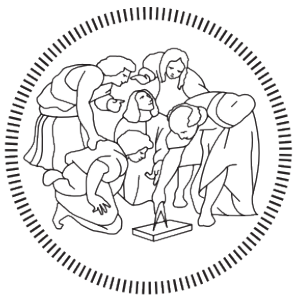
Text-based and graph-based analysis for fake news detection on social mediaa



POLITECNICO
MILANO 1863

Ennio Nasca

Student Id: 894144

Advisor: Prof. Marco Brambilla

Scuola di Ingegneria Industriale e dell'Informazione
Dipartimento di Elettronica, Informazione e Bioingegneria
Politecnico di Milano

This thesis is submitted for the degree of
Master of Science in Computer Science and Engineering

October 2019

I would like to dedicate this thesis to my family,
thank you for always believing in me.

Ringraziamenti

Sono ormai giunto alla fine di questo mio percorso accademico e guardandomi indietro vedo quanta gente mi è stata accanto in questi cinque anni, e non solo. So bene che questo è solo l'inizio di una nuova avventura, ma desidero esprimere la mia più profonda gratitudine verso molte persone.

Ringrazio anzitutto il mio relatore, il professor Marco Brambilla, senza il suo supporto e la sua guida questa tesi non esisterebbe.

Un grazie speciale va a chi mi è stato accanto in questi ultimi due anni, per avermi sempre spronato a dare il massimo ma soprattutto per aver reso questi anni indimenticabili e stupendi.

Inoltre desidero ringraziare gli amici di sempre che hanno contribuito a rendermi la persona che sono oggi. Perché se ho raggiunto questo traguardo è anche merito loro.

Infine un grazie particolare alla mia famiglia per avermi sempre sostenuto in ogni istante. Senza di loro tutto questo non sarebbe stato possibile.

Abstract

In the last decade, the spread of social media has disrupted the typical news flow, shifting the role of news content producer from traditional news sources, e.g. newspapers and television, to social platforms. This lead social media, such as Facebook and Twitter, to become the main source of news online with more than 2.4 billion internet users.

The main drawback of this new multi-directional flow of information is that it lacks one of the fundamentals steps of the typical editorial process, which is fact-checking. The large spread of low quality information poses several threats to our everyday society and democracy. Therefore, the problem of detecting fake news on social media has drawn global attention. One of the key difficulties in detecting fake news lies in the nature of its content, which is intentionally deceptive, thus the exclusive use of text in combination with existing algorithms is ineffective.

For this reason, this thesis focuses on an extended set of features that entail social information, such as user social engagement and news diffusion, with the goal of displaying the different contributions that various features have in the detection task.

In addition, given the nature of social platforms, this work proposes a new representation that models the relationship among users and articles as a graph and frames the task of classifying fake news as a node classification problem using graph neural network.

Sommario

Nell'ultimo decennio la diffusione dei social media ha modificato radicalmente il processo comunicativo, portando i social ad avere il ruolo che una volta avevano le fonti di notizie come ad esempio i giornali e le televisioni. Questo ha fatto si che social media come Facebook e Twitter siano diventati le principali fonti di notizie per molte persone, infatti basti pensare che circa 2.4 miliardi di utenti utilizzano giornalmente tali piattaforme.

Il principale svantaggio di questo nuovo flusso multi-direzionale delle informazioni è dato dal venir meno di uno dei passaggi fondamentali del classico lavoro editoriale, ovvero la verifica dei fatti e delle fonti. La vasta diffusione di notizie di bassa e dubbia qualità fa sorgere alcuni pericoli per quanto riguarda sia la nostra società che la nostra democrazia. Pertanto, il problema relativo all'identificazione delle fake news sui social media ha catturato l'attenzione mondiale. Una delle difficoltà principali nel trovare le fake news è intrinseca nella natura stessa della notizia, in quanto questa è scritta di modo tale da fuorviare il lettore. Ciò fa si che l'utilizzo di tecniche classiche, basate esclusivamente sul contenuto testuale della notizia, non risulti essere efficace in questo caso.

Per questo motivo questo lavoro di tesi considera come possibili caratteristiche discriminanti degli aspetti che coinvolgano anche la parte social, come ad esempio l'attività dell'utente e la diffusione delle notizie. Lo scopo è quello di rendere evidente quali siano i vari contributi che queste diverse caratteristiche apportano al problema dell'identificazione delle fake news.

Inoltre, data la natura delle piattaforme social, questo lavoro propone una nuova formulazione del problema che modella le relazioni tra utenti e articoli per mezzo di un grafo, e riconduce il problema di scovare le fake news a quello di classificare i nodi di un grafo per mezzo delle graph neural networks.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Context and problem statement

In the last decade, the Internet consolidated itself as a very powerful platform that has changed forever the way we do business, and the way we communicate with each other. According to recent statistics [1], the number of internet users is continuously increasing: in 2019 it is estimated to be equal to 4.51 billion of users, 9% more than 2018.

Two things, in our opinion, have pushed the Internet's growth: the social web and mobile technology. These two innovations have revolutionized the way people use and interact online. Up until now, the prevailing sentiments towards the Internet were enthusiasm, for the unprecedented ease of communication made possible by the spread of social media like Facebook, Twitter or LinkedIn; and euphoria, for the unimaginable reach of powerful search engines like Google or Yahoo. For example, since its creation in year 2004, Facebook has grown into a worldwide network of over 2,230 million active users, while Twitter follows with 326 million active users per month. Mobile technology, on the other hand, has made possible much greater reach of the Internet, increasing the number of Internet users everywhere, suffice it to say that 98% percent of Internet users in China are mobile[2], and 80% of the active users on Twitter are from mobile.

Unfortunately, this sense of collective euphoria leds hundreds of millions of people into a "blind trust" towards social media platforms, downplaying or outright ignoring its associated peril. Ironically enough, the very same traits of the Internet that determined its rapid development, i.e. it ease and speed of communication, also undermined its very foundations. Nowadays, the use social media platform is so widespread that it is considered

[1] https://internetworldstats.com/stats.htm
[2] https://hostingfacts.com/internet-facts-stats/

as the prime source of news content in many realities[3]. Because of that, social media have disrupted the news production process, presenting themself as an alternative to traditional media (newspaper, television, etc), a whole new world of opportunities and issues opens up, for example many business models now rely on the number of online users reached by article as the main source of profit.

Problem statement One of the problem that has received increasing attention, not only from academics, is the spread and proliferation of fake news, which are news articles that are intentionally and verifiably false, and could mislead readers. This problem, along with the extremely high rate at which new content is produced, poses new challenges in the fight against misinformation online, and urges the need of automatic systems that can help human fact checkers in discovering and debunking such content. The fundamental questions are two: 1) is it possible to identify fake news relying solely on news content and 2) is it possible to exploit social features to increase performance in a significant way?

1.2 Proposed solution

The goal of this thesis is to create a model that provides a useful tool in the identification process of fake news. Several solutions have been already proposed and a lot of recent literature focuses on this problem. Nonetheless, the majority of these works share a common limitation that lies in the fact that they focus exclusively on features extracted from news content, which is intentionally deceptive, and therefore not convenient.

In this work we try to extend the set of features considered in the detection of fake news, using also the ones retrieved from the social context, such as user social engagement and news diffusion. Moreover, the solution that we propose aims to highlight the difference in importance that have various set of features. The approach that we propose focuses on Twitter users, but it is applicable to other social platforms as well. It starts by collecting together the news related tweets, along with the user social profile and network of the people that posted them. These news articles are then processed with classical text mining operations while user profile are and further enriched using demographics and geolocation. Then, the classification are performed either at user or tweet level, testing out different machine learning models and comparing their performances. In this way, by means of permutation importance, it is possible to analyze the latent information that different features carry.

[3] www.pewresearch.org/fact-tank/2018/12/10/social-media-outpaces-print-newspapers-in-the-u-s-as-a-news-source/

Moreover, considering the intrinsic nature of a social platform, we decided to model the relationship among users and articles as a graph and frame the task of classifying fake news as a node classification problem using graph neural network.

1.3 Structure of the thesis

The structure of the thesis is as follows:

- **Chapter 2** defines and explains the background knowledge and concepts that are related to the work that has been performed for this thesis.
- **Chapter 3** presents past works, in the problem they try to answer and the solution they propose, that are related to this thesis.
- **Chapter 4** contains the problem and the research questions of this thesis, along with high level description of the employed methods that are used in this thesis.
- **Chapter 5** describes the source codes and actual implementations of the used methods.
- **Chapter 6** presents the dataset that has been collected, the results of the experiments and discusses these outcomes.
- **Chapter 7** concludes this report by summarizing the work, doing a critical discussion and presenting possible future directions and open issues.

Chapter 2

Background

In this chapter we present the concepts, models and techniques that are used throughout this work. First, we give some brief definitions, regarding Data Mining, Topic modeling, Machine Learning and Graph Theory that are at the core of the techniques and models that will be used in this thesis. Then we move on to describing the frameworks and libraries that are extensively used throughout this entire work. Finally we introduce the external services and APIs that are used in the data collection and in the features enrichment process.

2.1 Information Extraction

Information extraction is defined as the task of automatically extracting structured information from either unstructured and/or semi-structured machine-readable documents. Nowadays, due to the large availability of unstructured data the need of models that are able to process and perform reasoning on those is increasing. The actual representation of the data may vary enormously, e.g. text, images and videos, and it is strictly dependent from both the task at hand and the field of research. While the goal of information extraction is limited to the organization of information starting from unstructured data, the goal of the subsequent step, i.e. knowledge extraction, is to exploit the latent information that lies within the newly organized data representation in order to perform reasoning on it.

2.2 Data and Text Mining

The objective of text mining is to automatically extract new and unknown information from different unstructured textual documents. This information is then processed and turned into high-quality actionable knowledge. In the context of social networks, the rate at which new

content is produced is extremely high: in case of Twitter, at the time of writing, almost 6000 new tweets are sent every second. This scenario makes the analysis of the data coming from the social context extremely attractive especially for both the freshness of the content and its intrinsic heterogeneity, for example we can have pictures or videos with textual description and social metadata, such as user mentions and likes/shares. From a high-level prospective the general pipeline of a data mining task can be divide into a series of ordered step, below we report a brief description of those:

Selection Depending on the scientific goal of the research, a collection of raw data from relevant sources needs to be carried out, this step consists in the selection of the sources and the actual collection and scraping of useful data from the previously identified sources.

Cleaning Fundamental step that deals with the imputation or removal of errors, inconsistencies and noise that are typically found in raw data. In the specific case of text mining this step also takes into account the removal of *stopwords*, i.e. words that have no contribution in the significance of a sentence, e.g. punctuation and conjunction, *lemmatization*, i.e. the process of reducing words to their inflected form, and *pos tagging*, i.e. assign to each word the corresponding part-of-speech, e.g. noun, verb and adjective.

Transformation The pre-processed data is further transformed into a suitable numerical representation that can be given as input to the models used in the subsequent steps.

Mining Build a model that given as input the vectorized data is able to accomplish the desired task. An high-level categorization of the tasks divides them into two groups

Predictive: The extracted knowledge is useful to get insight on what may happen in the future

Descriptive: The extracted knowledge is useful to explain the behavior that is described in the data and hence what already happened

Validation Evaluate the obtained result and test their soundness using appropriate metrics and statistical test to compare different approaches

2.3 Word and Document embeddings

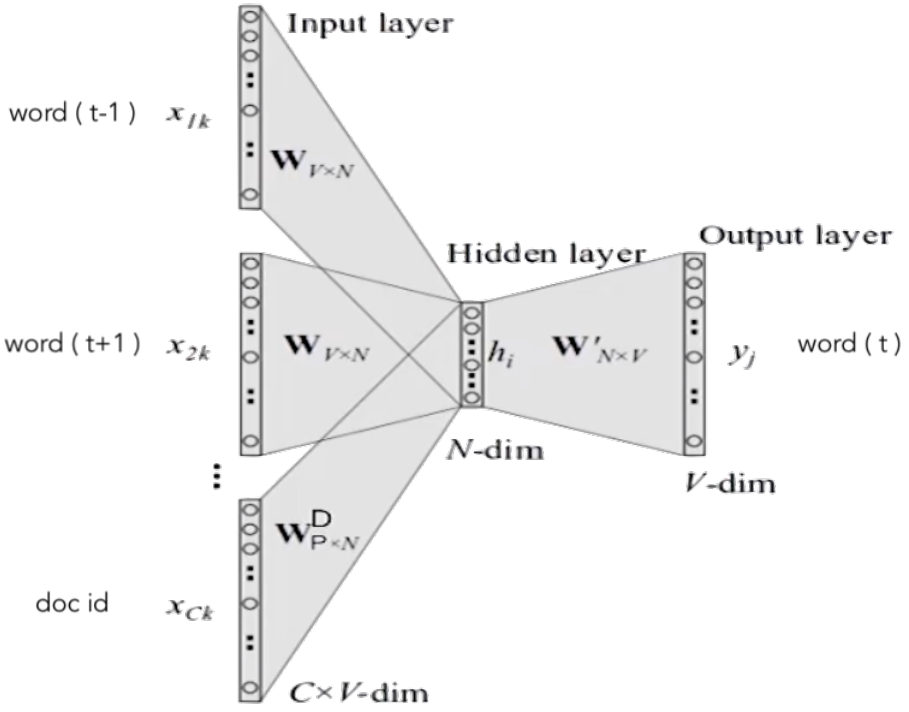
The easiest way to describe words through vectors of numbers is given by Bag of Words. This representation considers words as one-hot vectors, whose dimension depends on the size of

the vocabulary V, i.e. list of all words that appear in the overall collection of documents, a.k.a corpus. The problem with this representation is that words expressing similar concepts do not share similar representations and hence the context in which a word appear is discarded.

Word embedding approaches represent word by means of its neighbors. Words are mapped to vectors into a continuous, fixed size, representation space obtained by unsupervised learning. These representations encode both syntactic similarity and semantic similarity. Another advantage of this family of approaches lies in the fact that the dimensionality D of such vectors is much smaller than the size of the vocabulary.

The process for obtaining word embeddings is usually called Word2Vec[24]. In general, to train these embeddings we start from the one-hot encodings of target words and context words, and, at the end of process, output an embedding matrix $W \in \mathcal{R}^{V \times D}$. After training, the encoding x for a word is obtained as $x = oW$, being o the one-hot encoding of a word. The extension of this model to documents is called Doc2Vec[25]. It exploits the above observations by adding additional input nodes representing documents as additional context. Each additional node can be thought of just as an id for each input document as shown in figure 2.1.

Figure 2.1 Example of Doc2Vec model



2.4 Machine Learning

The most widely used definition of machine learning is that of Carnegie Mellon University Professor Tom Mitchell:

"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E"

The experience we are interested in can be represented as a dataset $D = \{x_1, \dots, x_N\}$ where $x_i \in \mathcal{R}^D$ is the i-th sample or data point defined in a D dimensional space. The task T can have different formulations:

Supervised Learning : Given as input a dataset with labeled samples $D = \{(x_i, y_i)\}_{i=1}^N$, being y_i is the ground truth of x_i, we want to learn a mapping that is able to predict the correct target label for new unseen data points

Unsupervised Learning : Given as input an unlabeled dataset $D = \{x_i\}_{i=1}^N$ the goal is to exploit regularities in D to build useful representation of the data

Throughout this work we focus mainly on supervised problems and in particular we try to solve the problem of classification, i.e. learn a mapping function that is able to assign to a new unseen data point x_i a class y_i from a finite set of possible values.

