al. (2011)	Decision Tree	Linguistic+Social	Private
Shu et al. (2017b)	Support Vector Machine	Linguistic+Social	TriFakeNews
Rashkin et al. (2017)	Max-Entropy, Naive Bayes Long Short-term Memory	Linguistic	NewsReliability
Horne and Adali (2017)	Support Vector Machine	Linguistic	BuzzFeed-Webis
Potthast et al. (2017)	Random Forest	Linguistic	Emergent
Ruchansky et al. (2017)	Recurrent Neural Network	Linguistic + Social	Twitter
Pérez-Rosas et al. (2018)	Support Vector Machine	Linguistic	Celebrity
Hosseinimotlagh and Papalexakis (2018)	Ensemble Clustering	Linguistic	BS Detector
Fairbanks et al. (2018)	Logistic Regression, Random Forest, Loopy Belief Propagation	Linguistic, Web Network	Private

news per class or have an unbalanced distribution between the classes. This scenarios can lead to overfitting or underfitting in learning models.

In this work, our approach for fake news detection explore distinct linguistic-based features. In contrast with previous works that used n-grams frequencies to represent news articles, our proposed approach encodes n-grams into information quantifiers values, which reduces the dimensionality of the feature space. We perform a detailed experimental evaluation, using multiple baselines (with trustworthy ground truths), that shows our approach outperforms the baseline (see Chapter 5).

3.3 Final Remarks

This chapter presented a brief review of the literature that investigated the problem of fake news related to its detection. We also present some of the datasets on the subject. We highlight that our approach (described in the next chapter) differs from the others reviewed by introduces the use of stylometric features to describe news articles.

Chapter 4

Proposed Approach

In this chapter, we introduce the proposed approach for identifying fake news. Our approach, which we call **LiarDetector**, consists of four main parts: (i) data preprocessing, in which we submit documents to a data cleaning process; (ii) feature extraction, in which we codify documents into numerical vectors; (iii) feature selection, in which we pick the most discriminative set of attributes; and (iv) classification, in which we identify whether a news articles is legitimate or false. Figure 4.1 shows an overview of our approach.

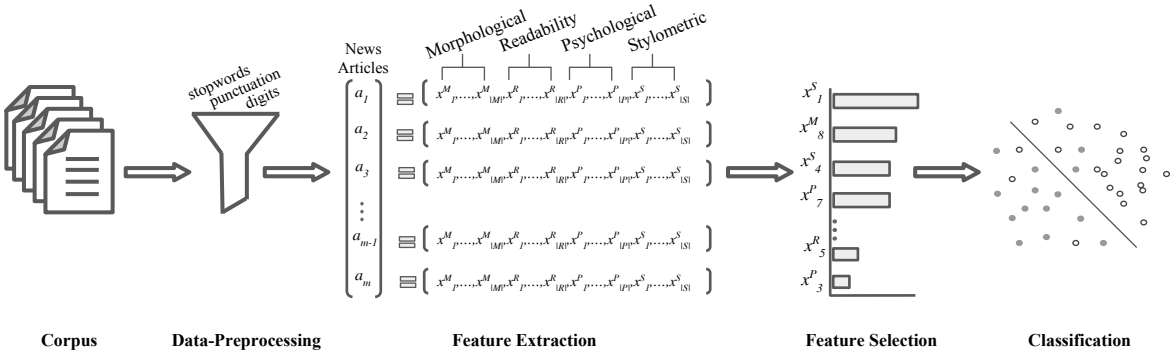


Figure 4.1: Overview of the proposed approach.

4.1 Data Pre-processing

In this first part, we start by removing stopwords, digits and punctuation of the news articles. After this, we submit the documents to a tokenization process and compute the frequency of each token in our corpus.

4.2 Feature Extraction

Recent works have argued that fake news articles are designed to induce affective and inflammatory emotions in readers, and contain text patterns related to understandability that differ from legitimate news (Horne and Adali, 2017; Pérez-Rosas et al., 2018; Bakir and McStay, 2018). This motivated us to investigate linguistic-based features that captures morphological, psychological, readability and stylometric patterns. These set of features are listed in Table 4.1 and we describe them below.

Table 4.1: Linguistic-based features used to represent news articles.

	Description	**Description**	**Description**
Morphological Features	Conjunction, coordinating	Pre-determiner	Interjection
	Numeral, cardinal	Verb, past tense	Verb, base form
	Determiner	Noun, proper, plural	Verb, present participle or gerund
	Foreign word	Noun, common, plural	Verb, past participle
	Preposition or conjunction, subordinating	Genitive marker	Verb, present tense, not 3rd singular
	Adjective or numeral, ordinal	Pronoun, personal	Verb, present tense, 3rd singular
	Adjective, comparative	Pronoun, possessive	WH-determiner
	Adjective, superlative	Adverb	WH-pronoun
	Modal auxiliary	Adverb, comparative	WH-pronoun, possessive
	Noun, common, singular or mass	Adverb superlative	Wh-adverb
	Noun, proper, singular	"To" as preposition/infinitive	Particle
Psychological Features	Summary Dimensions (word tone)	Biological Processes (ingest, health)	Affect (anger, sad, anxiety)
	Function Words (pronoun, negations)	Drives (power, risk)	Relativity (space, time)
	Punctuation Marks (comma, semicolon)	Other Gramar (quantifiers, interrogatives)	Personal Concerns (home, work)
	Perceptual Process (see, hear)	Time Orientation (focuspast, focuspresent)	Social (family, friend)
	Cognitive Processes (insight, certainty)	Informal Language (netspeak, filler)	
Readability Features	Flesch Reading Ease	Words per sentence	Long words
	Flesch Kincaid Grade	Capitalized words	Syllables
	McLaughlin's SMOG	Percentage of stopwords	Lexicon
	Gunning Fog	Urls	Sentences
	Coleman-Liau	Difficult words	Words
	Automated Readability	Characters	
	Linsear Write	Complex words	
Stylometric features	Jensen Shannon divergence	Normalized Shannon entropy	

Morphological Features. This set of features corresponds to the frequency of morphological patterns in texts. We obtain these grammatical patterns (e.g., prepositions,

adjectives, nouns) through part-of-speech tagging, which assigns each word in a document to a category based on both its definition and context.

Psychological Features. Psychological features capture the percentage of total semantic words in texts. We obtain the words' semantics by using a dictionary that has lists of words that express psychological processes (personal concerns, affection, perception). A complete list of these features is available at http://liwc.wpengine.com/compare-dictionaries.

Readability Features. This set of features captures the ease or difficulty of comprehending the sentences in the text. We obtain these features through readability scores (e.g., Gunning-Fog, Coleman-Liau, Flesch Reading Ease) and character, words, and sentences usage.

Stylometric features. This set of features corresponds to values of Normalized Shannon entropy and Jensen-Shannon divergence. The entropy is a measure of uncertainty. When applied over words distributions, entropy reflects the spread of the total words of a text among the different words available (Rosso et al., 2009). The intuition is that the more random or disordered a text, the richer the vocabulary. In this work, we use the normalized Shannon entropy H. H scores that are closer to 0 indicate repetitive words' usage patterns and scores closer to 1 reflect random patterns.

The divergence is a distance measure between two probability distributions. When applied over words distributions, JSD indicates the dissimilarity between distributions. Thus, divergence scores closer to 0 indicates similarity and closer to 1 dissimilarity. In what follows, we describe how we calculated these two information quantifiers.

Based on the token frequencies obtained in the data pre-processing step, we compute two probability distribution functions: p_i and $\langle p_i \rangle$. The former, which we call individual histogram, corresponds to the probability of a token t appears in a news articles of the class $c \, \epsilon \, C$; and it is given by

$$p_i^{(c,t)} = f_i^{(c,t)} \Big/ \sum_{i=1}^{N} f_i^{(c,t)}, \tag{4.1}$$

where N corresponds to the total number of tokens. The latter, which we call reference histogram, is defined as the token average probability over news articles of a class c considered, and it is defined as

$$\left\langle p_i^{(c)} \right\rangle = \left\langle f_i^{(c)} \right\rangle \Big/ \sum_{i=1}^{N} \left\langle f_i^{(c)} \right\rangle, \tag{4.2}$$

with $\left\langle f_i^{(c)} \right\rangle = \sum_{i=1}^{M} f_i^{(c,t)}/M$, where M is the total numbers of documents per class c.

We have a number of reference histograms equals to $|C|$ and, for each news articles, $|C|$ individual histograms. We note that we calculate the reference histograms using only the training set.