2.4.1 Document Classification

The task of document classification focuses on automatically labeling text documents on the basis of topics, style or purpose. This task exploits information extraction techniques and in case of similarity based model tries to compute the label of a document from the most similar ones. The vectorized representation of a document can be obtained using methods presented in previous sections. When working with a reduced dimension we can define a document in terms of its keywords distribution.

2.4.2 Models

In this subsection we give a general description of the main machine learning models that are used in the following chapters

Naive Bayes Family of probabilistic classifier that uses Bayes theorem to classify data points $\hat{y} = arg\,max_y \frac{P(x|y)\ P(y)}{P(x)}$ [40]

Logistic Regression Classifier that predicts the target label by assign a score $z_i = \sum_{j=0}^{N} w_j h_j(x_i)$ to each data point using a regression approach and computing the sigmoid value of this score $P(y = 1|x_i) = \frac{1}{1+e^{-z_i}}$

Decision Tree Family of models[33] that define the problem using an acyclic graph, that can be used to make decision. In each node a test on a selected attribute j is performed, branches represent the outcome of those tests. Leaf nodes are nodes that have no child and return the class prediction

Random Forest Ensemble learning method[22] that combines a set of unpruned decision trees classifiers. The class label is assigned by aggregation single predictions using majority voting. The random forest trees are trained using randomized features selections on bootstrapped datasets

Stochastic Gradient Descent Linear classifier that uses SGD[1] for training the model. For each sample the gradient is estimated and the weights of the model are updated with a decreasing strength a.k.a. learning rate

CatBoost[2] High performance and scalable library, developed by Yandex researchers, that performs gradient boosting on decision trees. The algorithm is particularly suitable when working with heterogeneous data coming from different source. Moreover, it handles directly categorical features without the need of pre-processing.

2.5 Topic Modeling

Topic Modeling is the task of discovering latent semantic structures in textual documents. To extract topics analysis statistical methods are employed as models. From a topic modeling point of view a document is modeled as a weighted combination of topics, while a topic is seen as a mixture of words. The family of topic models can be divided into two groups: parametric and non-parametric. The former receives as input an hyperparameter to define the number of topics that are extracted from the documents. The latter can estimate the probable number of topics directly from the provided collection of documents.

[1] https://en.wikipedia.org/wiki/Stochastic_gradient_descent

2.5.1 Latent Dirichlet Allocation

Latent Dirichlet Allocation model (LDA) is a three-level hierarchical, generative probabilistic and parametric model[5]. Each document of the corpus is represented as a finite mixture over an underlying set of topics, where each topic is characterized by a distribution over words.

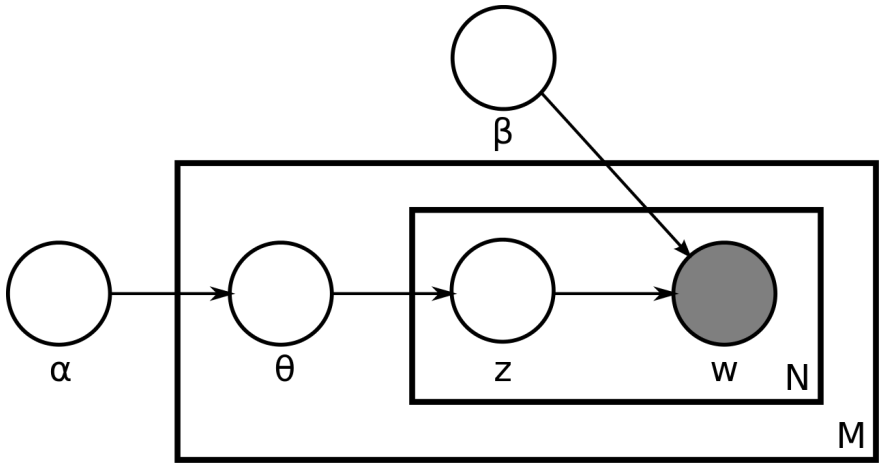


Figure 2.2 Plate representation of the LDA model

The graphical representation of LDA can be observed in Figure 2.2. The outer plate represents the M documents, while the inner one represents the repeated choice of topics and words within a document, with N as the number of words in a document.

- α is the parameter of the Dirichlet prior on the per-document topic distributions,
- β is the parameter of the Dirichlet prior on the per-topic word distribution,
- θ_m is the topic distribution for document m,
- φ_k is the word distribution for topic k,
- z_{mn} is the topic for the n-th word in document m
- w_{mn} is the specific word. It is grayed out, because it is the only observable variable in the system while the others are latent.

LDA is a parametric model, which means that the number of topics has to be provided as a parameter. For this reason, one of the most challenging tasks is to find the best number of topics when creating the LDA model.

2.6 Graph representation learning

In real life, problems may present complex underlying structures that need to be captured somehow. To deal with this issue a possible solution is to use an universal language that allows for an efficient representation. This representation can be obtained by means of graphs.

Graph representation learning, also known as network representation learning, is an area of research that has its foundations in network science, which uses networks for describing systems and extracting knowledge from them. Graph representation learning builds upon some concepts that originate from both graph theory and machine learning.

Classical tasks when working with networks are *node classification*, *link prediction*, and *community detection*. Representation learning on graph aims to find efficient task-independent features that can be directly extracted from the graph structure. A source of inspiration in this direction has been provided by the deep learning field.

In order to better understand the work described in the next chapters, the reader is presented with some mathematical notation and essential terminology that are considered necessary.

Definition 2.6.1. A graph $G = (V, E)$ is a data structure defined over sets of vertices $V = \{v_i : 1 \leq i \leq |V|\}$ and edges $E = \{(i, j) : v_i, v_j \in V\}$.

Definition 2.6.2. The neighbourhood of node v is the set $N(v)$ which contains all the nodes connected to node v by an edge.

The topology of a graph can be described in matrix notation using the adjacency matrix, that describes how nodes are connected one another, or the incidence matrix, that describes how nodes are connected in relationship to edges. Other important matrices in graph theory are the Degree matrix D and the Graph Laplacian, or Kirchhoff matrix, L.

Definition 2.6.3. The adjacency matrix $A \in \mathbb{R}^{|V| \times |V|}$ is the matrix whose general element is $a_{ij} = w_{ij}\ if\ \exists\ (i, j) \in E :\ 0 \leq i, j \leq |V|$ otherwise its value is 0.

Definition 2.6.4. The Degree matrix is a diagonal matrix whose general element $D_{ii} = \sum_j A_{ij}$ is equal to the number of neighbors of node i

Definition 2.6.5. The Feature matrix $X \in \mathbb{R}^{|V| \times d}$ is a matrix where each row X_i represents the vector of features associated with node i

2.7 Main technologies

In this section we list and briefly describe the main technologies use within this thesis. The list includes developing frameworks, code libraries and APIs

2.7.1 Developing Framework

Python

Python 3.6.8 has been chosen as programming language for this thesis. The main reasons of this choice is to be attributed to the popularity and flexibility of the programming language. Moreover, Python community offers a large collection of well-written libraries that have become a go-to reference in the Data Science field.

2.7.2 MongoDB

MongoDB is a free, open-source and cross-platform document database solution, i.e. data are stored in JSON-like documents. The main advantage over traditional relational databases is the flexibility that is introduced by the NoSQL schema that is particularly suitable when dealing with noisy and incomplete data.

2.7.3 Code Libraries

Scikit-learn

Scikit-learn is a free, open-source machine learning library for the Python programming language. The library offers a variety of classification, regression and clustering algorithms including the ones presented in 2.4.2

Pandas

Python Data Analysis Library widely used in the Data Science community. The library allows to manipulate documents, e.g. csv files, and transform those into Python objects, i.e. Dataframe, with a tabular representation that can be easily queried. Moreover, it provides many functionalities useful for cleaning, ordering and merging dataframe

Gensim

Gensim is a vector space and topic modeling library for python that offers a large selection of models, including LDA and Doc2Vec. The main advantage of Gensim lies in the memory

management, which ,thanks to a streaming approach, allows to process even very large corpora.

Newspaper3k

Newspaper3k is a Python library that simplifies article scraping and curation, similar to the requests library for HTTP requests. The library comes with an NLTK module that uses Natural Language Processing (NLP) to extract keywords from an article, along with the authors names and publication date. newspaper3k does linguistic analysis based on word frequency in the corpus.

Pytorch

PyTorch[3] is a machine learning library, developed by Facebook's artificial-intelligence research group, for the programming language Python, based on the Torch library it is widely used for applications such as deep learning and natural language processing.

Pytorch Geometric

PyTorch Geometric[4] (PyG) is an open-source geometric deep learning extension library for PyTorch. It consists of various methods for deep learning on graphs, also known as geometric deep learning, and other irregular structures. The library provides fast access to the implementation of multiple variants of Graph Convolutional Neural networks

2.7.4 APIs

2.7.5 Twitter

Twitter API is the main source of the social content within this work. The API offers free endpoints, with limited usage, that allow developers to access information concerning both user profile and tweets, along with important metadata concerning the retweet information of a status.

Botometer

Botometer is a joint project of the Network Science Institute (IUNI) and the Center for Complex Networks and Systems Research (CNetS) at Indiana University. Botometer checks

[3]https://pytorch.org
[4]https://pytorch-geometric.readthedocs.io

the activity of a Twitter account and gives it a score based on how likely the account is to be a bot. Higher scores are more bot-like.

Face++

Face++ Cognitive Services is a platform offering computer vision technologies that used on the profile images of a Twitter user allows to extract demographic information such as sex, age and ethnicity

Yandex

Yandex is a popular Russian tech company that offers a Geolocation API that allows to parse data extracted from user profile description into a structured form that can be used to obtain information about users country and city of residency.

Chapter 3

Related Work

In this chapter we discuss some of the most relevant researches that relate to the goals of this thesis. First, we present some works that study the news diffusion process and the new behaviors in news consumption on social media, i.e. websites and applications that are designed to allow people to share content quickly, efficiently, and in real-time. In particular, those works have provided some insights and tools that are later on used as building blocks in the data collection pipeline.

Then we shift our focus to researches that try to tackle the problem of news classification and user profiling using content and social features. The key contributions of these works are in the manipulation of the user generated content and social interactions, in order to extract and understand the user profile.

To conclude, after a brief presentation on some of the datasets, and their characteristics, used for the task of fake news detection and classification, we define the position of this work in the literature.

3.1 News diffusion and consumption

Historically, the receivers of news and information, produced by professional media organizations, have been viewed as an indistinguishable mass that passively consumed the provided information [23]. The audience, in that particular case played a marginal, or no, role in the construction of the news and, in general, had zero communication with the media source [15].

The mono-directional flow, between media organizations and news consumers, changed with the advent of internet technologies, and in particular with Web 2.0[1], which facilitated the integration of the audience in the process of news creation.

A radical change occurred with the diffusion of social platforms, which, following the definition of Kaplan [19] is described as: "a group of Internet-based applications that build on the ideological and technological foundations of Web 2.0, and that allow the creation and exchange of user-generated content". Because of this, the information flow had to be re-organized into a new hierarchy, that suited the multi-directionality that is intrinsic in a social network.

In recent years, social media and news diffusion have attracted the attention of many researchers in the community, and this confirmed by the large volume of new scientific papers that are produced on this topic [2][34][3]. In general, those works focus on different aspects, such as the relation between news source and users [32], the polarization of the network of a user [21] and the formation of echo chambers and also news dissemination [11][13].

In the last decade, we have witnessed a shift in the consumption of news from traditional news sources, e.g. newspapers and television, to social media platforms. Nowadays, social medias, such as Facebook, Twitter, YouTube, Snapchat have become the main source of news online with more than 2.4 billion internet users. The reasons for this change can be attributed to the fact that social platforms allow to produce news in a way that is faster and cheaper, while providing users with the ability to interact with the information thorough share and comment.

The main drawback of news consumption on social media lies in the fact that it lacks one of the fundamentals steps of the typical editorial process, which is fact-checking. When we speak about disinformation, we refer to the disclosure of deceptive, unfounded or grossly distorted information, that mimic the content of true news. In fact, one of the many definitions of fake news is: news articles that are intentionally and verifiably false, and could mislead readers [1]. The key elements in this definition are the verifiability of the news and the intent of the writer, which could be for financial or political purpose.

A pioneering work, conducted by Vosoughi *et al.* [37], in the study of news diffusion, found, from the analysis of true and false cascades originated from more than 4.5 million

[1]https://it.wikipedia.org/wiki/Web_2.0

tweets, that falsehood diffused significantly farther, faster, deeper, and more broadly than the truth in all categories of information.

In particular, this behavior is more prominent in case of political news rather than terrorism, natural disasters or science. Moreover, they found that false news piece generates reactions such as surprise or disgust, while true ones emotions such as sadness, anticipation and trust.

The research analyzed also the users who spread false news and found that they have significant less followers and friends and that bots contribute in equal parts to the diffusion of true and false news, pointing out that the difference in diffusion is to be attributed to humans.

The principal causes for which humans are vulnerable to fake news are *Naive Realism* [38], i.e. news consumers deem their perception of reality as the only accurate one and people that present different views are regarded as uninformed or irrational, and *Confirmation Bias* [27], consumers tend to favor news that support and confirm their previous beliefs. Because of confirmation bias, consumers tend to group into echo chambers [12], in which individuals are largely exposed to conforming opinions.

Additionally, social platforms and content providers are increasingly personalizing the user experience in order to carefully filter the content and only present the users with information or elements that may be considered useful or interesting. The potential problem with this tendency is the creation of *filter bubbles* [28], in which recommendation algorithms inadvertently amplify ideological segregation.

Multiple studies in this direction analyzed the interactions between users and news in the context of political events, namely the 2016 USA presidential election [1][29][14][6] and the brexit campaign in the UK [4]. All this works found a strong polarization in the communities, where users that shared similar political beliefs were more likely to interact with each other.

Hermedia et al.[17], while studying the role of social media in the flow of news and information, found that a large number of users values their personal social media network as a way to filter news, rather than solely relying on the judgment of professional news sources or journalists.

3.2 News classification and user profiling

The wide spread of fake news on social networks can have serious consequences in the overall credibility and balance of the entire ecosystem, as it has been shown during the U.S.

2016 presidential election and campaign where fake news were fabricated to mislead[2] or confuse[3] voters.

The aforementioned outcomes highlight the need of new automated techniques that can help in the process of distinguishing between true and false news, since it is impossible for human fact-checkers to manually verify the veracity of every single news piece.

In general, when we refer to fake news we are considering a family of news that can have different characteristics. For example, *click-bait* are stories that are deliberately fabricated to gain more website visitors and increase advertising revenue, this is typically achieved using sensationalist headlines to grab attention; *propaganda* are news created to deliberately mislead audiences, promote a biased point of view or particular political cause or agenda; *satire* are fake news stories written for entertainment and parody purposes.

The majority of fake news are written with a style that emulates the one of professional media sources, this makes the identification difficult and nontrivial when using solely news content based features; therefore, there is a need of auxiliary information, such as user social engagement or user social network, that can help for the task at hand. Unfortunately, one of the major issues when dealing with social features related to fake news lies in the unstructured, noisy and incomplete nature of the data [35].

The task of fake news detection can be formulated, using the notation of Shu et al. [31], as the creation of a prediction function $\mathcal{F} : \mathcal{E} \rightarrow \{0, 1\}$, where the prediction on a news article a uses information related to the publisher p_a and the content of the news c_a along with the social engagement represented by users and statuses.

Typically, frequent set of features, used to infer user's credibility and reliability, are extracted from the social context and are related to social demographics, such as account creation date, number of post authored and number of friends/followers, etc. [9]. Other important features may be extracted from the user generated content, for example linguistic-based features using embedding approaches [30] or user's opinion and sentiment towards the news [18].

Moreover, because users have shown a tendency to form groups with people that share similar beliefs, network based features provided extremely useful features when classifying news pieces. This is the case of Monti et al. [26], who managed to classify fake news using geometric deep learning [7] and graph attention [36] using an approach that relies solely

[2]https://time.com/4783932/inside-russia-social-media-war-america/

[3]https://www.nytimes.com/2016/11/28/opinion/fake-news-and-the-internet-shell-game.html

on network based features, which are defined in terms of diffusion network, i.e. a network that tracks the trajectory of the spread of news[10], where nodes represent users and edges represent the information flow across them.

Depending on the type of fake news we want to detect, different approaches and models can be employed. For example, in case of rumors, i.e. news piece whose veracity is still unverified at the time of spreading, we can define the task as a classification problem where the ground truth is the type of news, e.g. rumor/non-rumor [39], or its authenticity [41].

3.3 Datasets

Despite the increase in attention on the problem of fake news detection there are no common agreed upon benchmarks for this task. In this section, we will give a brief overview on some of the most important ones that are also publicly available.

BuzzFeedNews[4]: This dataset comprises a complete sample of news published in Facebook from 9 different verified pages related to left, right and mainstream political news organizations (3 sources each). The overall, unbalanced, dataset consists of 2,282 posts, labeled by 5 expert fact-checkers as true, false or a mixture, collected over a week during the U.S. 2016 presidential election.

CREDBANK[5]: Large crowdsourced datasets that comprises more than 60 million English tweets grouped into ~1000 real-world events, each annotated by 30 human annotators on Amazon Mechanical Turk.

FakeNewsNet[6]: The dataset uses fact-checking websites to obtain news contents for fake news and true news, and enriches the news features with social ones related to the user profile and timeline.

Unfortunately, no existing dataset can provide all possible features of interest, especially in case of social context and user profile most of the features are missing, due to privacy reasons and regulations imposed by the APIs term of service. Because of these limitations the use of those datasets, in the scope of this thesis, is not applicable, hence we need to resort to a custom pipeline for data collection.

[4]https://github.com/BuzzFeedNews/2016-10-facebook-fact-check
[5]https://github.com/compsocial/CREDBANK-data
[6]https://github.com/KaiDMML/FakeNewsNet

3.4 Position of this work

Features-oriented research for fake news detection aims to determine which features provide the most effective knowledge to distinguish between true and false news. The objective of this work is to analyze the different sources of features, namely news content and social context, and understand how those can give valuable contribution to predict the veracity of a news piece.

In particular, we further differentiate the family of social features into three sets, i.e. user-based, tweet-based and network-based, giving particular attention to network-based features. In this last case, we formulate the fake news classification problem as a node classification task using geometric deep learning and graph convolution.

Chapter 4

Methodology

In this chapter we aim to present the main ideas of this work and the high level steps that we have gone through in the development of this thesis. In order to do so, we illustrate the core idea, the goals and the research questions, that have been the starting point of this work. In addition, the reader is supplied with some definitions that can be useful for the reading. Lastly, we show the proposed data collection solution at high level, with a focus on the different macro-phases and their inputs and outputs.

4.1 Core idea

Being able to characterize and model the user-news interaction can expose many latent features such as user opinions, stance and membership to a particular community even if not explicitly declared. At the same time, the analysis of news diffusion, across different media sources, can reveal useful insights about the bias and trustfulness of the news content provider.

The core idea of this work builds on the assumption that deceptive content is intentionally written with a style that tries to imitate authoritative news source, and hence it is difficult to detect using automatic systems that rely exclusively on linguistic based features. The attempt of this thesis is to create a model that is able to distinguish between fake and true news, using a broader set of fetures, e.g. user based, network based and stance based, and analyze the importance that those features have in the classification task.

4.2 Objective and research goals

The research starts with the purpose of answering the following questions:

- Is it possible to identify fake news using only linguistic features?
- Can social context features increase performances in a significant way?
- If the answer to the previous question is yes, what features are the most discriminative and why?

The main goal of this thesis is to put to the test if the usage of non linguistic based features, in particular those related to news diffusion and social engagement, can improve the results obtained in the case of a classifier trained on representations extracted solely from news content. Moreover, among the research questions, we ask ourselves whether it is possible to scale this approach to a large social network context, like Twitter.

4.3 Social raw data

Due to multiple reasons, e.g. privacy and social network terms of service, there is a lack of available datasets that incorporate the news sharing behavior of users online. The scarcity of such data results in the necessity of developing a custom pipeline that allows us to collect a large amount of data from a social network of choice, e.g. Twitter, leveraging existing API both from internal and external services of the social platform.

In the following sections we define the problem of the data collection and introduce a general pipeline that can serve for the scope of this thesis. The high-level description aims to define all the requirements and assumptions that have been made before moving on the the actual implementation described in the next chapter.

4.3.1 Social Network choice

Among all the available social platforms that are largely used in our everyday life, the choice of Twitter can be attributed mainly to the fact that it is a text-based platform, while others like Instagram or Snapchat are image-based social network that are rarely used to share news. Other advantages of that support the choice of this social network are:

- Twitter provides free API that gives access to a large amount of data that can be easily filtered for our need
- The content found on the social platform is generally perceived as more formal with respect to other social network, e.g. Facebook
- Collection of information concerning user profile and tweets do not require an explicit consent of the user, if the user willingly decides to have a public profile

4.3.2 Twitter terminology

Here we cover the meaning of some of the terms that will be largely used in the following sections and that have a slightly different meaning than the original terms related to the social network

User : Unique account registered on Twitter

Tweet : A tweet associated to a news identified by the URL it directly refers to

Retweet : A particular type of tweet that is a re-shared version of another tweet, originally posted by another user

Quote : A particular type of tweet that is a direct mention of another tweet, originally posted by another user

User Timeline : List of tweets, in reverse chronological order, that are associated to the same user

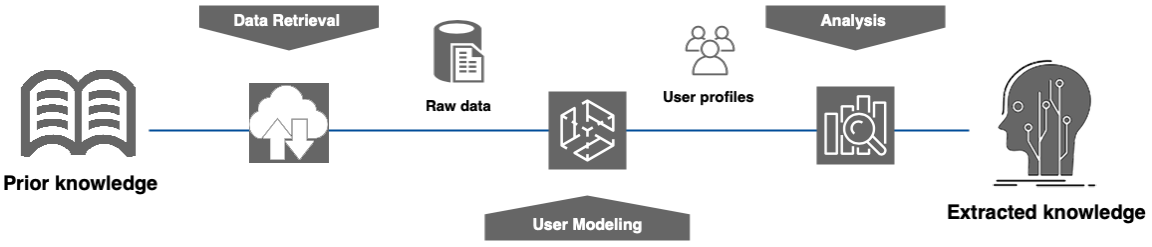
News Source : Newspaper or News-agency that belongs to set of pre-determined media sources

Article : Piece of text published by a News Source and identified by a unique URL

4.4 Data collection pipeline

The general data collection pipeline to extract raw data from Twitter can be divided into three main steps, namely data retrieval, user modeling and data analysis as shown is figure 4.1.

Figure 4.1 High level overview of the data collection pipeline



As a preliminary step, in order to collect news-tweets identified by URLs, we have to give the definition of the trusted and untrusted news sources as prior knowledge to the problem. In fact, when dealing with misinformation, the first challenge lies in the difficulty of assessing the trustfulness of the claim. For this reason, scalability is a very important aspect,

because the manual verification of the veracity of every single news piece is practically intractable. For this reason, we decide to rely on well established and trusted fact-checking organization[1][2][3] as the providers of a list of sources that have been proven to share either reliable or fake news.

In the data retrieval step, using Twitter API, a series of requests are made to retrieve information concerning both the users and the tweets that have interacted with the previously identified news sources. Additionally the information provided by Twitter is enriched using external APIs that provide useful information regarding the user profile or social behavior.

In the user modeling stage raw data is processed to extract useful and valid information in order to obtain important insights for the problem at hand.

Finally in the last step an analysis is carried out over the extracted user profiles. The underlying idea is that the information retrieved so far can give powerful insights to tackle a variety of different problems, even ones that are beyond or not related to the scope of this work. In fact, this process may be found useful to answer some of the following research question:

User characterization Analyze patterns in the user behavior and identify users accordingly, e.g. Bot vs Human or Reliable user vs Fake news spreader

Community Detection Identify group of users that share similar beliefs or that are more exposed to fake news and mistrusted sources

News Diffusion Identify patterns in the diffusion of news coming from different sources or concerning different topics and categories

4.4.1 News Sources

The first requirement for the data collection is given by the set of news sources that we intend to study throughout this thesis. The news sources have been selected among vary news content providers, e.g. Newspapers, news agencies and magazines. For each news source we start by identifying all the possible domains that are owned by that source. Besides, some sources categorize the content they provide by introducing some patterns in the URL: those patterns can than be exploited to extract categories such as *politics*, *world*, *technology* etc.

[1] https://www.snopes.com
[2] https://www.politifact.com
[3] https://www.factcheck.org

4.4.2 Data retrieval

Leveraging Twitter API we can perform a series of request to extract a set of tweets. These request can be done either using Twitter's Search API or Twitter's Streaming API. The former involves polling Twitter's data through the query of custom keywords or username. Twitter's Search API gives you access to a dataset that already exists, meaning that it returns data related to tweets that have occurred in the past. The obtained data is the one matching the searching criteria, e.g. keywords, language and location. Twitter's Streaming API, on the other hand, is a push of data as tweets happen in near real-time. The output of this step can be represented as a triplet

$$(u, T, A)$$

where u is the user account extracted along with its user profile, T is the set of news-related tweets associated to users u and A is the set of news articles referenced by tweets in T.

This component can be therefore modeled as a function $f(X) \rightarrow \{u, T, A\}$ that given as input a news article X, defined by a URL, returns as output a set of triplets (u_i, T_i, A_i) where $U = \{u_0, \ldots, u_N\}$ is the set of all users that have shared the input news X, $T_i = \{t_0, \ldots, t_M\}$ is the set of news-related tweets belonging to user u_i and $A_i = \{a_0, \ldots, a_M\}$ is the set of articles referenced by tweets in T_i and such that $X \in A_i \forall i$

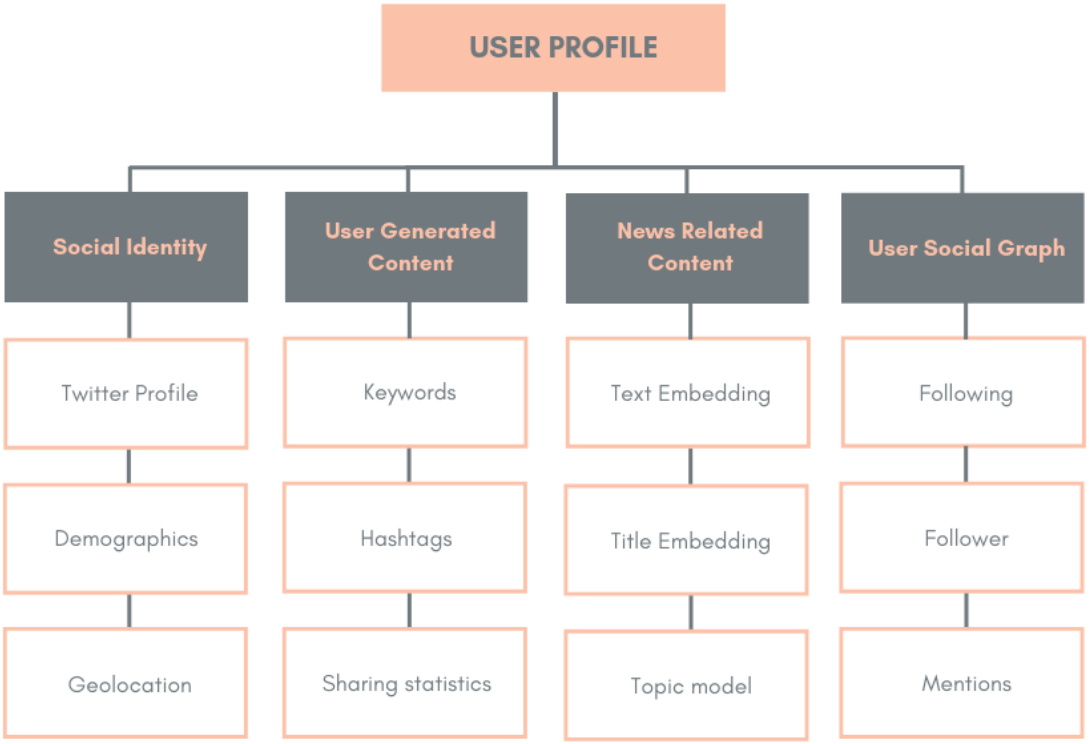
4.4.3 User modeling

The goal of the user modeling component is to organize the raw data collected by the data retrieval component, i.e. user profiles, tweets and articles, in a structured way. This new representation will not only transform the data into structured, actionable knowledge but also allow for an easier exploration of the data. To best represent all the possible facets of a user we divided the profiling of a user into four different dimensions, namely Social Identity, User Generated Content, News Related Content and User Social Graph, as depicted in figure 4.2.

Social Identity

This dimension focuses on the Twitter user information and neglects the content and social information. This part of the profile is obtained for the most part using Twitter API, which directly provides the user profile and description along with some user's statistics. Exploiting the external APIs presented in section 2.7.4 we can further enrich the social identity of a user

Figure 4.2 Proposed four dimension user profile model



retrieving an estimation of demographic information, e.g. sex, age and ethnicity, along with the country and city he or she lives in.

User generated content

This dimension focuses on the content that the user shares on the social platform either by tweeting, retweeting or quoting a status that contains a news source URL. The keywords model extracts the user's interests from the tweets in terms of tweet's frequent terms, while the Hashtag model maps every hashtag that the user tweeted in the past to its frequency. Finally, sharing statistics concern not only the number of like and tweets that a user has posted, but also identify the tendency of a user of posting "original" content as opposed to a pure retweeting user.

News related content

In this dimension the focus is shifted towards the actual content inside the articles. Documents features are obtained from the analysis of all articles and divided into embeddings models

and topic model. While embeddings models allow to represent a document as a continuous vector of fixed size, topic models represent a document as a distribution of topics that can be easily visualized and described by a mixture of words. In this dimension we do not include features concerning the news source that provided the content, because the news source is the actual provider of the target label we are interested in, and in doing so we would have data leakage.

User Social Graph

In the user social graph dimension we finally analyze the social connections that a user has in the social network and represent this information as a graph. The social relations that a user develops in the platform can be obtained from Twitter user's profile in terms of following and followers. Those are encoded as a list of either a username or a user unique identifier. Because of the exponential growth of the graph as users are added, we define an alternative social relation given by mentions. In fact, tweets can contain direct reference to another Twitter user using the notation *@username*

4.4.4 Data Analysis

Once the raw data is collected and pre-processed, the analysis can be carried out. Data analysis is the process of evaluating data using analytical and statistical tools to discover useful information and to extract novel and actionable knowledge. This newly extracted information enables machine learning models to perform the different tasks that will be presented in the case studies.

Chapter 5

Implementation

In this chapter we present the reader with the entire development of the data collection pipeline discussed in chapter 4. In the first section we start by describing the selection process of news source and later on move to the actual data collection process, which is carried out using Twitter API.

5.1 News Source

This first phase is necessary to define the list of sources that are going to be considered for this work. This collection of sources is later on used as the main filter for the requests performed to the Twitter API, and therefore it needs to be selected in a way that it is neither too broad or too narrow, since in both extremes we would not be able to capture important behaviors from the user model or the news diffusion patterns.