We use the histograms calculated previously to codify each news articles by the H and JSD. To obtain the entropy values for each document, we use the individual histograms. We calculate the divergence scores over the reference and individual histograms of each news articles. For both of them, we consider 2 as the logarithm base.

As result of the feature extraction process, we will have each article represented by 33 morphological, 14 readability, 93 psychological and $2 \times |C|$ stylometric features. Regarding the stylometric features, they are calculated per class, that is, if the problem has two classes there will be four attributes representing each document in the corpus.

4.3 Feature Selection

To build our classification approach with the most relevant features, we apply a variation of the information gain based on iterations. In each iteration, we obtain ranked values of features importance in the decreasing order, and then, we pick the most

relevant feature. Thus, the next iteration will have $n - 1$ features to be ranked by information gain. For example, given a threshold of 25% and set of features of size 144, we will selected 36 relevant features.

4.4 Classification

We applied supervised learning algorithms to classify news articles. In supervised learning, a training set (documents with known class labels) is provided and used to build a classification model. The resulting model corresponds to a function that takes as input a feature vector representing an article $a \epsilon \mathbb{R}^d$ and produces an output $\hat{y} \epsilon C$. Once this function is learned, it is used to classify documents with unknown class labels. Here we consider a binary classification problem: given and article a, our learning model has to predict whether a is fake or real.

4.5 Final Remarks

In this chapter, we presented our proposed approach, LiarDetector, for identifying fake news. LiarDetector comprehend four linguistic-based set of features: morphological, psychological, readability and stylometric. The latter component is the core of our approach. By encoding probability distributions of tokens (e.g., n-grams) as entropy and divergence values, we reduce the dimensionality of the feature space, which increases the efficiency of machine learning algorithms. In the next chapter, we will see that we also can increase effectiveness with the use of stylometric features.

Chapter 5

Experimental Evaluation

In this chapter, we give details about the experimental evaluation we conducted and discuss the results. Our main goals are to verify the suitability of our content-based features and assess how effective our classifier is at distinguishing fake from real news. We start by describing the materials and methods we use — datasets, third party libraries, algorithms and metrics —, and then, we report results — using multiple datasets and distinct algorithms — of our approach in comparison with the baseline.

The experiments were grouped into two sections according with the model evaluation technique we applied. In the first section, we report results obtained over the *Celebrity* and *FakeNewsnet* dataset through *leave-one-out* technique. On the other hand, in the second section, we describe results attained over the *Emergent* and *Fake.br* datasets by using *10-fold cross validation*. We have applied distinct model evaluations due to the different sizes of the datasets.

Previous works have found that fake news often displays a divergence between the news headline and the body text (Horne and Adali, 2017; Silverman, 2015): (i) a headline declares a piece of information to be false and the body text declares it to be true (or vice-verse); and (ii) fake news packs the main claim of the article into its title, allowing the reader to skip reading the body article, which tends to be short,

repetitive, and less informative when compared with real news. These divergences between the textual pieces of news articles motivated us to apply the classification models at different granularities: considering only the news headline and only the body text (content).

5.1 Materials and Methods

In this section, we describe the datasets, third party libraries, machine learning algorithms and metrics that we use in our experimental evaluation.

Data. We evaluate our approach over four datasets that have been used in previous works: **Celebrity** (Pérez-Rosas et al., 2018), **Fakenewsnet** (Rashkin et al., 2017), **Emergent** (Silverman, 2015) and **Fake.br** (Monteiro et al., 2018). For the Fake.br dataset, we translate all the samples from Portuguese to the English language. This was need because to obtain the psycho-linguistic features, we only have a dictionary with lexicons in English. Regarding the Emergent dataset, we only consider samples that have the same label for both news headline and content. We also consider that articles labeled as "reporting" false rumors are false as well. Table 5.1 shows the class distribution of news articles per dataset.

Table 5.1: Class distribution and statistics for the datasets. The values corresponds to the number of samples

	Celebrity	**Fakenewsnet**	**Emergent**	**Fake.br**
Fake	240	211	468	3600
Real	240	211	468	3600

Baseline. We consider the classification model Fake news Detector (FNDetector), presented in Pérez-Rosas et al. (2018), as our baseline. This model represents documents by using four sets of linguistic features: n-grams (unigrams + bigrams), psychological, readability and syntactical features. The readability and psychological features are the same we use (see Table 4). The syntactical features are TF-IDF values of production

rules based on context free grammars, i.e., *NN^NP->commission (NN — a noun — is the root, NP — noun phrase — is the parent node, and commission the child node. We choose FNDetector because it achieved high effectiveness on fake news identification task; and it shares some features sets that we also study.

Learning algorithms. We use the following supervised learning algorithms to build the classification models: Support Vector Machine (`SVM`), K-Nearest Neighbor (`KNN`), Random Forest (`RNF`) and Gaussian Naive Bayes (`MNB`). Our goal is to show that our feature sets can lead to accurate results even with distinct learning strategies (support vectors, ensemble, probabilistic and on-demand).

Third party material. We applied the NLTK Bird and Loper (2004) part-of-speech tagger to compute morphological features; Stanford Parser (Klein and Manning, 2003) to extract syntactical features; Textstat[1] library to obtain readability metrics, LIWC (Pennebaker et al., 2015) library to obtain semantic patterns; and Googletrans [2] to translate the samples of Fake.br dataset. We use the implementations of machine learning algorithms of Scikit-learn[3] library with default parameters.

Model evaluation. We perform our evaluations using stratified 10-fold cross-validation and leave-one-out techniques. Due to Fakenewsnet and Celebrity datasets have a relatively small number of samples, we applied leave-one out in our classification process over these datasets. On the second hand, we use stratified ten-fold cross-validation over the Fake.br and Emergent datasets because they present a lager number of samples.

Effectiveness Metrics. To measure the quality of the classification models, we use the measures Precision (PR), Recall (RE), F-measure (F1) and F-measure macro (F1-macro). The collected results are complemented with confidence intervals of $\alpha = 95\%$.

Hardware setup We run our experiments in a MacOS (high sierra version) operation

[1]http://pypi.python.org/pypi/textstat/
[2]https://pypi.org/project/googletrans/
[3]http://scikit-learn.org/

system with a Intel Core i7 processor of 2.5GHz and memory RAM of 16GB.

5.2 Leave-one out: Celebrity and Fakenewsnet

In this section, we report the classification results that we obtain over Celebrity and Fakenewsnet corpus by using leave-one out model evaluation technique. As we discuss in Chapter 2, the LOOCV does not introduce randomness, thus we cannot complement the results with confidence intervals.

Before we compare LiarDetector with the baseline, we first verify which n-gram granularity is better to extract the stylometric component of our approach. Table 5.2 and 5.3 shows the F-measure values attained by unigrams, bigrams and both when the stylometric features are computed over the news headlines and contents of Celebrity and Fakenewsnet dataset, respectively.

Table 5.2: Stylometric features F1 scores with distinct combinations of n-grams representations and algorithms over the Celebrity dataset.

Text	Stylometric Features	Support Vector Machine		K-Nearest Neighbor		Gaussian Naive Bayes		Random Forest	
		Fake	Real	Fake	Real	Fake	Real	Fake	Real
Headline	Unigrams	**0.43**	**0.44**	**0.43**	**0.38**	**0.41**	**0.39**	**0.45**	**0.37**
	Bigrams	0.41	0.43	**0.45**	0.24	**0.47**	0.19	0.42	0.28
	Unigrams+Bigrams	0.38	0.41	0.41	0.36	0.39	0.33	0.45	0.32
Content	Unigrams	**0.65**	0.26	**0.62**	**0.67**	0.62	0.34	**0.58**	0.57
	Bigrams	**0.50**	**0.64**	**0.48**	**0.61**	**0.56**	**0.63**	**0.54**	**0.49**
	Unigrams+Bigrams	0.59	0.52	0.37	0.60	0.58	0.39	0.57	0.5

Table 5.3: Stylometric features F1 scores with distinct combinations of n-grams representations and algorithms over the Fakenewsnet dataset.