5.1.1 Source selection

The definition of the news source collection is performed through manual selection of sources scraped from well established fact-checking organization that associates to each source a label of either trustfulness or misinformation. This label is used as ground truth for the case study presented in the next chapters. The collection consists of 92 English speaking sources among which 50 are newspapers and 42 are news agencies. Moreover, every source is associated with a list of owned domain and since most of the URLs within the same domain follow a consistent pattern it is possible to define a *url_category_index* attribute that identifies the position (zero indexed) of the article category in the URL, splitted by character *"/"*.

Example of category identification from a Foxnews article URL

URL : www.foxnews.com/politics/trump-dossier.amp

Fox News : $\{url\_category\_index : 2\}$

Category : Politics

5.1.2 Categories

Despite having a category index for some of the news sources, every source relies on its own custom labeling structures and terminology, hence two different sources may refer to the same category by two different terms, e.g. *finance* and *business*. This lack of consistence among sources results in almost a hundred different categories. In order to obtain a better representation of them, all the different categories are regrouped into 8 macro categories defined by means of ontological classes.

Category Name	Distinct labels
Politics	6
World	14
Business	17
Sport	13
National/Local news	15
Entertainment/Arts	20
Style/Food/Travel	29
Science/Technology/Health	19

Table 5.1 Macro categories name with number of simple source-category associated

5.2 Twitter Pipeline

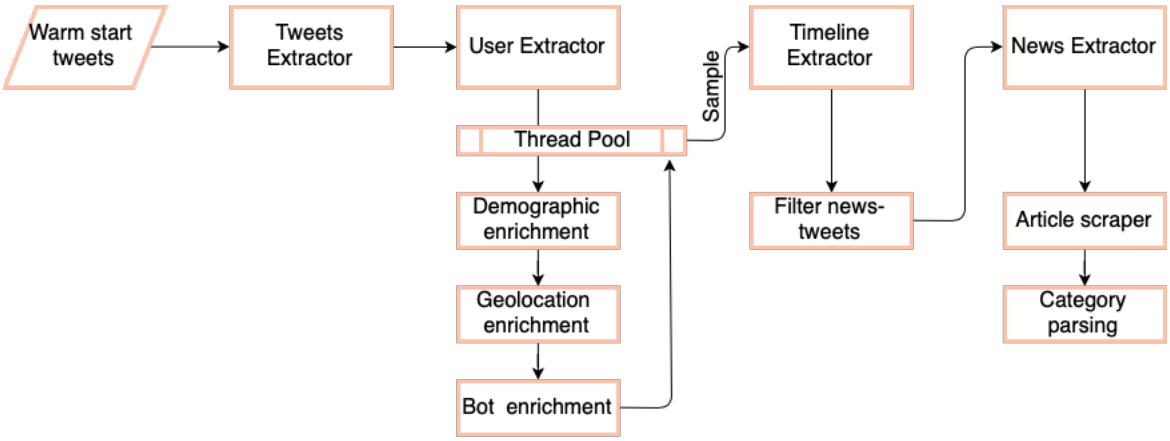
In this section we present a detailed version of the data collection pipeline. First, we give overview on the different components of the pipelines and then, for each one of them, we perform a detailed analysis. The whole pipeline is based on a Python application that manages the data collection step and interacts with MongoDB through an external library[1]

5.2.1 General overview

Adopting the previously declared notation (u,T,A) we can identify the three main components of the pipeline as the one regarding the extraction of users, tweets and news articles. A

[1]https://api.mongodb.com/python/current/

Figure 5.1 High level schematic overview of the components that together constitute the data collection pipeline



general representation of the pipeline is depicted in figure 5.1. Below a brief explanation of every single component and its functionalities:

Warm start tweets : Set of news-tweet ids used as input for the tweet extractor component

Tweet Extractor : It is a component that given as input a set of tweet ids chunks them and performs a series of requests to the Twitter API resulting in a collection of tweets. Each tweet is associated with the id of the user that sent it

User Extractor : It is a component that given as input a set of users, either username or id, makes a batch of request to Twitter API to retrieve users information. This information are later on enriched with sub-modules that query external APIs to get information related to geolocation, demographics and bot score for that user

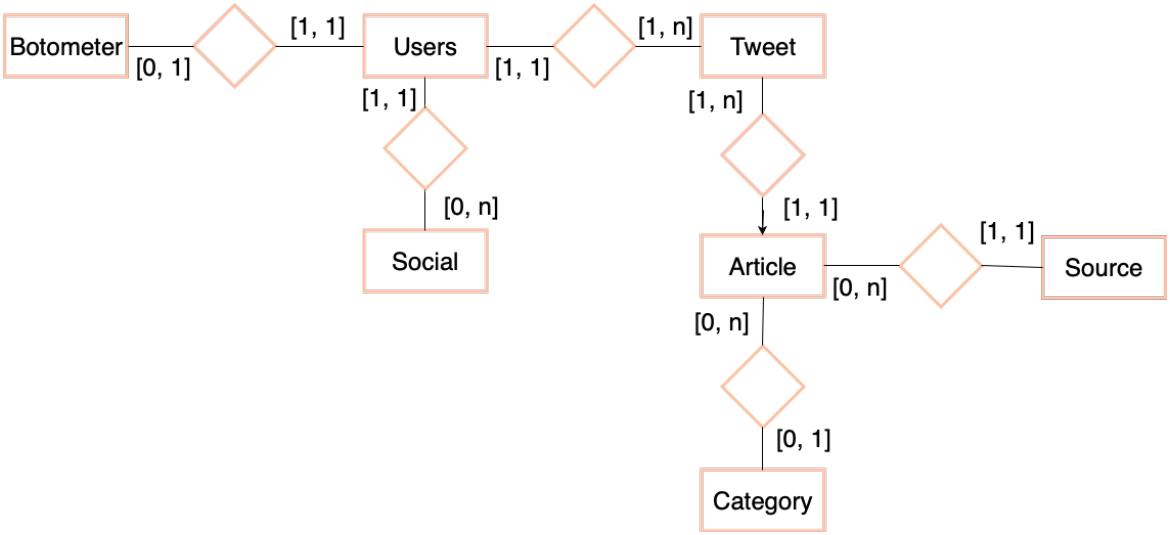
Timeline Extractor : It is a component that receives as input a sample of users ids, then, for each one of them, retrieves its timeline and filters the tweet by removing all the non news related tweets that the user has sent. The filtered tweets are passed as input to the next component

News Extractor : It is a component that given the news-tweets ids uses the newspaper3k library to scrape the URLs found in the tweet and, if possible, assigns a category to the article by parsing the URL

For efficiency reason, all the steps related to the extraction of users, timelines and articles are parallelized using the python *multiprocessing* library. Moreover, every component

interacts with the MongoDB instance to store the new data obtained at every step. The data is stored using a JSON format and the collections used for this thesis are illustrated in figure 5.2

Figure 5.2 Overview of the MongoDB collection schema



5.2.2 Warm start tweets

Because of the usage rate limit imposed by Twitter search API, a large set of tweets ids, useful to start the analysis required for this thesis, has been previously obtained using Twitter Streaming API. This set is the result of a large volume of tweets, pushed by the Twitter endpoint, which are filtered to retrieve only news-related tweets. The filtering process is done by comparing the URL contained in the tweet with the pre-selected domains present in the MongoDB *Source* collection.

5.2.3 Tweets Extractor

The tweets extractor component reads from a file the list of tweet ids obtained in the warm start step and groups them into chunks, where each chunk contains 100 ids, which is the maximum number of tweets per request that the Twitter *Statuses Lookup* endpoint allows to retrieve. Since the attention of this work is focused on English speaking articles and users, all the tweets for which is not possible to infer the language, or that have a language different from the English one, are discarded. In table 5.2 an example, taken from the URL www.politico.com/magazine/story/2019/05/15/australia-elections-rwanda-prisoners-refugee-swap-us-226875, of

the partial structure of a tweet stored in MongoDB. At this stage of the pipeline the tweet collections consists of almost 1.5M tweets.

Attribute	Value
_id	1129130162795233281
text	BREAKING: Obama admin secretly agreed with Australian PM Malcolm Turnbull to relocate 2 Rwandan terrorists
user_id	726756088
favourite_count	0
retweet_count	7198
created_at	2019-05-16 21:03:03
retweeted	True
quoted	False
news_source	Politico

Table 5.2 Example of tweet stored in the MongoDB Tweet collection.
N.B. user and tweet ids are anonymized

5.2.4 User Extractor

The User Extractor module retrieves, from MongoDB, all the distinct user ids or users screen name for the tweets obtained up to this point. Same as before, to respect the count limit imposed by the Twitter *lookup_users* endpoint user's ids are divided into chunks of size 100. For every user, Twitter returns a series of information related to the name, profile, description, location and statistics of the user. Those information, with the exception of the statistics, are directly inserted by the user and Twitter does not perform any check on them, so the veracity and consistency of this information is limited.

To speed up the execution, requests are parallelized using a pool of threads, each one of them processing multiple chunks at the same time. For every user we also define the social connection we want to study. The social type is expressed in terms of followers or following relationship and results in a complete list of users that can be used to construct a social graph. In table 5.3 we show an example of the partial structure of an anonymized user stored in MongoDB.

User Enrichment

The user data collected so far is further processed to extract additional information about demographic and geolocation. In addition, using an external API the probability of a particular user being a Bot or not is retrieved.

Demographic Enrichment The goal of this sub-module is to enrich the user profile by means of analysis over the content of the profile picture. The component first checks the existence of the user's profile image then, if the URL is a valid one, a request is sent to the *detect* endpoint of Face++ API to verify whether there is a single visible face in the profile picture: In this case, under the assumption that the profile picture is portraying the actual user, a second request is sent to the *anylyse* endpoint. In case of success the API gives back an estimate of the sex, age and ethnicity of the previously identified face.

Geolocation Enrichment As previously stated, Twitter allows users to insert personal information without any verification from the social network. One of the information that can be inserted by the user is a string defining the location of the user. Although this information is neither mandatory nor validated this submodule tries to transform the textual location, if present, into a structured representation that outlines the province, area, country name and country code (following the alpha-2 ISO codes). The enrichment is carried out using Yandex API, which first checks whether the location provided by the user is an existing one, and in case of a positive response returns a geo-location object that contains the aforementioned information.

Bot Enrichment The goal of the bot enrichment component is to extend the user information provided by Twitter by adding a set of scores that describe the probability that a certain user is a Bot, i.e. a fully autonomous entity that manages the profile by means of temporized and pre-defined actions. The external service is called Botometer, it is provided by the Indiana University, and analyses the user behavior and profile by distinguishing between English and non-english content. The analysis focuses on four different aspects, namely friends, network, temporal and users, and assigns a score to each one of them.

5.2.5 Timeline Extractor

Due to memory and efficiency reasons, only a sample of the users collected is processed for timeline extraction. The Timeline extractor uses a pool of threads to parallelize the requests that are sent to the *user_timeline* endpoint. The APIs allow to retrieve a maximum of 3200 tweets from the user's timeline with a count limit of 200 for every request. In this work we have decided to set to 600 the number of requested tweets from the timeline. The API returns the tweets in reverse chronological order, the most recent ones first, and also includes retweets and quoted statuses into the timeline. The tweets are then parsed in order to filter out tweets that are not related to any news article, i.e. do not contain any valid news source URL.

Attribute	Value
_id	7267768098
description	Conservative Female stronglifter Love my Family, Dog and America
favourites_count	137952
followers_count	4587
friends_count	4663
statuses_count	157467
created_at	2012-07-30 20:07:14
location	Key West, Fl
t_age	39
t_eth	WHITE
t_gender	2

Table 5.3 Example of user stored in the MongoDB Users collection.
N.B. user id is anonymized

5.2.6 News Extractor

Once all the news related tweets are obtained, the news extractor component deals with the scraping of the news content and metadata. A powerful library that allows for automatic news source scraping is newspaper3k which beside parsing the html page referenced by the URL offers useful natural language processing functions that are used to automatically extract keywords and tags. Moreover, the news extractor component also entails a submodule that given the URL extracts the news source and, if possible, parses the category of the article. In table 5.4 we provide an example of the partial structure of an article stored in MongoDB.

Attribute	Value
_id	6375746976652d6c6f737365732d7
authors	Matt Velazquez
category_aggregate	sports
keywords	['season', 'letdown', 'points', 'losses', 'suffer' 'trip', 'bucks', 'games', 'consecutive', 'suns']
scrape_date	2019-04-28 09:21:01
source_name	USA Today
text	Losses to the Jazz and Suns show the Bucks are susceptible to a letdown. That's all that ...
title	Bucks suffer road letdown with first consecutive losses of season

Table 5.4 Example of article stored in the MongoDB Articles collection.

A problem that we may encounter in this step of the pipeline is content duplication, which may occur because of marketing campaigns from news sources or URL shortening. In this scenario, a possible solution for the former problem can be achieved at experiments time, by simply checking that all articles have unique text corpus, while for the latter problem we can scrape the URLs and follow all the HTTP responses before proceeding with the actual data crawling.

Chapter 6

Experiments and results

The goal of this chapter is to show how the methodology and the implementation, discussed in the previous chapters, have been put to the test in different experiments. First, we present and analyze the final version of the dataset. The objective of this analysis is to assess the quality of the data and, eventually, discover anomalous behaviors due to errors in the various steps of the collection pipeline or in the news source definition phase. Then, we present two different case study where we assess the performance of several methods, in combination with different set of features. In the end, the results are discussed and compared, in order to understand which formulation of the problem is the most promising one along with the associated feature importance.

6.1 Dataset

First of all, the dataset collection pipeline described in the previous chapters is launched using an ASW EC2 instance for a two week period. The number of users, tweets and articles objects retrieved in this period are reported in table 6.1. In total, the dataset consists of more than 6M objects distributed among 9 different collections for a storage size of almost 28Gb.

Collection	**Number of objects**
USERS	415672
TWEETS.	4216523
ARTICLES	614063
SOURCES	87
CATEGORIES	8

Table 6.1 Number of objects per collection contained in the final version of the dataset

A first consideration can be done on the news sources. In fact, out of the 92 sources that were selected as valid ones, 87 of them have been found on at least one tweet. Moreover, for every macro category, we have collected at least one article. In reality, as we will see in the following sections, for each of the eight categories at least 10k articles have been scraped.

Carrying on the analysis, the discrepancy that catches the eye is the difference, in the order of magnitude, between the number of retrieved news-tweets and the number of scraped news articles. This difference is due to two main reasons:1) Multiple tweets may refer to the same URL, because of a retweeted or quoted status, 2) Many news source produce content that is not strictly textual. In fact, some of the filtered URLs are linked to videos or images posted by the news source, e.g. https://www.cnn.com/specials/live-video-0?adkey=bn. In this cases, the newspaper3k library fails to return any valid metadata and the articles are skipped, although the tweets are still stored in the collection.