Text	Stylometric Features	Support Vector Machine		K-Nearest Neighbor		Gaussian Naive Bayes		Random Forest	
		Fake	Real	Fake	Real	Fake	Real	Fake	Real
Headline	Unigrams	0.64	0.68	**0.63**	0.69	0.59	0.65	**0.64**	0.64
	Bigrams	**0.67**	0.53	0.56	0.68	**0.68**	0.53	**0.67**	0.55
	Unigrams+Bigrams	**0.64**	**0.68**	**0.62**	**0.71**	**0.6**	**0.70**	**0.6**	**0.67**
Content	Unigrams	0.65	0.50	**0.69**	**0.71**	0.65	0.47	**0.68**	**0.71**
	Bigrams	0.60	0.74	0.63	**0.75**	**0.67**	**0.75**	**0.71**	0.61
	Unigrams+Bigrams	**0.68**	**0.76**	0.62	**0.74**	0.66	0.51	0.70	0.63

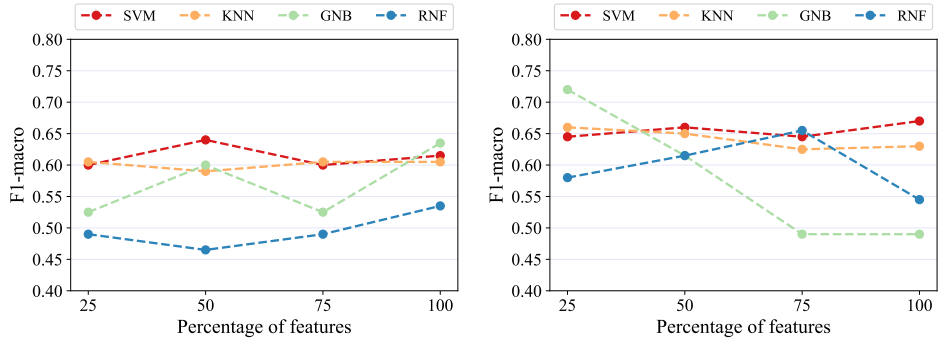
We can note in Table 5.2 that, considering the **headline**, the unigrams representation achieves gains values of at least to 2.32% for the real class when compared with bigrams and unigrams+bigrams. For the fake class, the bigrams representation

attains superior values when combined with KNN and SVM – gains of 4.6% and 14.6%, respectively. Thus, we choose the **unigrams** representation to extract the stylometric features of the news **headline**, since it is presents a better compromise between the two classes. Analogously, we choose the **bigrams** representation to extract the stylometric features of the **news contents**. Nevertheless the unigrams representation achives higher F1 values for the fake class with all the algorithms, it presents poor F1 values with SVM and GNB for the real class.

Reggarding the Fakenewsnet dataset, we selected the **unigrams+bigrams** and **unigrams** representation to extract the stylometric features for the **headlines** and **contents** of news, respectively. We can observe in Table 5.3 that, with respect to the headlines, the unigrams+bigrams representation achieves F1 values gains up to 31% — with GNB — for the real class, and, although it attains F1 value of 11.6% inferior to the bigrams representation — with RFN —, it presents a better compromise between the two classes. For the news content, we can see that the unigrams+bigrams achieves higher F1 values (gains of at least 2.7%) for both classes with SVM; the bigrams attains higher F1 values for the real class with KNN and GNB, and for the fake class with GNB and RNF; and the unigrams attains better F1-macro values for the KNN and RNF (0.70 and 0.69 respectively).

Once we have selected the n-grams granularaties for calculating the stylometric set of features, we are able to combine it with the morphological, psychological and readability sets to build our classification approach. As describe in Chapter 4, to build our classifier with the most relevant features, we apply a variation of the Information Gain algorithm. Figure 5.1 and 5.2 show the results obtained by LiarDetector when we consider 25%, 50%, 75% and 100% of the most relevant features over Celebrity and Fakenewsnet datasets, respectively.

We can see in Figure 5.1a that LiarDetector achieves and overall higher F1-macro values — 0.61, 0.60, 0.63, 0.53 for SVM, KNN, GNB and RNF — when we consider **100%** of the most relevant features extracted from the news headline. With respect to the



(a) Headline results. (b) Content results.

Figure 5.1: LiarDetector results over the Celebrity dataset considering different percentages of relevant features.

news content (Figure 5.1b), LiarDetector is more accurate when it uses **25%** of the most relevant features — 0.64, 0.66, 0.72, 0.52 for `SVM`, `KNN`, `GNB` and `RNF`.

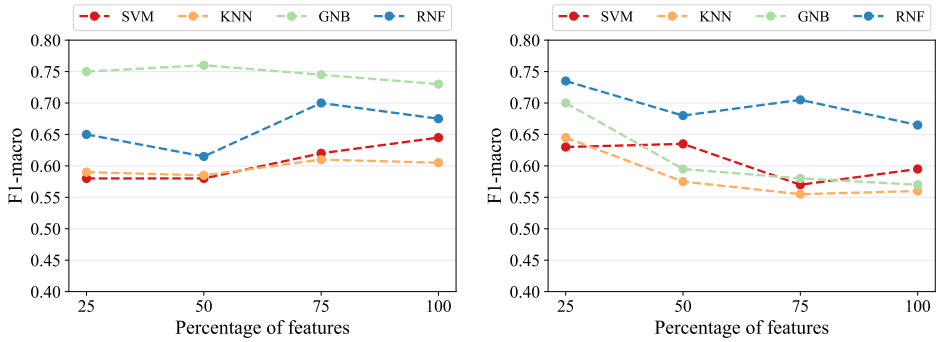
Table 5.4 shows the results attained by the LiarDetector and FNDetector over the Celebrity dataset considering features extracted from the headline and content of news. For the headline, LiarDetector achieves gains of 74.3% and 73.5% for the fake and real classes (with `GNB`) when compared to the baseline. On the other hand, when combined with `RNF` and `SVM`, FNDetector outperforms our approach — 3.2% of gains with the `SVM` and 6% and 25% of gains with the `RNF`. When the `KNN` is used, both classification models have the same effectiveness (in terms of F1 values). Concerning to the news content, our approach overperforms the baseline when it is combined with `KNN` and

Table 5.4: Classification results of models trained over **Celebrity** dataset. PR corresponds to precision, RE to recall and F1 to F-measure.

Text	Approach	Metric	Support Vector Machine		K-Nearest Neighbor		Gaussian Naive Bayes		Random Forest	
			Fake	Real	Fake	Real	Fake	Real	Fake	Real
Headline	LiarDetector	PR	0.62	0.61	0.61	0.6	0.61	0.67	0.54	0.55
		RE	0.61	0.62	0.58	0.63	0.75	0.52	0.67	0.41
		F1	0.62	0.61	**0.60**	**0.61**	**0.68**	**0.59**	0.6	0.47
	FNDetector	PR	0.64	0.64	0.61	0.6	0.38	0.35	0.61	0.63
		RE	0.65	0.62	0.58	0.63	0.4	0.33	0.67	0.55
		F1	**0.64**	**0.63**	**0.60**	**0.61**	0.39	0.34	**0.64**	**0.59**
Content	LiarDetector	PR	0.64	0.65	0.64	0.68	0.68	0.79	0.59	0.58
		RE	0.65	0.64	0.71	0.61	0.84	0.61	0.56	0.61
		F1	**0.65**	**0.64**	**0.68**	**0.64**	**0.75**	**0.69**	0.57	0.59
	FNDetector	PR	0.64	0.65	0.61	0.65	0.52	0.71	0.66	0.74
		RE	0.67	0.62	0.7	0.56	0.94	0.14	0.79	0.59
		F1	**0.65**	**0.64**	0.65	0.61	0.67	0.23	**0.72**	**0.65**

GNB. The F1 gains values are of at least 4.6% for the real and 4.9% for the fake class. Both approaches achieved equal effectiveness using the SVM algorithm. Regarding RNF, the baseline outperforms LiarDetector in both classes. Therefore, the **most accurate** classifiers is **LiarDetector+GNB** for both news headline and content.

We can see in Figure 5.2a that LiarDetector achieves and overall higher F1-macro values — 0.62, 0.61, 0.74, 0.70 for SVM, KNN, GNB and RNF — when we consider **75%** of the most relevant features extracted from the news headline. With respect to the news content (Figure 5.2b), LiarDetector is more accurate when it uses **25%** of the most relevant features — 0.63, 0.64, 0.70, 0.73 for SVM, KNN, GNB and RNF.



(a) Headline results. (b) Content results.

Figure 5.2: LiarDetector results over the Fakenewsnet dataset considering different percentages of relevant features..

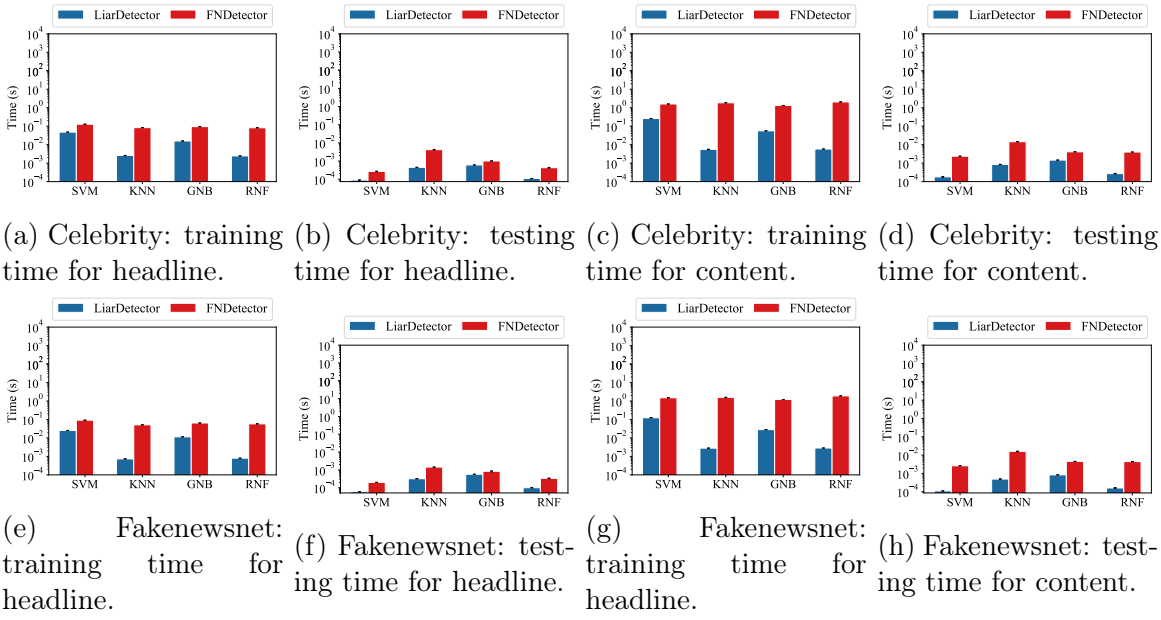
Regarding the Fakenewsnet dataset, the most effective models are **FNDetector+RNF**, **LiarDetector+GNB**, for the news headline, and **FNDetector+RNF** for the content. In Table 5.5, we can notice that our approach attains higher F1 values than the baseline — for the headlines — when it uses the KNN and GNB algorithms, however the baseline outperforms LiarDetector when combined with SVM and RNF. Therefore, the combination FNDetector+RNF and LiarDetector+GNB attain equals values of F1-macro 0.74%. With respect to the news content, our approach outperforms the baseline with all algorithms, except RNF — F1 gains values of at least 4.1% and 5.2% for the fake and real classes. The baseline combined with RNF attain a higher F1 value (0.78%) for the fake class.

Table 5.5: Classification results of models trained over **Newsnet** dataset. PR corresponds to precision, RE to recall and F1 to F-measure.

Text	Approach	Metric	Support Vector Machine		K-Nearest Neighbor		Gaussian Naive Bayes		Random Forest	
			Fake	Real	Fake	Real	Fake	Real	Fake	Real
Headline	LiarDetector	PR	0.62	0.62	0.60	0.63	0.73	0.76	0.67	0.76
		RE	0.63	0.61	0.68	0.54	0.77	0.72	0.82	0.59
		F1	0.62	0.62	**0.64**	**0.58**	**0.75**	**0.74**	0.73	0.67
	FNDetector	PR	0.65	0.64	0.59	0.63	0.7	0.69	0.71	0.79
		RE	0.64	0.65	0.69	0.53	0.69	0.71	0.82	0.67
		F1	**0.64**	**0.65**	0.64	0.57	0.70	0.70	**0.76**	**0.73**
Content	LiarDetector	PR	0.64	0.63	0.66	0.63	0.68	0.77	0.74	0.74
		RE	0.69	0.58	0.67	0.63	0.85	0.57	0.77	0.71
		F1	**0.66**	**0.60**	**0.66**	**0.63**	**0.75**	**0.65**	0.75	0.72
	FNDetector	PR	0.61	0.58	0.58	0.54	0.6	0.76	0.74	0.78
		RE	0.62	0.56	0.57	0.54	0.9	0.34	0.83	0.68
		F1	0.61	0.57	0.58	0.54	0.72	0.47	**0.78**	**0.72**

Figure 5.3 shows the execution time of training and testing FNDetector and LiarDetector classifiers over the Celebrity and Fakenewsnet datasets. We can note that in all scenarios, our approach is faster than the baseline. Considering the Celebrity corpus, LiarDetector achieve execution times up to 33 times faster than the baseline during the training (with `GNB`), and 9 times faster during the testing (with `KNN`) using features derived from the news headlines.

With respect to the content, our approach is up to 354× and 16× faster than the



(a) Celebrity: training time for headline. (b) Celebrity: testing time for headline. (c) Celebrity: training time for content. (d) Celebrity: testing time for content.

(e) Fakenewsnet: training time for headline. (f) Fakenewsnet: testing time for headline. (g) Fakenewsnet: training time for headline. (h) Fakenewsnet: testing time for content.

Figure 5.3: Classification time in seconds (logarithm scale) of our approach vs baseline.

baseline during the training and testing process. Therefore, we can conclude that that the classifier which guarantee the best compromise between effectiveness and efficiency is the **LiarDetector+GNB** no matter the text granularity considered.

Reggarding the Fakenewsnet corpus, our approach achieved execution times up to 72× faster than the baseline during the training (with GNB), and 4 times faster during the testing (with KNN) using features extracted from the news headlines. For the content, our approach is up to 662 and 32 times faster than FNDetector during the training and testing process. We then can conclude that the classifier which guarantee the best compromise between effectiveness and efficiency is also the **LiarDetector+GNB** no matter the text granularity considered.

5.3 10-fold cross validation: Emergent and Fake.br

In this section, we report the classification results that we obtain over Emergent and Fake.br corpus by using stratified 10 fold cross validation model evaluation technique. Since cross validation introduces randomness we have to complement the results with confidence intervals. In this case, we have three possible situations when comparing two means: (i) **if there is no overlap across the means, the higher mean determines the the most accurate classifier**; (ii) **if there is overlap across the means and each confidence interval comprises the other mean, both classifiers are equally effective**; and (iii) **if there is overlap and one mean is not comprised in the other mean confidence interval, we must perform t-test**. Regarding the t-test, we define the null and alternate hypotheses as follows:

- **Null Hypothesis (H0):** There is no significant difference between means values attained with LiarDetector and FNDetector approaches.
- **Alternate Hypothesis (H1):** There is a significant difference between means values attained with LiarDetector and FNDetector approaches.

Analogously to the previous section, before apply the feature selection process, we first verify which n-gram granularity is better to extract the stylometric component of our approach. Table 5.6 and 5.8 shows the F-measure values attained by unigrams, bigrams and both when the stylometric features are computed over the news headlines and contents of Emergent and Fake.br dataset, respectively.

Table 5.6: Stylometric features F1 scores with distinct combinations of n-grams representations and algorithms over the Emergent dataset.

Text	Stylometric Features	Support Vector Machine		K-Nearest Neighbor		Gaussian Naive Bayes		Random Forest	
		Fake	Real	Fake	Real	Fake	Real	Fake	Real
Headline	Unigrams	0.83±0.02	0.84±0.02	0.84±0.03	0.84±0.03	0.79±0.03	0.80±0.03	0.81±0.03	0.81±0.03
	Bigrams	0.78±0.03	0.79±0.03	0.77±0.04	0.80±0.03	0.76±0.04	0.80±0.03	0.77±0.05	0.79±0.03
	Uni+Bigrams	0.84±0.02	0.84±0.02	0.83±0.03	0.83±0.03	0.80±0.03	0.82±0.03	0.81±0.04	0.81±0.04
Content	Unigrams	0.72±0.01	0.41±0.04	0.78±0.02	0.78±0.02	0.65±0.03	0.45±0.04	0.78±0.03	0.75±0.04
	Bigrams	0.85±0.03	0.85±0.03	0.84±0.03	0.84±0.03	0.82±0.04	0.84±0.03	0.82±0.03	0.79±0.04
	Unigrams+bigrams	0.54±0.09	0.60±0.10	0.61±0.09	0.53±0.14	0.63±0.06	0.36±0.18	0.61±0.07	0.50±0.14

Considering the news headlines of the Emergent dataset, the combinations `unigrams+SVM` and `unigrams+bigrams+SVM` are equally accurate for the both classes. `Unigrams+bigrams+SVM` achieves higher values than `bigrams+SVM` for the fake an real classes ($p = 0.018 < 0.05$) — gains of 7% and 6.3%, respectively. The combination `unigrams+SVM` also attains more accurate values than `bigrams+SVM` ($p = 0.07 < 0.05$ and $p = 0.017 < 0.05$ for fake and real classes). Therefore, we can conclude that when combined with the `SVM` classifier the best n-grams to use are unigrams and unigrams+bigrams.