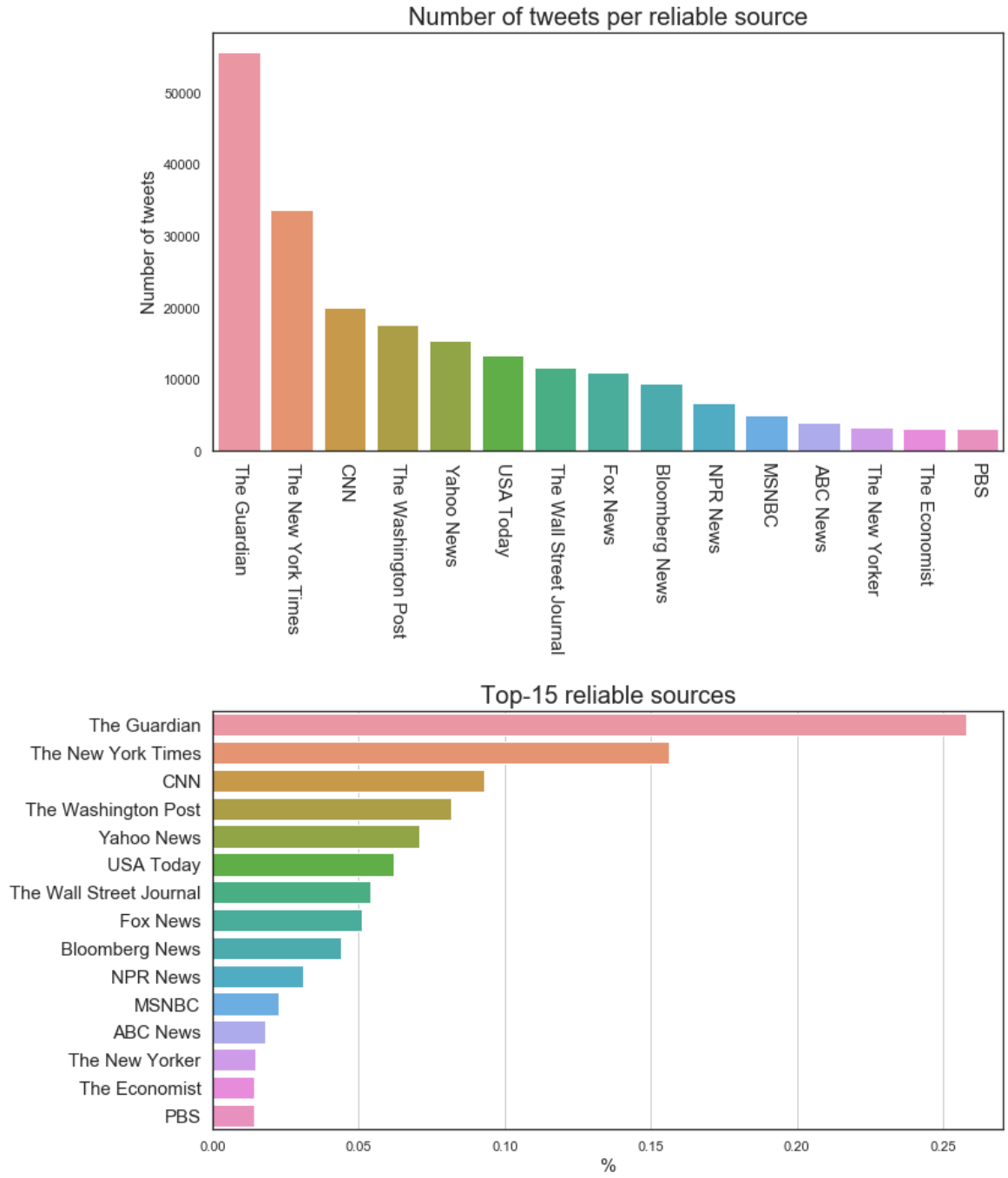
As described in the previous chapter, the pipeline collection initialization made use of a set of initial tweets. The Warm start tweets, used as input to the pipeline, are included in the ~4M tweets. The collection was carried out using Twitter Streaming API over a three month period going from January 2019 to March 2019 and the total collection contains around 1.4M tweets

6.2 Tweets

The first set of objects that we analyze are the tweets. The analysis focuses on the distribution of tweets across the different source. It is important to notice that this distribution differs from the one of the articles since multiple tweets can be associated with the same article, which is the case when a tweet referring to an article is retweeted by multiple users. The analysis on the warm start tweets is performed multiple times considering as separated fact-checking sources (figure 6.1) and sources that have a proven track record of misinformation (figure 6.2). The two plots represent respectively the absolute number of tweets that belong to each source and the percentage of tweets that are associated to each source. For visualization purposes we limit the plots to the first 15 sources, but consider all sources in the experiments.

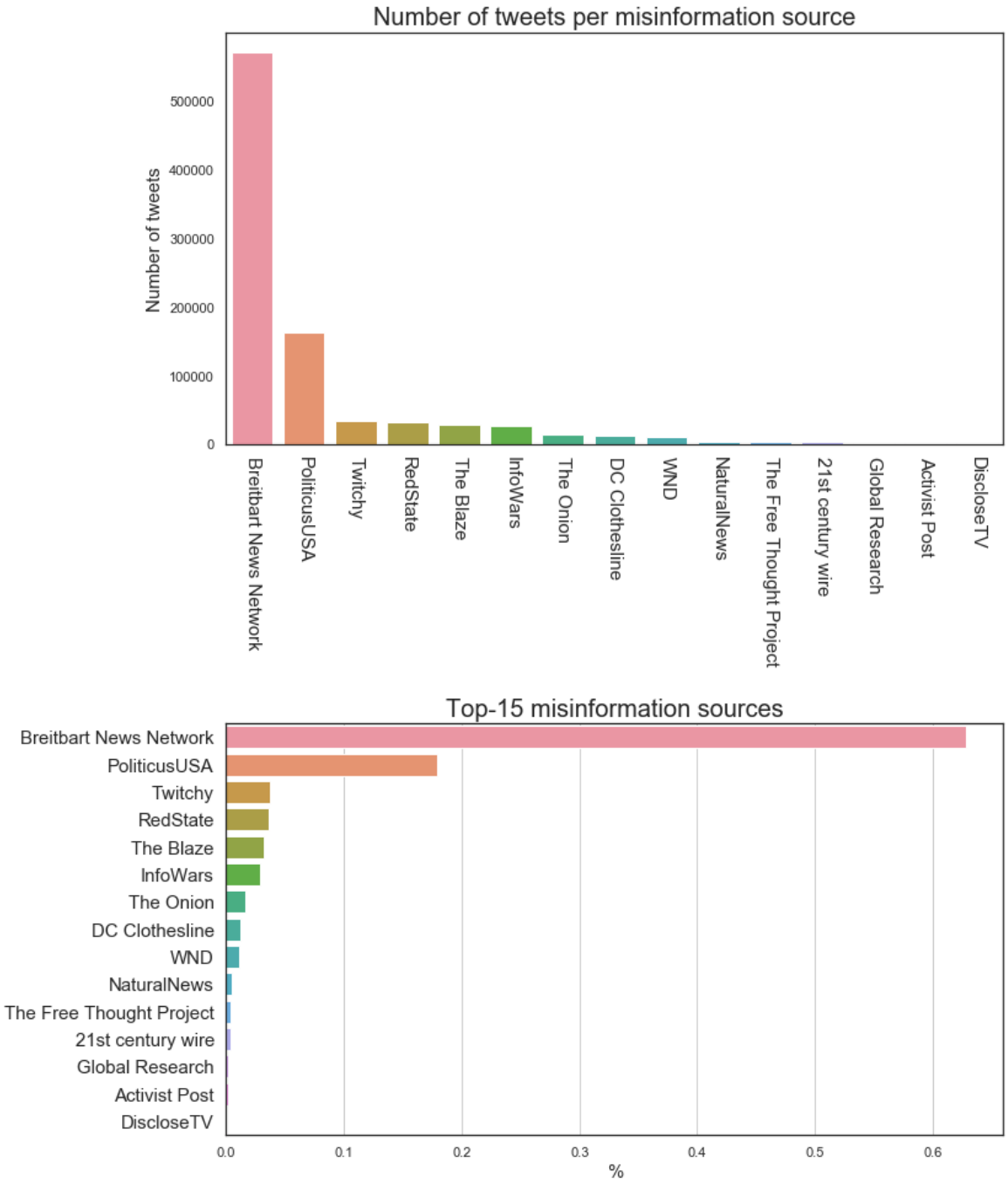
The strange distribution in figure 6.2 shows that *Breitbart* alone accounts for more than sixty percent of the total misinformation tweets. This behavior is later on identified thanks to the comparative analysis on the number of retweets and quotes statuses vs original tweets described in the following paragraph. In fact, we will see a disproportion between the number of retweets linked to articles from reliable and misinformation sources in the warm start set. In particular, in the warm start set, out of the 1.4M tweets one third is associated to retweets of articles whose source is *Breitbart News Network*.

Figure 6.1 Tweets distribution, in terms of absolute number of tweets and percentage, for the top-15 sources that are labeled as reliable



The analysis on tweets distribution is also performed on the tweets extracted from user's timeline. Especially, we want to see if the unbalanced distribution of misinformation sources introduced by *Breitbart* is leveled out or not. For the timeline extraction step, a random

Figure 6.2 Tweets distribution, in terms of absolute number of tweets and percentage, for the top-15 sources that are labeled as misinformation



sample of 166k users is considered and their corresponding distribution is illustrated in figure 6.3, given the dimensionality of the dataset the sampling is necessary mainly for efficiency and time constraints. In fact, rate limits imposed by Twitter API allow 900 requests every

15 minutes window and the timeline extraction for 40% of the users more than doubled the number of tweets in the overall dataset. Another aspect to notice is that 5 of the most important american newspapers by circulation[1] are among the top-15 sources that we have found in the user's timeline: this is an expected behavior since the majority of our users are within the USA.

Original / Retweets / Quotes analysis In this analysis we divide tweets into three different categories, namely original, retweets and quotes. A tweet t_i sent by user u_i is called original if the user u_i is not retweeting nor quoting someone else status, i.e. user u_i is the original user that sent the tweet. For the distinction between the other two categories Twitter API provides a boolean field describing whether the tweets is a quote or retweet. The distribution obtained by considering all tweets, i.e. warm start tweets and user timeline's tweets, is illustrated in figure 6.4. In general, the distribution among the original tweets and quote/retweet statuses is balanced. Insights about the dominance of *Breitbart*, for the misinformation tweets in the warm start set, are obtained by distinguishing among the warm start tweets related to reliable and misinformation sources. In fact, it is possible to notice a significant difference between the rate at which original content is produced with respect to the number of non-original statuses, i.e. retweets and quotes. Focusing on the misinformation tweets we found that out of the 571k tweets related to *Breitbart* the non-original ones were 547k, of which 480k and 67k retweets and quotes respectively. Moreover, all of these tweets are associated to roughly 3k unique articles.

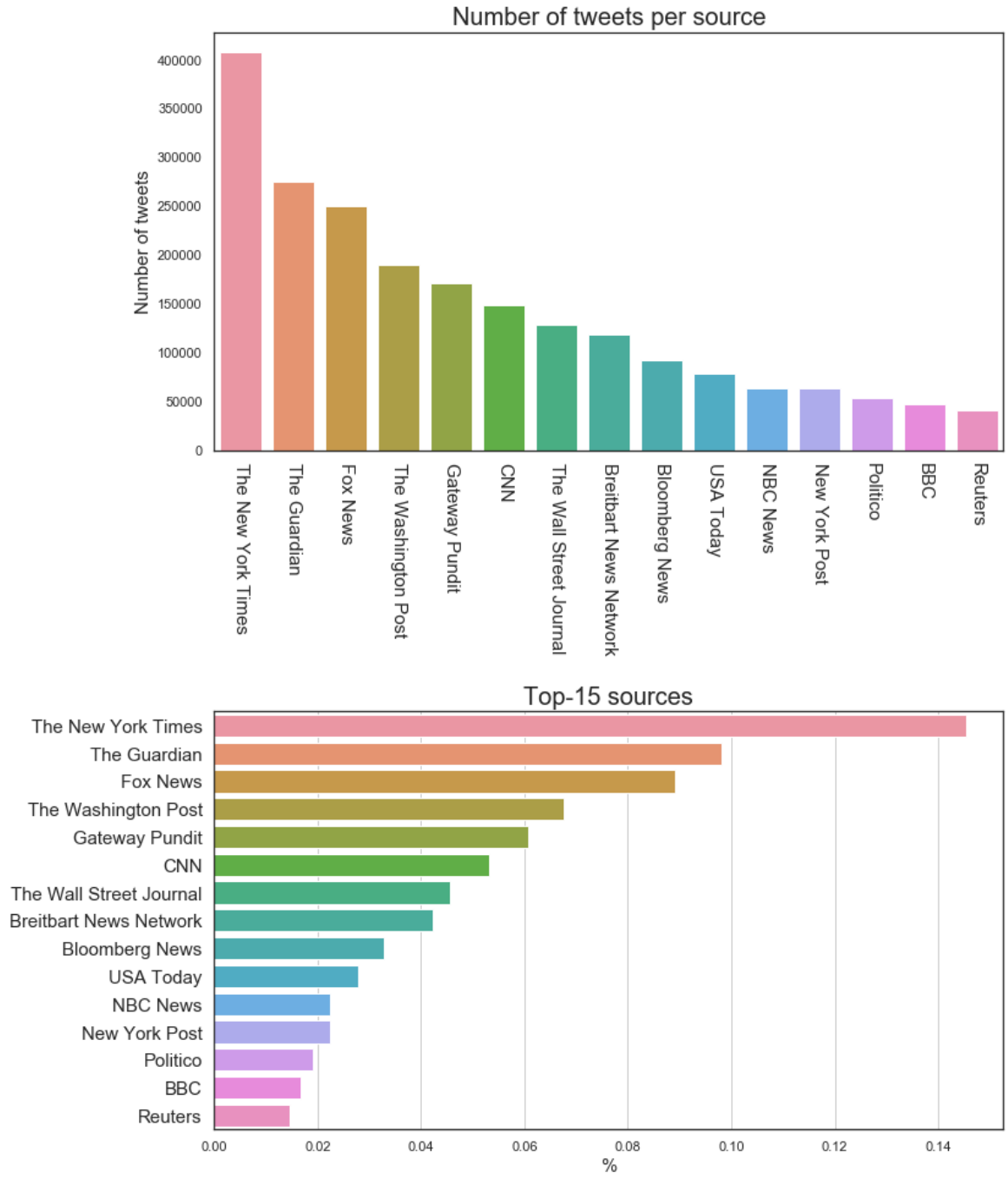
6.3 Articles and News Sources

In this section we perform a deeper analysis over the distribution of articles, sources and categories. First, an analysis, similar to the one conducted on tweets, is carried out on all the articles that belong to the dataset. The plot visible in figure 6.5 shows distribution for the top-20 sources. In this case we see that the majority of the scraped articles come from reliable sources such as *The New York Times*, *The Guardian* and *The Wall Street Journal*, while sources like *Breitbart* are outside the top-10. This result is consistent with the fact that fake news broadly spread across social network, hence a large number of tweets refers to a small number of articles.

Another level of inspection is performed over the macro categories. The goal is to ensure a reasonably balanced dataset across the 8 different classes. To perform this study we have selected all the 160K articles that have an associated category extracted directly

[1] https://en.wikipedia.org/wiki/List_of_newspapers_in_the_United_States

Figure 6.3 Timeline's tweets distribution, in terms of absolute number of tweets and percentage, across all sources



from the URL. The plot in figure 6.6 shows the different individual categories that have been reorganized to halve the number of categories. As expected, the most popular category is

Figure 6.4 Distribution of tweets type, from all sources, over the two sets:warm start and user timelines

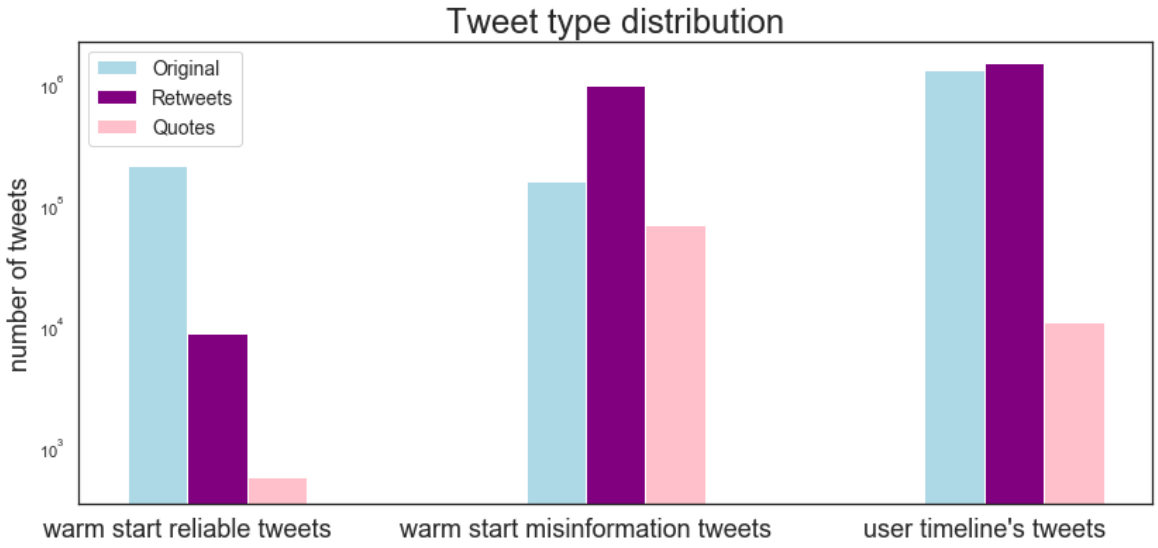
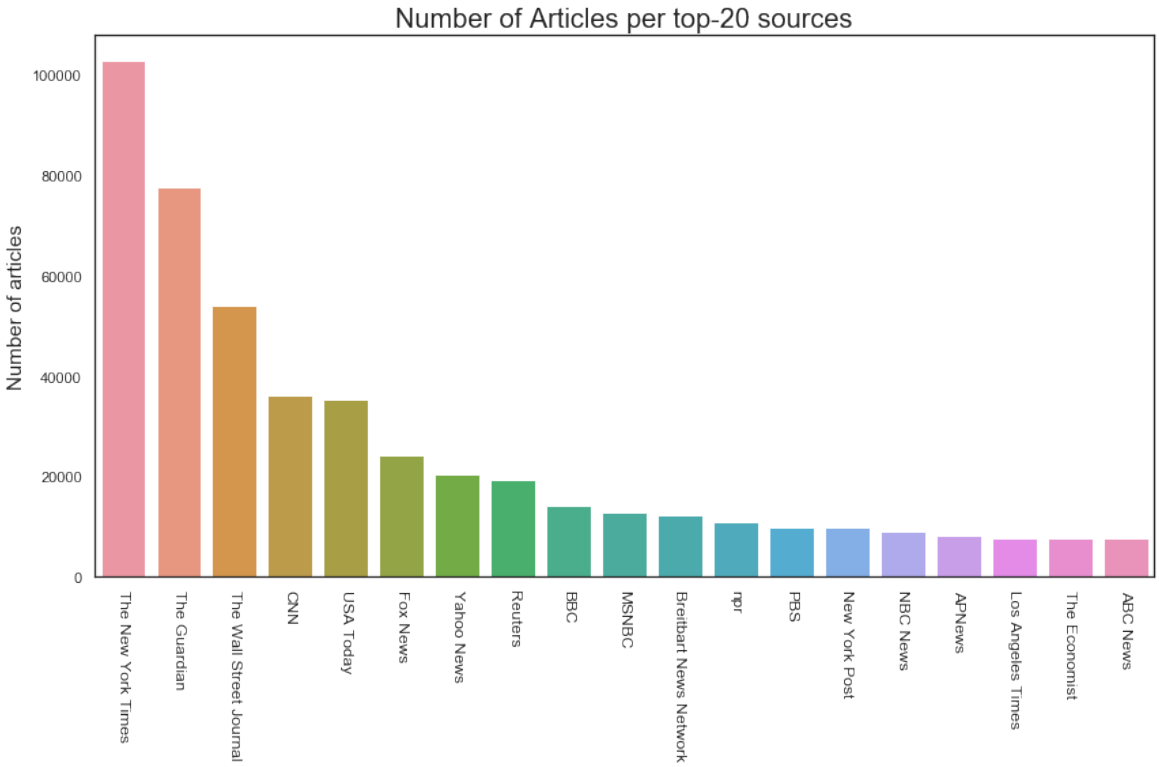


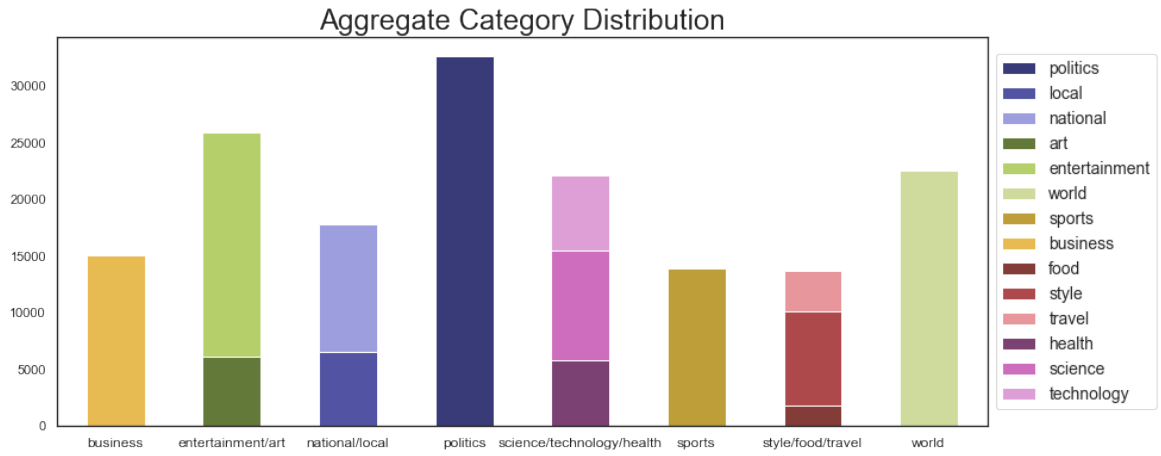
Figure 6.5 Distribution of articles across all sources



politics with more than 30k unique articles while articles about food and travel are the least common.

A possible reason to explain the difference in the number of articles across different topics may be attributed to the fact that news source are highly polarized towards politics, but this is not the case. A more realistic assumption might be that the majority of the fake news is related to topics such as politics and health, rather than sport and food. To prove that the first statement, i.e. sources are mainly focusing on politics, is not true, a second analysis is conducted on the category distribution for each individual source. This analysis shows, as expected, that some sources share mostly articles regarding single topics, which are not only related to politics. For example, most of the articles coming from *The Guardian* talk about world and entertainment, while other sources like *Fox News* and *USA Today* have their main focus on politics and sport respectively.

Figure 6.6 Distribution of article's per macro categories



A limitation that comes with this data is that categories are only available for a subset of all the articles. To overcome this problem we have to extend the analysis to all articles. To do so we shift the focus to the actual text inside the articles. To investigate the content of all the scraped articles we adopt another effective visualization technique called tag clouds[2], i.e. visual interpretation of textual data with terms frequencies mapped to word's size. The images shown in figure 6.7 are obtained as the cloud representation of keywords and tags extracted from the articles. The extraction is achieved using newspaper3k and nltk library during the news articles collection phase.

Further manual inspection is conducted on the high frequency of words, e.g. *Trump*, *president* and *White house*. A highly likely reason for which those are the most frequent words is that: 1) the most common category is politics and 2) the majority of the users analyzed in this work come from English speaking countries, in particular from USA. In

[2]https://en.wikipedia.org/wiki/Tag_cloud

(a) Article's keywords (b) Article's tags

Figure 6.7 Tag cloud visualization on keywords and tags extracted from news content

fact, by looking at the articles we found that tags like China and Congress mostly refer to the two main political news related to the USA, scilicet the trade war between Trump and the Chinese government and the Congress denial to fund the Mexican wall.

6.4 Users

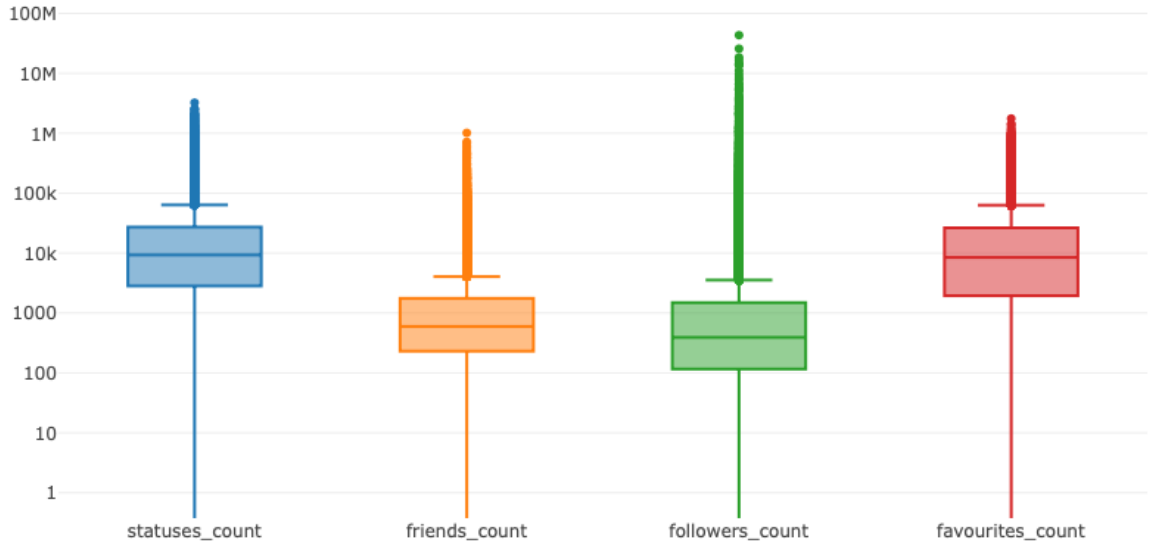
In this section we examine the user's profile using features that are either directly provided by Twitter API, e.g. statistics related to the count of friends, followers, statuses and favorites, or returned by the demographic and geolocation enricher component.

Twitter First of all, since the statistical features returned by Twitter API are all numerical box plots[3], i.e. in which the three horizontal lines of the rectangle represent the median, the second and third quartiles, prove to be a suitable visualization technique.

Second, we perform manual exploration over outliers to shine a light on anomalous accounts and identify patterns that explain this behaviours in the user's profile. From the plot in figure 6.8 we see the presence of many outliers whose followers count is above 10k. Those accounts are retrieved and a manual inspection is carried out considering the description provided in the user profile. We find out that most of these accounts are connected to celebrities or important people in different fields, e.g. Sandra Smith, Co-anchor of @AmericaNewsroom, Josh Rogin, Columnist for the Washington Post, and actor and producer James Woods. Other outliers are identified as non individual users. For example, the Twitter user associated to *The New York Times* has 43M followers and 0.5M published statuses, while *Guardian News* has almost 3M followers and 0.2M statuses.

[3]https://en.wikipedia.org/wiki/Box_plot

Figure 6.8 Box plot visualization of the user's features retrieved through Twitter API



Another analysis is conducted on sharing statistics, which are defined as the set of features that characterize the user sharing behaviors. In particular, we focus our analysis on the users for which the timeline has been extracted. The plot in figure 6.9 shows the number of tweets per user, from the image we can notice a significant number of users that falls inside the first bin (range [1, 43]). The average number of tweets per user is around 24 and this can be considered as a good indicator of the user's real preferences and interests.

The second plot 6.10 illustrates the distribution of users with respect to the user retweet rate. The distribution highlights the presence of two distinct user groups, which we will refer to as original and pure retweeter. The former contains users that generate original content and do not share news sent by others, hence the retweet rate is zero. The latter consists of users that spread news by quoting or retweeting other user's statuses, hence the retweet rate is one.

Demographic The Face++ features enrichment component used to estimate the age, sex and ethnicity of users from their profile images was successful for 45% of the users in the dataset. The distribution of those users is visible in the pie charts in figure 6.11. We see that the external API had some difficulties in estimating the sex of a user from its profile image. For what concerns the age estimation the distribution seems plausible as it identifies almost half of the users as people above their forties while the others are mostly estimated in the two ranges 20-30 and 30-40. Above 80% of the ethnicity distribution comprehends either

Figure 6.9 Histogram with 50 bins and logarithmic scale of users distribution per number of tweets

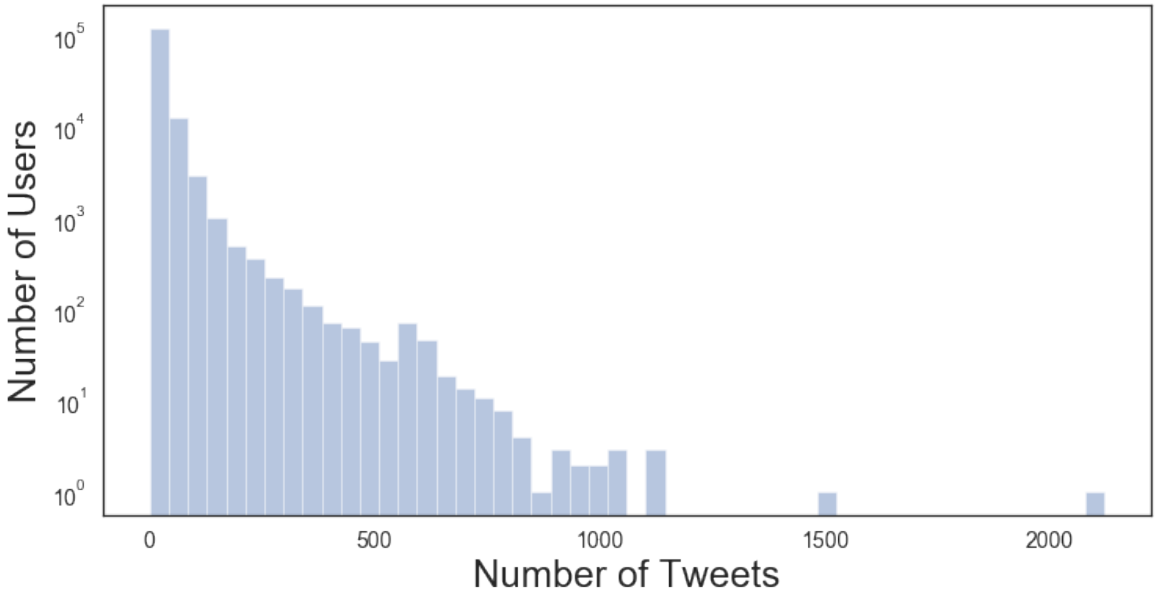
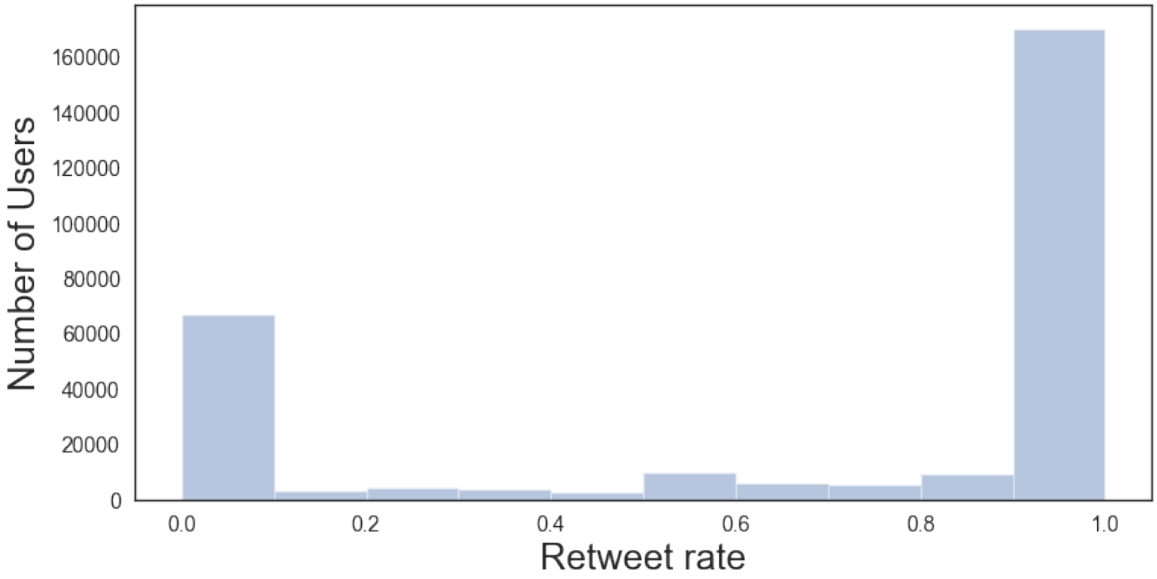
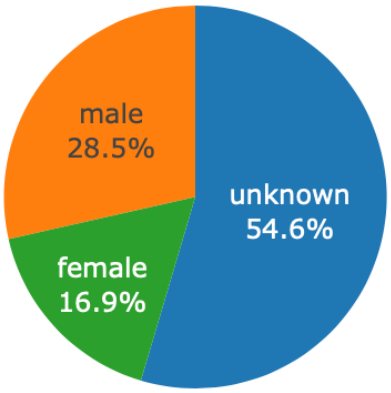