When we examine the `KNN` results for the news headlines in Table 5.6, the unigrams and unigrams+bigrams present the same ability in distinguishing fake from real samples. For the fake class, bigrams are less effective than unigrams+bigrams and unigrams ($p = 0.04 < 0.05$). For the real class, bigrams are equally accurate to unigrams ($p = 0.06 > 0.05$), but they attains lower values than unigrams+bigrams ($p = 0.012 < 0.05$). Hence, we can conclude that when combined with the `KNN` classifier the best n-grams to use are unigrams+bigrams.

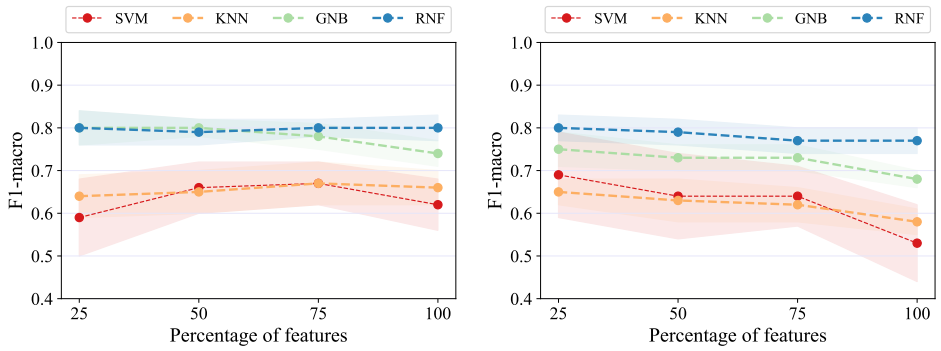
With respect to the `GNB` results for the news headlines, unigrams, bigrams and

unigrams+bigrams are equally accurate. All the combinations have overlaps that comprises the other means, except unigrams+bigrams and bigrams for the fake class. In this case, we can not reject the null hypothesis because $p = 0.09 > 0.05$.

Regarding the `RNF` results for the news headlines, unigrams+bigrams obtains higher F1 values than bigrams with gains of 5% and 2.5% for fake and real classes. Unigrams and unigrams+bigrams are equally accurate for both classes. Unigrams present higher F1 values than the bigrams for the real class, but they present the same effectiveness for fake class ($p = 0.16 > 0.05$). Thus, we can conclude that when combined with the `RNF` classifier the best n-grams to use are unigrams+bigrams.

Considering the overall effectiveness of the algorithms, we choose the **unigrams+bigrams** representation to extract the stylometric features of news headlines. Figure 5.4a shows the results obtained by LiarDetector when we consider 25%, 50%, 75%, 100% of the most relevant features over the Emergent dataset. With exception of `SVM` with 25% and `GNB` with 100%, all the other scenarios present overlaps that results in equally accurate classifiers. Thus, we choose the threshold with less features (25%) to compare with the baseline.

With respect to the news content of the Emergent dataset, we can see in Table 5.6 that the bigrams representation attains the higher F1 values when combined with all the learning algorithms. We then choose **bigrams** to extract the stylometric features



(a) Headline results. (b) Content results.

Figure 5.4: LiarDetector results over the Emergent dataset considering different percentages of relevant features.

of news content. Figure 5.4b shows the results obtained by LiarDetector when we consider 25%, 50%, 75%, 100% of the most relevant features over the Emergent dataset. With exception of the scenario that considers 100% of relevant features, all the others scenarios present overlaps that results in equally accurate classifiers. Thus, we choose the threshold with less features (**25%**) to compare with the baseline.

We can see in Table 5.7 that for the `SVM` and `KNN` algorithms, FNDetector and LiarDetector have the same ability of distinguish between fake and real news samples — considering the headlines. For the `GNB`, our approach and the baseline are equally accurate considering the fake class, but FNDetector outperforms Liardetector in the real class ($p = 0.004 < 0.05$). With respect to the `RNF`, the both models also achieved similar F1 values for the fake and real class ($p = 0.07 > 0.05$).

Table 5.7: Classification results of models trained over **Emergent** dataset. PR corresponds to precision, RE to recall and F1 to F-measure.

Text	Approach	Metric	Support Vector Machine		K-Nearest Neighbor		Gaussian Naive Bayes		Random Forest	
			Fake	Real	Fake	Real	Fake	Real	Fake	Real
Headline	LiarDetector	PR	0.72±0.10	0.75±0.13	0.63±0.05	0.65±0.06	0.78±0.05	0.83±0.04	0.83±0.02	0.79±0.06
		RE	0.62±0.28	0.66±0.21	0.67±0.07	0.61±0.06	0.84±0.05	0.75±0.07	0.76±0.10	0.84±0.03
		F1	**0.56±0.20**	**0.62±0.08**	**0.65±0.05**	**0.63±0.05**	**0.81±0.03**	0.79±0.04	**0.79±0.05**	**0.81±0.03**
	FNDetector	PR	0.72± 0.09	0.74± 0.10	0.66± 0.04	0.68± 0.06	0.83± 0.03	0.82± 0.03	0.73± 0.03	0.86± 0.03
		RE	0.66± 0.19	0.67± 0.22	0.69± 0.07	0.64± 0.05	0.81± 0.04	0.84± 0.03	0.88± 0.03	0.67± 0.06
		F1	**0.64± 0.08**	**0.62± 0.13**	**0.67± 0.05**	**0.66± 0.04**	**0.82± 0.03**	**0.83± 0.02**	**0.80± 0.02**	**0.75± 0.03**
Content	LiarDetector	PR	0.78±0.12	0.81±0.11	0.68±0.04	0.64±0.03	0.78±0.04	0.74±0.04	0.79±0.03	0.84±0.05
		RE	0.75±0.20	0.68±0.22	0.59±0.06	0.71±0.06	0.71±0.06	0.80±0.04	0.84±0.06	0.76±0.05
		F1	**0.70±0.12**	**0.67±0.14**	**0.63±0.04**	0.67±0.03	**0.74±0.04**	**0.76±0.03**	**0.81±0.03**	**0.80±0.03**
	FNDetector	PR	0.72±0.02	0.71±0.01	0.56±0.03	0.55±0.02	0.73±0.04	0.78±0.03	0.65±0.03	0.72±0.03
		RE	0.70±0.04	0.73±0.02	0.52±0.04	0.58±0.03	0.81±0.05	0.69±0.05	0.78±0.04	0.58±0.06
		F1	**0.71±0.02**	**0.72±0.01**	0.54±0.03	0.56±0.02	**0.76±0.03**	**0.73±0.03**	0.71±0.03	0.64±0.03

We can see in Table 5.7 that LiarDetector achieves higher F1 values than the baseline when combined with `KNN` and `RNF` algorithms — considering the contents. The F1 gains for the fake class is up to of 16.6% and, for the real class is 19.6%. Regarding the `SVM` and `GNB` algorithms, both models presents similar effectiveness.

Considering the news headlines of the Fake.br dataset (see results in 5.8), the combinations `unigrams+SVM`, `unigrams+bigrams+SVM` and `bigrams+SVM` present the same ability of classifying fake news. However, for the real classes `unigras+bigrams+SVM` achieves higher F1 values than `unigrams+SVM` and `bigrams+SVM` ($p = 0.018 < 0.05$). And `bigrams+SVM` are more accurate than `unigrams+SVM` since $p = 0.017 < 0.05$. Thus,

Table 5.8: Stylometric features F1 scores with distinct combinations of n-grams representations and algorithms over the Fake.br dataset.

Text	Stylometric Features	Support Vector Machine		K-Nearest Neighbor		Gaussian Naive Bayes		Random Forest	
		Fake	Real	Fake	Real	Fake	Real	Fake	Real
Headline	Unigrams	0.71±0.03	0.68±0.01	0.69±0.03	0.73±0.04	0.70±0.02	0.41±0.07	0.69±0.06	0.72±0.03
	Bigrams	0.73±0.03	0.71±0.02	0.67±0.08	0.70±0.06	0.61±0.09	0.70±0.05	0.68±0.07	0.68±0.06
	Unigrams+Bigrams	0.71±0.06	0.75±0.04	0.70±0.05	0.75±0.03	0.69±0.02	0.41±0.07	0.67±0.03	0.72±0.02
Content	Unigrams	0.90±0.02	0.90±0.02	0.92±0.01	0.92±0.01	0.89±0.02	0.89±0.02	0.92±0.02	0.92±0.02
	Bigrams	0.82±0.03	0.86±0.02	0.71±0.05	0.81±0.02	0.90±0.01	0.89±0.01	0.75±0.01	0.82±0.01
	Unigrams+Bigrams	0.93±0.01	0.93±0.01	0.84±0.03	0.88±0.02	0.89±0.02	0.89±0.02	0.89±0.08	0.90±0.06

we can conclude that when combined with the SVM classifier the best n-grams to use are unigrams+bigrams.