Figure 6.10 Histogram with 10 bins of users distribution per retweet rate

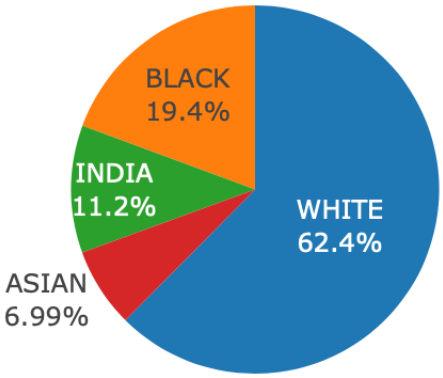


white or black people, and this is consistent with the idea that the majority of the users in the dataset come from the USA[4].

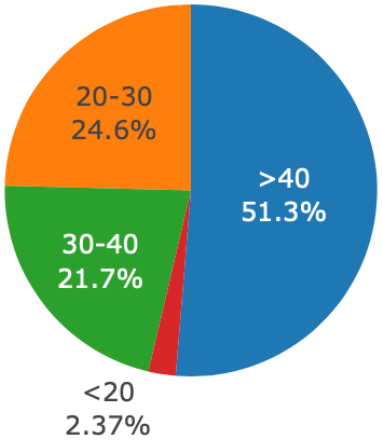
[4]https://en.wikipedia.org/wiki/Race_and_ethnicity_in_the_United_States



(a) User's estimated sex distribution



(b) User's estimated ethnicity distribution

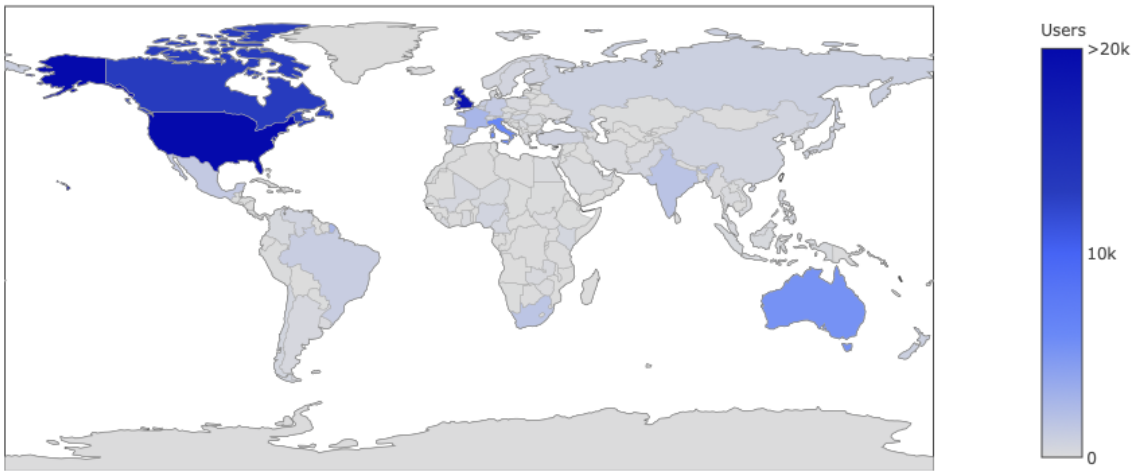


(c) User's estimated age distribution

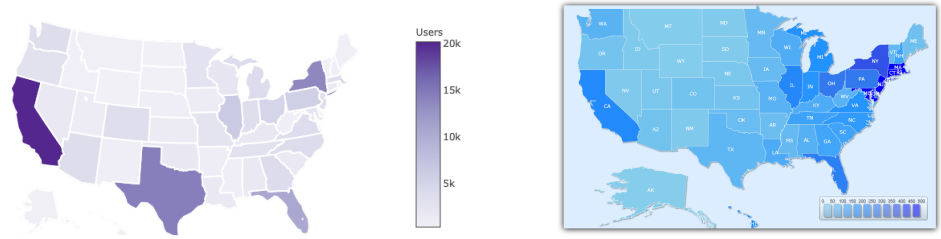
Figure 6.11 Distribution of Face++ estimated features

Geolocation The Geolocation enrichment, performed through Yandex Geolocaction API, was successful for 60% of the users. By means of a visual analysis with color maps it is possible to see that the majority of the users are located in the USA (~170k) or come from an English speaking country, e.g. Great Britain and Australia, but despite that we still have a non neglectable share of users (~50k) that are located, or originally from, non-english speaking country, namely France, Spain, Italy and South Africa. This is easily understandable from figure 6.12.

Figure 6.12 World color map visualization of the user's location retrieved through Yandex Geolocation API



A second analysis is conducted only on USA users, and their distribution across different states is compared with the demographic map of the USA. From the two images we can conclude that the users distribution is representative of the different country, since states such as California, Texas, New York and Florida show the highest number of Twitter users (>10k).



(a) Users distribution according to their estimated location for USA

(b) population density per square mile for USA

Figure 6.13 USA per country users and population density distribution

6.5 Case studies

Extracting relevant information from the dataset presented so far, we focus our attention on supervised learning tasks; In particular we investigate two different case studies:

Article classification : Label articles as true or misinformation exploiting textual content and social features

User classification : Label users as reliable or not based on the shared content and user profile model

Since the tasks at hand are supervised ones we need to define our ground truth labels. In the first case, the label is inherited directly from the prior knowledge, described in the data collection chapter, related to the news source that generates the content. In the second case, since we do not have any prior knowledge on the users, we assign the label based on the user's ratio of shared true news-tweets.

For each task, we train, using 10-fold cross validation, the different models described in section 2.4.2 and visualize their performance using a confusion matrix[5] similar to the one shown in table 6.2

		Predicted value	
		P	**N**
Actual value	**P**	True Positive (TP)	False Negative (FN)
	N	False Positive (FP)	True Negative (TN)

Table 6.2 Schema of a confusion matrix

where the different sections of the matrix have the following meanings:

- True Positive (**TP**): when predicted fake news pieces are actually annotated as fake ones
- True Negative (**TN**): when predicted true news pieces are actually annotated as true ones

[5]https://en.wikipedia.org/wiki/Confusion_matrix

- False Negative (**FN**): when predicted true news pieces are actually annotated as fake ones
- False Positive (**FP**): when predicted fake news pieces are actually annotated as true ones

By formulating fake news detection as a classification problem, the metrics adopted to assess the goodness of a model are two, namely accuracy and AUROC. The former is defined as

$$ACC = \frac{TP + TN}{N}$$

where N is the total number of predictions. The latter is defined as the area under the Receiver Operating Characteristic (ROC) curve, i.e. curve that compares the performance of probabilistic classifiers by looking at the trade-off between the true positive rate TPR and the false positive rate FPR for different threshold values. TPR and FPR are defined as follows:

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

$$FPR = \frac{FP}{FP + TN}$$

and adopting the formulation on [16], the AUC is defined as:

$$AUC = \frac{\sum(n_0 + n_1 + 1 - r_i) - n_0(n_0 + 1)/2}{n_0 n_1}$$

where r_i is the rank of the i-th fake news piece and n_0 (n_1) is the number of fake (true) news pieces.

6.5.1 Article Classification

In this section we focus on the first case study. Our goal is to build a model that receives as input an article and predicts whether it is misinformation or not. First of all, using the approach presented in chapter 4 we identify two sets of sources, which are listed in table 6.3, that are the provider of the ground truth label. We achieve this by separating URLs in two main categories based on the websites they link to: either websites containing misinformation or traditional, fact-based, news outlets. Because of this, we have to discard all news source proxy features, i.e. features related to news source that have a high correlation with the target, otherwise we would introduce a problem of data leakage.

To perform article classification we evaluate multiple scenarios based on different sets of features, namely textual content, topics and social engagement. In particular, concerning

True source (60)	**Misinformation source** (32)
The Wall Street Journal, The New York Times, New York Daily News, The Washington Post, Chicago Sun-Times, The Dallas Morning News, Chicago Tribune, Newsday, Huston Chronicles, Orange County Register, The Mercury News, The Philadelphia Inquier, Star Tribune, The Arizona Republic, The Plain Dealer, Los Angeles Times, USA Today, The Denver Post, Las Vegas Review-Journal, The Star-Ledger, Tampa Bay Times, Honolulu Star-Advertiser, The San Diego Union-Tribune, The Boston Globe, CNN, NBC News, ABC News, CBS News, Al Jazeera, Bloomberg News, NPR News, Fox News, Politico, Voice of America, United Press International, Time, Reuters, Vox, APNews, BBC, npr, MPR News, Lowell Sun, The Nation, Masslive, News Boston, WRAL, The Atlantic, Fortune, Cambridge Day, CBC News, News Channel 8, The Guardian, HillReporter, The Globe and Mail, Yahoo News, MSNBC, The New Yorker, The Economist, PBS	New York Post, The Blaze, Newsmax, Newsweek, The Onion, Breitbart News Network, Huffington Post, Metro Daily News, NTK Network, Daily Wire, Gateway Pundit, RedState, DC Clothesline, InfoWars, PoliticusUSA, Activist Post, The Free Thought Project, WND, Before it's news, Lewrockwell, 21st century wire, Twitchy, NaturalNews, GOVERNMENT SLAVES, Global Research, World Truth, AnoNews, ClickHole, Veterans Today, DiscloseTV, RealFarmacy, Gomerblog

Table 6.3 List of news sources names that have are considered as reliable or misinformation

textual content we focused our attention on the corpus of the article, i.e. the main text, and the title, which is often a good proxy of the actual content.

Articles Text and Title

For this task, we start by sampling at random, from the *articles* collection, a balanced set with 106k articles. Each article is then pre-processed using gensim and ntlk library to remove punctuation symbols and english stopwords. The vectorized representation of the document, using either text or title, is obtained from a Doc2Vec model trained for 10 epochs and with a fixed size representation of either 300 or 100. The different models are trained and evaluated for every configuration and the result are visible in table 6.4, where the values in the tuple (x, y) represent accuracy and AUROC, respectively.

From this table we can see that Random Forest outperforms all the other models and that the features extracted from the corpus of an article allow to train better classifiers then features extracted solely from the title. In this case, we do not perform any parameter tuning on the models, although this could lead to a small improve in performance. As a result of this first experiment we can see that a classifier trained solely on content features is not able to identify well fake news. As a starting point, we adopt the accuracy and AUROC of Random Forest as a baseline for all the further article classification approaches.

Features Type **Model Name**	Text 300	Text 100	Title 300	Title 100
Naive Bayes	(0.56, 0.52)	(0.56, 0.52)	(0.53, 0.55)	(0.53, 0.54)
Random Forest	(0.69, 0.76)	**(0.71, 0.78)**	(0.59, 0.64)	(0.59, 0.62)
SGD	(0.51, 0.53)	(0.52, 0.53)	(0.51, 0.53)	(0.51, 0.55)
Decision Tree	(0.57, 0.57)	(0.58, 0.58)	(0.53, 0.53)	(0.52, 0.52)
Logistic Regression	(0.53, 0.55)	(0.53, 0.55)	(0.50, 0.54)	(0.52, 0.55)

Table 6.4 Performance measure in the form of (Accuracy, AUROC) for article classification on the different features configurations obtained using Doc2Vec on either the text or title of the article and a representation of size 300 or 100.

Articles social engagements

In the second experiment, in order to include features related to article's social engagement, we decided to model the relations between articles by means of a graph $G = (V, E)$, where $V = \{a_1, \ldots, a_N\}$ is the set of N articles that we want to classify and $E = (e_{ij})$ is the set of edges. In our analysis, we consider a set of M users $U = \{u_1, \ldots, u_M\}$ where each user u_k is associated to a set of shared articles $A_k = \{a_{k1}, \ldots, a_{kT}\}$. Then, we create an edge e_{ij} between two articles (a_i, a_j) if it exists at least one user u_k that shared both articles a_i and a_j, meaning that $a_i, a_j \in A_k$. To construct the graph, we evenly sample $N \approx 80K$ articles from the user timeline's tweets and create two structures that map users to articles and viceversa, the set of edges E is obtained as a result of their combination. In table 6.5 we report a complete summary with all the graph dimensions.

Name	**Size**
N	80742
M	95904
$\|E\|$	4.848.032
d	100 / 300
Train set	48444
Validation set	16148
Test set	16150

Table 6.5 Summary over articles graph dimensions

Machine learning and deep learning models are not directly applicable to graph structures, hence we resort to a new set of models that belongs to the family of Geometric Deep Learning[7], which extend well known and efficient models, such as neural networks, to the non-Euclidean graph domain. In particular, to perform node classification we use

Graph neural networks, a multi-layer architecture that aggregates the information of local neighborhoods.

The general aggregation function for GNN is the following:

$$h_v^k = \sigma(\mathbf{W_k} \sum_{u \in N(v)} \frac{h_u^{k-1}}{|N(v)|} + \mathbf{B_k} h_v^{k-1})$$

where matrices $\mathbf{W_k}$ and $\mathbf{B_k}$ are the trainable parameters at the $k-th$ layer in the neural network, σ is any non-linear activation function (e.g. ReLu or Tanh) and h_i^k is the embedding of node i at layer k, such that the input of the first layer and the output of the last one are $h_i^0 = x_i$ and $h_i^K = z_i$. Furthermore, since the same aggregation parameters can be shared between all the nodes, the network has the ability to generalize to new unseen elements.

The overall architecture used in our experiments is illustrated in figure 6.14 where the convolution operation is carried out using the method proposed by Kipf[20], which, in matrix notation, can be written as:

$$\mathbf{H}^{(k+1)} = \sigma(\hat{\mathbf{D}}^{-\frac{1}{2}} \hat{\mathbf{A}} \hat{\mathbf{D}}^{-\frac{1}{2}} \mathbf{H}^{(k)} \mathbf{W_k})$$

where the adjacency matrix $\hat{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ is modified to include self-loops for each node, and the normalization $\hat{\mathbf{D}}^{-\frac{1}{2}} \hat{\mathbf{A}} \hat{\mathbf{D}}^{-\frac{1}{2}}$, overt its associated diagonal degree matrix D, preserves the scale of the features while making all rows sum up to one.