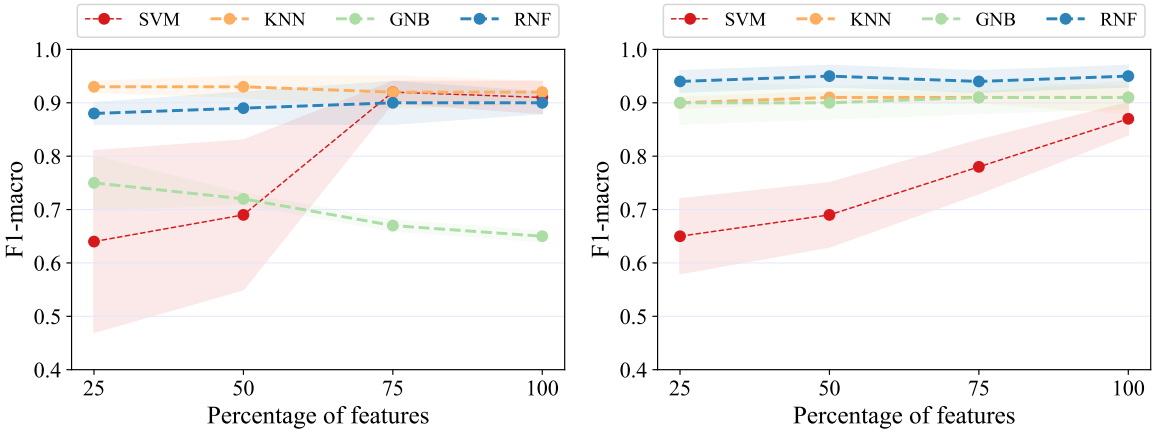
When we examine the KNN algorithm, all the combinations of stylometric features present overlap of means. Hence, we can conclude that the three representations are similarly effective. For the GNB classifier, unigrams+bigrams and unigrams are equally accurate, and moreover, both attain better F1 values than the bigrams for the fake class ($p = 0.015 < 0.05$ and $p = 0.014 < 0.05$ for unigrams+bigrams and unigrams). For the real class, the bigrams representation achieves F1 value gain of 70% when compared with the others. Thus, we can conclude that when combined with the GNB classifier the best n-grams to use are bigrams.

Regarding the RNF algorithms, we can note that there are overlaps in all the n-grams representations for both classes. For the real class, we accept the **H0** hypothesis (there is no significant difference between the means values), since we obtained p values of 0.61 and 0.74 when comparing unigrams+bigrams with bigrams, and unigrams with bigrams, respectively.

In this scenario (Fake.br and news headlines), we have overlaps using KNN and RNF algorithms, and a better effectiveness of unigrams+bigrams with SVM and bigrams with GNB. Comparing this last results, we can conclude that the first is more accurate, since $p = 0.015$ and $p = 0.018$ for fake and real classes. Thus, we choose **unigrams+bigrams** to extract the stylometric features for news headlines.

Figure 5.5 shows the results obtained by LiarDetector when we consider 25%, 50%, 75%, 100% of the most relevant features over the Fake.br dataset. We can see that

both the extreme of x-axis present the better F1 values considering all the algorithms – with respect to the headlines. Whe we consider 25% of the most relevant features, we have F1-macro values of 0.64 ± 0.17, 0.69 ± 0.14, 0.92 ± 0.02, 0.91 ± 0.03. For 100%, we have F1-macro values of 0.88 ± 0.02, 0.89 ± 0.03, 0.90 ± 0.04, 0.90 ± 0.02. Thus, we choose to use **100%** of the most relevant features to compare with the baseline.



(a) Headline results.

(b) Content results.

Figure 5.5: LiarDetector results over the Fake.br dataset considering different percentages of relevant features.

Considering the news contents of the Fake.br dataset, we can note in Table 5.8 that the `unigrams+KNN` representation attains higher F1 values for both classes — gains up to 29.5% and to 13.5% for fake and real classes. On the other hand, all the representation are equally accurate when combined with `GNB`. Regarding the `RNF` algorithm, we achieve higher F1 values with unigrams and unigrams+bigrams for both classes. These both representation are equally effective for fake — $p = 0.29 > 0.05$ and real classes.

For the `SVM` algorithm, unigrams+bigrams attains higher values for the fake ($p = 0.03 < 0.05$) and real ($p = 0.02 < 0.05$) classes when compared with unigrams. It also outperforms the bigrams representation. Hence, we can conclude that both unigrams and unigrams+bigrams are good opttions to extract the stylometric features of the news content. We choose **unigrams** representation.

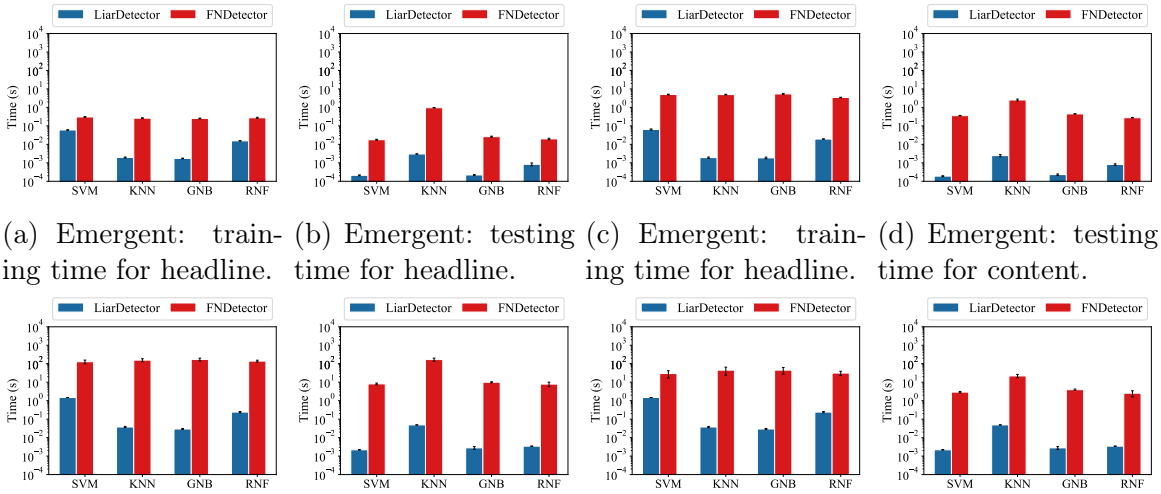
We build our approach considering **100%** of most relevant linguistic features for the news content, since it leads to a higher F1-macro with all algorithms (see Figure 5.5b). We can note in Table 5.9 that with respect to the – **news content** — both FNDetector and LiarDetector are equally accurate in distinguishing the reliability of news articles — their F1-values means overlaps with all the algorithms.

We can see in Table 5.9 that with respect to the — **news headline** — the `SVM` and `RNF` algorithm, LiarDetector is more effective than the baseline with gains up to 9.7% for the fake and at least 14% for the real class. Both our approach and the baseline have a similar ability of distinguishing fake from real news articles For the `GNB` algorithm, FNDetector outperforms our approach with F1 gains values of 13% and 52% for the fake and real classes, respectively. Hence, we can conclude that the most accurate combinations are **`FNDetector+KNN`**, **`LiarDetector+KNN`**, **`LiarDetector+RNF`** and **`LiarDetector+SVM`**.

Table 5.9: Classification results of models trained over **Fake.br** dataset. PR corresponds to precision, RE to recall and F1 to F-measure.

Text	Approach	Metric	Support Vector Machine		K-Nearest Neighbor		Gaussian Naive Bayes		Random Forest	
			Fake	Real	Fake	Real	Fake	Real	Fake	Real
Headline	LiarDetector	PR	0.88±0.10	0.95±0.04	0.93±0.04	0.92±0.01	0.61±0.00	0.88±0.02	0.88±0.05	0.91±0.07
		RE	0.95±0.01	0.87±0.13	0.92±0.01	0.93±0.04	0.95±0.01	0.40±0.01	0.91±0.08	0.88±0.07
		F1	0.91±0.02	0.90±0.04	0.92±0.02	0.92±0.02	0.74±0.01	0.55±0.01	0.90±0.02	0.89±0.02
	FNDetector	PR	0.57±0.13	0.92±0.01	0.93±0.04	0.92±0.01	0.83±0.05	0.84±0.05	0.75±0.07	0.89±0.05
		RE	0.90±0.05	0.22±0.10	0.92±0.01	0.93±0.04	0.85±0.05	0.83±0.05	0.92±0.04	0.69±0.11
		F1	0.72±0.10	0.30±0.35	0.92±0.02	0.92±0.02	0.84±0.05	0.84±0.05	0.82±0.06	0.78±0.08
Content	LiarDetector	PR	0.93±0.01	0.80±0.10	0.90±0.04	0.92±0.05	0.90±0.03	0.92±0.02	0.93±0.04	0.96±0.00
		RE	0.81±0.07	0.90±0.01	0.92±0.05	0.90±0.05	0.92±0.02	0.90±0.03	0.96±0.00	0.93±0.04
		F1	0.86±0.04	0.84±0.06	0.91±0.03	0.91±0.03	0.91±0.02	0.91±0.02	0.95±0.02	0.94±0.02
	FNDetector	PR	0.78 ±0.10	0.93 ±0.02	0.90 ±0.04	0.92 ±0.05	0.90 ±0.03	0.92 ±0.01	0.93 ±0.01	0.96 ±0.02
		RE	0.92 ±0.03	0.61 ±0.10	0.92 ±0.05	0.90 ±0.05	0.92 ±0.01	0.90 ±0.03	0.96 ±0.02	0.92 ±0.01
		F1	0.83 ±0.07	0.79 ±0.09	0.91 ±0.03	0.91 ±0.03	0.91 ±0.02	0.91 ±0.02	0.94 ±0.02	0.94 ±0.02

Figure 5.6 shows the execution time of training and testing FNDetector and LiarDetector classifiers over the Emergent and Fake.br datasets. We can note that in all scenarios, our approach is faster than the baseline. Considering the Emergent corpus, LiarDetector achieve execution times up to 146 times faster than the baseline during the training (with `GNB`), and 320 times faster during the testing (with `KNN`) using features derived from the news headlines.



(a) Emergent: training time for headline. (b) Emergent: testing time for headline. (c) Emergent: training time for headline. (d) Emergent: testing time for content.

(e) Fake.br: training time for headline. (f) Fake.br: testing time for headline. (g) Fake.br: training time for content. (h) Fake.br: testing time for content.

Figure 5.6: Classification time in seconds (logarithm scale) of our approach vs baseline over the Emergent and Fake.br datasets

With respect to the content, our approach is up to 2899× and 1933× faster than the baseline during the training and testing process. Therefore, we can conclude that that the classifier which guarantee the best compromise between effectiveness and efficiency is the**LiarDetector+GNB** when the features were computed using the news headlines, and **LiarDetector+RNF** using the news contents.

For the Fake.br datasets, LiarDetector achieve execution times up to 1536 times faster than the baseline during the training (with `GNB`), and 1410 times faster during the testing (with `KNN`) using features derived from the news headlines. With respect to the content, our approach is up to 5839× and 3568× faster than the baseline during the training and testing process. Therefore, we can conclude that that the classifier which guarantee the best compromise between effectiveness and efficiency is the **LiarDetector+RNF** when the features were computed using both the news headlines and contents.

5.4 Final remarks

In this chapter, we presented the experimental evaluation that we use to assess the efficiency and effectiveness of our proposed approach. In terms of classification quality (F1 values), we attained results equal or higher than the baseline in all scenarios (with exception of Fakenewsnet-content, **FNDetector+RNF** is more accurate). Regarding efficiency (training and testing times), **LiarDetector** is up to 5839 times faster than the baseline when it is build with Gaussian Naive Bayes and unigrams over the news content of Fake.br corpus.

Chapter 6

Conclusions

In this dissertation, we presented a novel classification approach based linguistic features for identifying fake news. Through a detailed experimental evaluation, we showed that LiarDetector accurately distinguishes fake from real news, in different domains (entertainment, business and world context) and using distinct classification strategies (i.e., Support Vector Machine and Random Forest).

In comparison with the baseline, our approach presented achieved gains up 74.3% of F1 score (classification quality). In addition, considering the execution time, LiarDetector was up to 5839 times faster than FNDetector. The efficiency gain is a result of the dimensionality reduction provided by the stylometric set of feature we use, which represents news articles by four features instead of a high dimensional representations (n-grams or syntax trees). We highlight that our set of features can be used not only over the fake news domain, but over general text classification problems. Hence, we can accept our firts hypothesis: "*A classification approach that considers distinct sets of linguistic-based features can lead to accurate prediction values*".

Among our findings, we highlight that (i) **our approach can accurately classifying news reliability by using only the headline of the news articles**; (ii) **stylometrics features are able to achieve higher F1-values, even when they**

are used solely with the learning algorithms. The latter finding prove our second hypothesis "*Applying information quantifiers to encode high dimensional term-matrix can improve both effectiveness and efficacy results of learning models*". It also implies that the quantifiers entropy and divergence not only encrypt the information of sparse term-matrix, but also add discriminative information when representing news articles.

The first finding confirms our third hypothesis: "*Building learning models with features extracted from news articles headlines can lead to competitive effectiveness results, when compared with models build on news body text.* It also suggests that if our approach were applied to social media, we could achieve accurate results by inferring the veracity of social posts – since these platforms provide a limited number of characters. In addition, we note that although the news headline represents short text, it plays an important role in the news misinformation eco-system: many readers tend to share news based solely on their headline, without checking the main content of the news (Blom and Hansen, 2015; Silverman, 2015; Horne and Adali, 2017). In addition, news content often contains only images and videos.

6.1 Limitations

As limitations of this work, we can cite the hardness of finding datasets with trusted ground truths and large number of samples. These factors impact directly on the generalization ability of learning models and the reliability of results. We can also mention that since our approach is based solely on news text patterns, it probably will not perform well in a scenario that fake stories were written using text patterns extremely similar to real news ones. This scenario would require a more sophisticated fake news fabrication process, but it is a potential scenario due to the inherent sensationalist that many news.

6.2 Publications

As results of this work, we presented the following contributions:

- **Detecting Hate, Offensive and Regular Speech in Short Comments** (accepted in the 23rd Brazilian Symposium on Multimedia and the Web, 2017).
- **An Approach to Identify and Monitor Haters in Online Social Networks** (accepted in the Workshop of Thesis Dissertation in the 23rd Brazilian Symposium on Multimedia and the Web, 2017).
- **A Topic Agnostic Approach for Identifying Fake News.** (accepted in the 1st International Workshop on Misinformation, Computational Fact-Checking and Credible Web – within World Wide Web Conference, 2019)
- **An Information Theory Approach for Identifying Fake News** (submitted to the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval)

6.3 Future Work

As future work, we intend to evaluate the effectivenness of others stylometric quantifiers as data representation alternatives, such as the Jaccard, Canberra and Sørensen divergences. We also plan to combine LiarDetector with social engagements features to capture users behavior related to news articles veracity. Another direction that we aim to study consists on the extraction of attributes from web sites (such as advertisements and visual makeup information) to characterize different types of news.

Bibliography

Allcott, H. and Gentzkow, M. (2017). Social media and fake news in the 2016 election. *Journal of Economic Perspectives*, 31(2):211–36.

Baeza-Yates, R. and Ribeiro-Neto, B. (2013). *Recuperação de Informação-: Conceitos e Tecnologia das Máquinas de Busca*. Bookman Editora.

Bakir, V. and McStay, A. (2018). Fake news and the economy of emotions: Problems, causes, solutions. *Digital Journalism*, 6(2):154–175.

Biau, G., Devroye, L., and Lugosi, G. (2008). Consistency of random forests and other averaging classifiers. *Journal of Machine Learning Research*, 9(Sep):2015–2033.

Bird, S. and Loper, E. (2004). Nltk: the natural language toolkit. In *Proceedings of the Association for Computational Linguistics on Interactive poster and demonstration sessions*, page 31.

Blake, A. (2018). A new study suggests fake news might have won donald trump the 2016 election.

Blom, J. N. and Hansen, K. R. (2015). Click bait: Forward-reference as lure in online news headlines. *Journal of Pragmatics*, 76:87–100.

Bolon-Canedo, V., Sanchez-Marono, N., and Alonso-Betanzos, A. (2011). Feature selection and classification in multiple class datasets: An application to kdd cup 99 dataset. *Expert Systems with Applications*, 38(5):5947–5957.

Braun, J. A. and Eklund, J. L. (2019). Fake news, real money: Ad tech platforms, profit-driven hoaxes, and the business of journalism. *Digital Journalism*, pages 1–21.

Castillo, C., Mendoza, M., and Poblete, B. (2011). Information credibility on twitter. In *Proceedings of the 20th international conference on World wide web*, pages 675–684.

Cawley, G. C. and Talbot, N. L. (2003). Efficient leave-one-out cross-validation of kernel fisher discriminant classifiers. *Pattern Recognition*, 36(11):2585–2592.

Chandrashekar, G. and Sahin, F. (2014). A survey on feature selection methods. *Computers & Electrical Engineering*, 40(1):16–28.

Domonoske, C. (2016). Students have dismaying inability to tell fake news from real, study finds.

Edkins, B. (2016). Americans believe they can detect fake news. studies show they can't. url = `www.forbes.com/sites/brettedkins/2016/12/20/americans-believe-they-can-detect-fake-news-studies-show-they-cant/`.

Fairbanks, J., Fitch, N., Knauf, N., and Briscoe, E. (2018). Credibility assessment in the news: Do we need to read? In *Proceedings of the MIS2 Workshop held in conjuction with 11th International Conference on Web Search and Data Mining*, pages 799–800.

Fisicaro, C. and Gauvin, L. (2018). Part of speech tagging.

Gottfried, J. and Shearer, E. (2016). News use across social media platforms 2016. `http://www.journalism.org/2016/05/26/news-use-across-social-media-platforms-2016`.

Horne, B. D. and Adali, S. (2017). This just in: fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news. In *Proceedings of the 2nd International Workshop on News and Public Opinion*.

Hosseinimotlagh, S. and Papalexakis, E. E. (2018). Unsupervised content-based identification of fake news articles with tensor decomposition ensembles. In *Proceedings of the MIS2 Workshop held in conjuction with 11th International Conference on Web Search and Data Mining*, pages 799–800.

Klein, D. and Manning, C. D. (2003). Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 423–430. Association for Computational Linguistics.

Kohavi, R. et al. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Montreal, Canada.

Kumar, S. and Shah, N. (2018). False information on web and social media: A survey. *Social Media Analytics: Advances and Applications*.

Langley, P., Iba, W., Thompson, K., et al. (1992). An analysis of bayesian classifiers. In *Aaai*, volume 90, pages 223–228.

Lazer, D. M., Baum, M. A., Benkler, Y., Berinsky, A. J., Greenhill, K. M., Menczer, F., Metzger, M. J., Nyhan, B., Pennycook, G., Rothschild, D., et al. (2018). The science of fake news. *Science*, 359(6380):1094–1096.

Lesne, A. (2014). Shannon entropy: A rigorous notion at the crossroads between probability, information theory, dynamical systems and statistical physics. *Mathematical Structures in Computer Science*, 24:240–311.

Lewis, D. D. (1998). Naive (bayes) at forty: The independence assumption in information retrieval. In *European conference on machine learning*, pages 4–15. Springer.

Liao, Y. and Vemuri, V. R. (2002). Use of k-nearest neighbor classifier for intrusion detection. *Computers & security*, 21(5):439–448.

Marcus, J. (1992). *Mesoamerican writing systems: Propaganda, myth, and history in four ancient civilizations*. Princeton University Press Princeton.

Mimno, D., Wallach, H. M., Naradowsky, J., Smith, D. A., and McCallum, A. (2009). Polylingual topic models. In *Proc. of the Conf. on Empirical Methods in Natural Language Processing*, pages 880–889.

Mitchell, T., Buchanan, B., DeJong, G., Dietterich, T., Rosenbloom, P., and Waibel, A. (1990). Machine learning. *Annual review of computer science*, 4(1):417–433.

Mitra, T. and Gilbert, E. (2015). Credbank: A large-scale social media corpus with associated credibility annotations. In *Proceedings 9th International AAAI Conference on Web and Social Media*, pages 258–267.

Mohri, M., Rostamizadeh, A., and Talwalkar, A. (2012). *Foundations of machine learning*. MIT press.

Monteiro, R. A., Santos, R. L., Pardo, T. A., de Almeida, T. A., Ruiz, E. E., and Vale, O. A. (2018). Contributions to the study of fake news in portuguese: New corpus and automatic detection results. In *International Conference on Computational Processing of the Portuguese Language*, pages 324–334. Springer.

Narayanan, V., Barash, V., Kelly, J., Kollanyi, B., Neudert, L.-M., and Howard, P. N. (2018). Polarization, partisanship and junk news consumption over social media in the us. *Oxford Internet Institute*.

Pains, C. (2018). Movimento de pais contra vacinação cresce no mundo: No brasil cobertura é estável mas com leve queda. `https://oglobo.globo.com/sociedade/saude/movimento-de-pais-contra-vacinacao-cresce-no-mundo-21620399#ixzz5OYhrwYo3`.

Pang, B., Lee, L., and Vaithyanathan, S. (2002). Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing*, volume 99, pages 79–86. Association for Computational Linguistics.

Pennebaker, J., Boyd, R., Jordan, K., and Blackburn, K. (2015). The development and psychometric properties of liwc2015. 10.15781/T29G6Z.

Pérez-Rosas, V., Kleinberg, B., Lefevre, A., and Mihalcea, R. (2018). Automatic detection of fake news. In *International Conference on Computational Linguistics*, pages 3391–3401.

Potthast, M., Kiesel, J., Reinartz, K., Bevendorff, J., and Stein, B. (2017). A stylometric inquiry into hyperpartisan and fake news. *CoRR*, abs/1702.05638.

Quinlan, J. R. (1986). Induction of decision trees. *Machine learning*, 1(1):81–106.

Rashkin, H., Choi, E., Jang, J. Y., Volkova, S., and Choi, Y. (2017). Truth of varying shades: Analyzing language in fake news and political fact-checking. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 2931–2937.

Risdal, M. (2016). Getting real about fake news. `https://www.kaggle.com/mrisdal/fake-news/home`.

Rosso, O. A., Craig, H., and Moscato, P. (2009). Shakespeare and other english renaissance authors as characterized by information theory complexity quantifiers. *Physica A: Statistical Mechanics and its Applications*, 388:916–926.

Ruchansky, N., Seo, S., and Liu, Y. (2017). Csi: A hybrid deep model for fake news detection. In *Proceedings of the Conference on Information and Knowledge Management*, pages 797–806.

Shalev-Shwartz, S. and Ben-David, S. (2014). *Understanding machine learning: From theory to algorithms*. Cambridge university press.

Shu, K., Sliva, A., Wang, S., Tang, J., and Liu, H. (2017a). Fake news detection on social media: A data mining perspective. *ACM SIGKDD Explorations Newsletter*, 19:22–36.

Shu, K., Wang, S., and Liu, H. (2017b). Exploiting tri-relationship for fake news detection. *CoRR*, abs1712.07709.

Silverman, C. (2015). Lies, damn lies, and viral content: How news websites spread (and debunk) online rumors, unverified claims and misinformation. *Tow Center for Digital Journalism*, 168.

Silverman, C. and Singer-Vine, J. (2016). Most americans who see fake news believe it, new survey says. BuzzFeed News.

Sokolova, M., Japkowicz, N., and Szpakowicz, S. (2006). Beyond accuracy, f-score and roc: a family of discriminant measures for performance evaluation. In *Australasian joint conference on artificial intelligence*, pages 1015–1021. Springer.

Tandoc Jr, E. C., Lim, Z. W., and Ling, R. (2018). Defining "fake news" a typology of scholarly definitions. *Digital Journalism*, 6(2):137–153.

Tardáguila, C., Benevenuto, F., and Ortellado, P. (2018). A new study suggests fake news might have won donald trump the 2016 election.

Wang, W. Y. (2017). "liar, liar pants on fire": A new benchmark dataset for fake news detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 422–426.

Witten, I. H., Frank, E., Hall, M. A., and Pal, C. J. (2016). *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.

Wong, T.-T. (2015). Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation. *Pattern Recognition*, 48(9):2839–2846.

Zainal, A., Maarof, M. A., and Shamsuddin, S. M. (2006). Feature selection using rough set in intrusion detection. In *TENCON 2006-2006 IEEE Region 10 Conference*, pages 1–4. IEEE.