The model, built using the pytorch geometric library, receives as input the graph G, represented by its adjacency matrix $A \in \mathcal{R}^{N \times N}$, and the features matrix $X \in \mathcal{R}^{N \times d}$, i.e. matrix whose rows are the Doc2Vec representation of each article, and outputs a matrix $H \in \mathcal{R}^{N \times 2}$ with the predicted label for each article.

The model training is performed using Adam, with a learning rate of 0.01 and a weight decay of $5e^{-4}$. Moreover, nodes are randomly divided into train, validation and test set and early stopping is adopted to prevent overfitting.

In our experiments we considered two possible dimensions for the input features matrix, specifically 100 and 300, and the measured accuracies are 85.5%and 86.6%, respectively. Instead, in figure 6.15, we illustrate the confusion matrix and ROC curve measured over the test set in the two different configurations of X.

The results show a significant performance improvement for the graph neural network, that despite its low complexity, it is able to correctly classify more than 85% of the articles combining social engagement and content features. In order to further understand if this increase in performance is to be attributed to the social engagement alone, or to the combination of the two different families of features, we retrain the model over a different features matrix. In this case, the matrix $X \in \mathcal{R}^{N \times 5}$ is obtained using as features the Local Degree

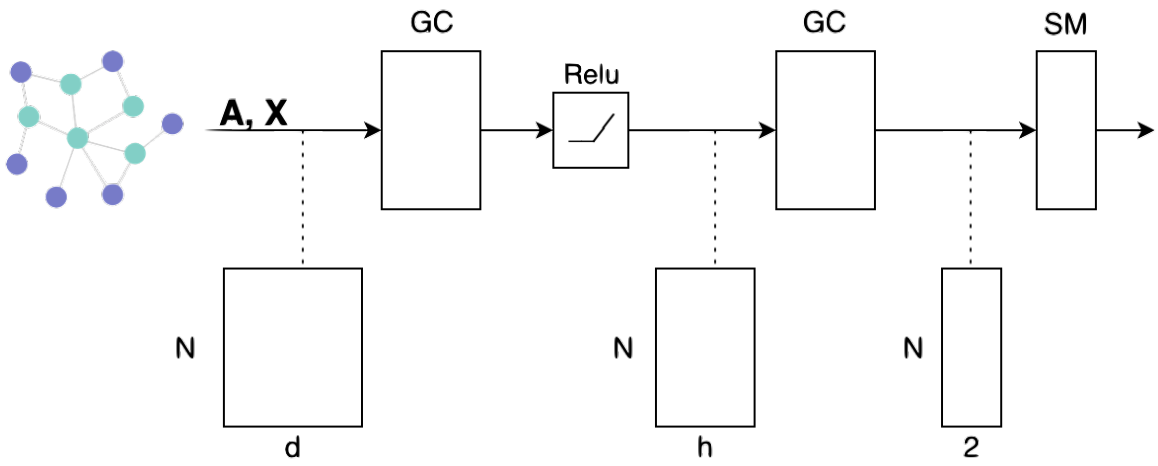


Figure 6.14 The architecture of our 1-layer graph neural network. Top row: GC is the graph convolution operation, SM is the softmax layer and Relu is the activation function. Bottom row: input/output tensor received and produced by each layer, in our case $h = 16$

Profile[8], that only provides statistical information about the neighborhood of node i, whose attributes become:

$$\mathbf{x}_i = \{\deg(i),\ \min(DN(i)),\ \max(DN(i)),\ \text{mean}(DN(i)),\ \text{std}(DN(i))\}$$

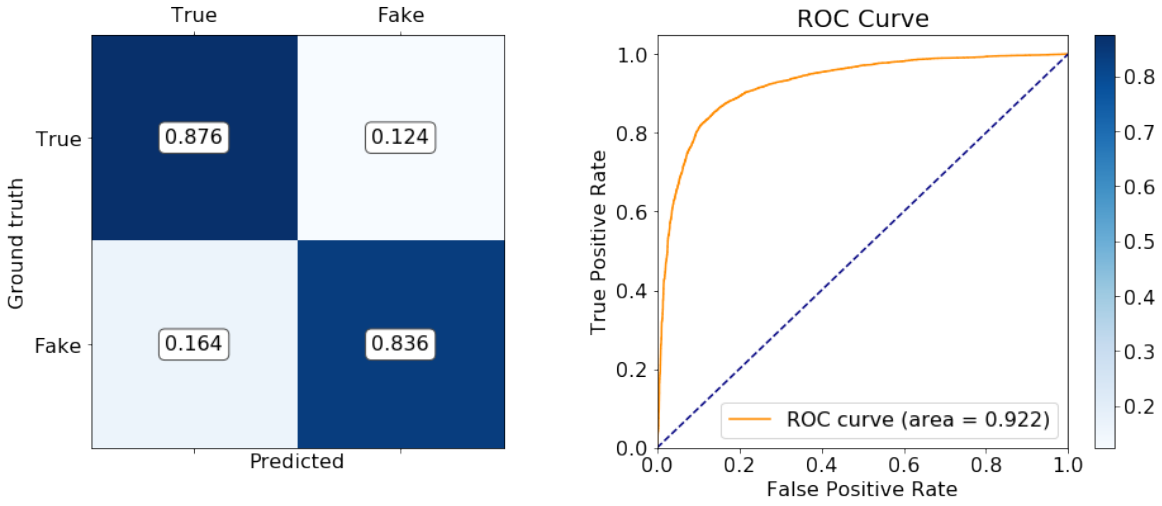
where $DN(i) = \{\deg(j) \mid j \in \mathcal{N}(i)\}$ is the set containing the degree of all neighbors for node i. As a result, we observe a noteworthy drop in performance that brings accuracy and AUROC down to 67.0% and 0.72, respectively. From this, we can conclude that social relations between articles provide useful features, but the increase in performance is to be attributed to the combination of content and social features.

Articles tweet

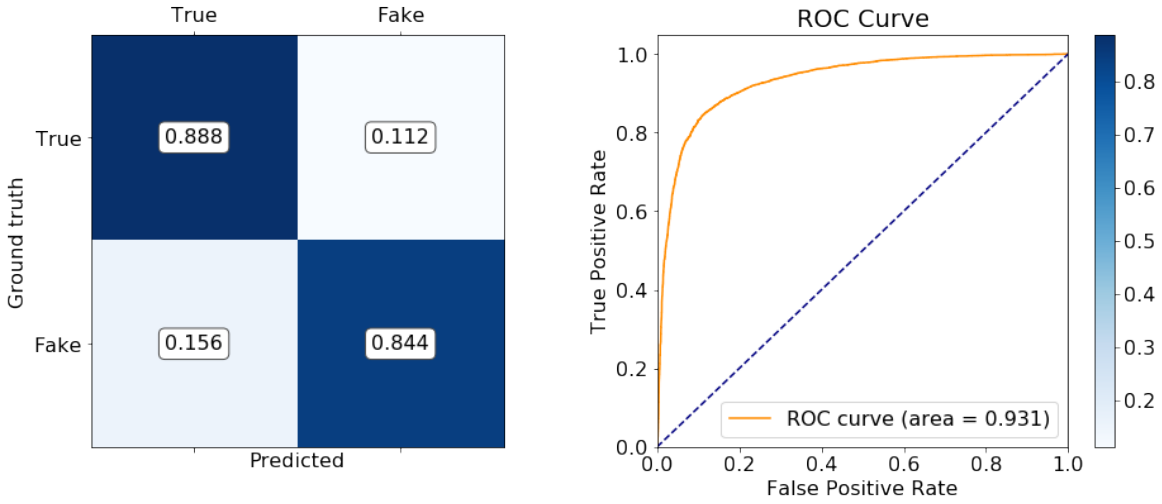
In this third experiment we shift the focus to the news tweets rather than the articles. In this context, tweets inherit the ground truth label from the article that they refer to and the classification is performed at this new level.

We start the analysis by sampling, from the user's timeline, a balanced dataset with ~100k tweets. In case of tweet classification, we neglect the article textual content to focus on the social features, as described in table 6.6, that are related to both tweets and users.

A sanity check is performed on the so extracted dataset to ensure that tweets type, i.e. retweet and quote, across the two different labels are not too unbalanced. The inspection shows that this is the case, since we have 25k and 36k retweets associated with reliable and



(a)



(b)

Figure 6.15 Confusion Matrix and ROC curve plots of our 1-layer graph neural network in the two different configurations of features matrix with $d = 100$ **(a)** and $d = 300$ **(b)**

misinformation sources, respectively. Before training the models, we need to perform some pre-processing. We cast boolean features to binary values, impute the missing values on age and gender, using mean and mode, and perform one-hot encoding on ethnicity. Moreover, standard scaler normalization is applied to all features. In this experiment, similarly to article classification, Random Forest stands out as the best model with an accuracy of 68%, and a complete report of the performance measured on the five models is given in table 6.7.

To better understand how the different features related to tweets and users affect the classification task, we perform permutation importance using the *eli5* library, which measures

Name	**Description**
statuses_count	int - number of tweets sent by the user
friends_count	int - number of accounts that follow the user
followers_count	int - number of accounts that the user follows
favourites_count	int - number of statuses that the user has favorited
retweeted	boolean - whether it is a retweet or not
quoted	boolean - whether it is a quote or note
favourite_count	int - number of users that favorited the tweet
retweet_count	int - number of users that retweeted the tweet
polarity	float - polarity score over tweet message
subjectivity	floar - subjectivity score over tweet message
t_age*	int - estimate of the user age
t_eth*	string - estimate of the user ethnicity
t_gender*	int - estimate of the user gender
bot_categories*	set - botometer scores associated to different aspects of user behaviors
display_scores*	set - botometer overall classification results

Table 6.6 Description of features used for tweets classification. Features are related to the tweet itself or to the user that sent the tweet.

Model Name	Accuracy	AUROC
Naive Bayes	0.61	0.65
Random Forest	**0.68**	**0.73**
SGD	0.62	0.67
Decision Tree	0.60	0.60
Logistic Regression	0.63	0.67

Table 6.7 Models performance measures in case of news-tweet classification

the change in accuracy when a feature is not available. The importance of a feature is associated with a weight, and in table 6.8 we outline the most important features found using Random Forest as the candidate model.

From this result, we can notice that both tweet and user related features provide useful knowledge to predict the veracity of the news, although the best proxy for this task is given by the reliability of a user, measured in terms of the number of followers.

* This feature may be missing

Weight	Feature name	Feature type
0.0416 ± 0.0014	followers_count	user
0.0169 ± 0.0019	retweet_count	tweet
0.0168 ± 0.0019	friends_count	user
0.0148 ± 0.0013	favourites_count	user
0.0133 ± 0.0013	statuses_count	user
0.0058 ± 0.0022	retweeted	tweet
0.0045 ± 0.0011	polarity	tweet
0.0041 ± 0.0012	display_scores_english	user

Table 6.8 Permutation importance feature weights obtained in case of Random Forest as candidate model

6.5.2 User classification

In this section we move to the problem of user classification. The goal here is to label users as reliable, i.e. the news shared are mostly true ones, or not, using features derived from the shared content and the user profile model. To create a proper dataset for this task, we start by gathering, for each user i, all the tweets $T_i = \{t_{i1}, \ldots, t_{iN}\}$ that we have collected for that user. In particular, we count how many of those tweets are associated with valid and misinformation sources and how many are retweets. At this stage, the dataset contains all 417k users in our collection, but for some of them we have few tweets, therefore we set a threshold $k = 10$ on the minimum number of tweets a user must have to be considered in the following phases. As a result the collection shrinks down to ~89k users, where the average number of tweets and retweets are 38.04 and 22.10, respectively.

The ground truth on a user is then inferred from the ratio of tweets associated with true articles, in our case, we consider a user reliable if this ratio is above 60%. In addition, as we have seen in previous chapters, users tend to have two distinct behaviors, i.e. share original content or only retweet others people statuses, and we report the label distribution in case of those behaviors in table 6.9.

User behavior	Reliable	Unreliable	Size
Share original content	87.60%	12.40%	7072
Only retweetes	34.69%	65.30%	37506
All users	51.88%	48.12%	89344

Table 6.9 Distribution of reliable and unreliable users in case of a pure retweet and original content sharing behavior

Despite the significant difference between the percentages of users that share original content, it is important to notice that those users account for less than 8% of the total number

of users, and hence we proceed with our analysis considering the case of a balanced datasets. Nonetheless we will conduct additional tests also on this subset of users, in order to spot any form of overfitting. For the user classification task we will focus on different set of features and highlight the increase in performance that it is due to the different features. For this particular case study we will train and compare only two different models, namely Random Forest and CatBoost.

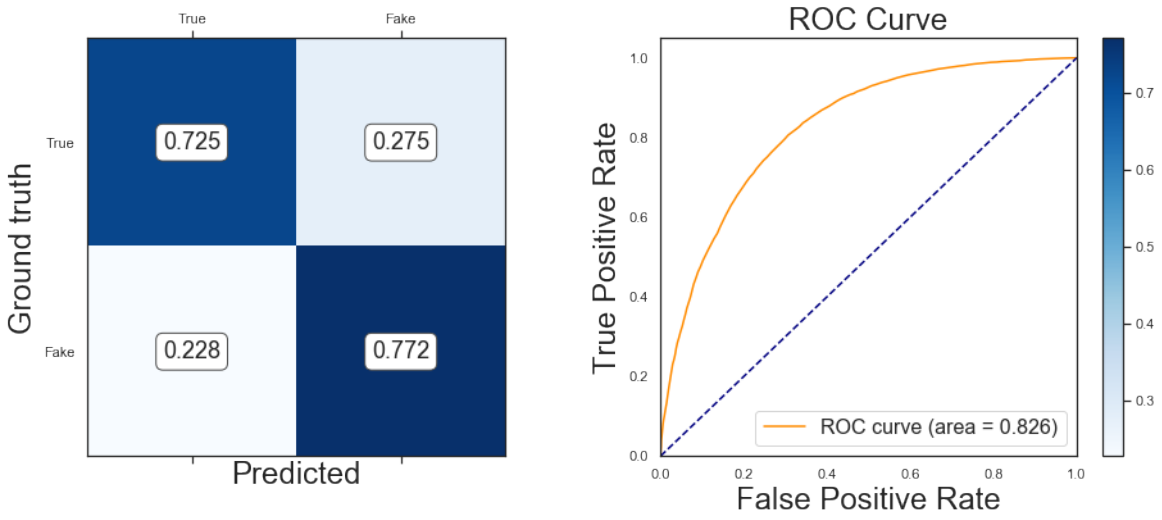
User profile features

To perform the first set of experiments, and define a baseline, we enrich the dataset with features related to the user profile only. In this case, the set of features is limited to: statistical features provided by Twitter, e.g. counts on the number of followers, following, statuses and favorites, and the same additional features, shown in the last five rows of table 6.6, provided by the botometer and Face++ API. The evaluation is carried out using a 10-fold cross validation which returns an accuracy of 0.74 ± 0.09 for Random Forest and 0.76 ± 0.00 for CatBoost, while the accuracy of both models when classifying only users that are pure retweeter is 0.71. A second analysis using the confusion matrix and ROC curve is done by splitting the dataset into training (70%) and test (30%) and the results obtained for the two models are depicted in figure 6.16. From these plots we can see how simple statistics related to users prove to be even more useful than news content features alone, since in this baseline the AUROC achieved is greater then the one obtained by the same model in case of article classification.

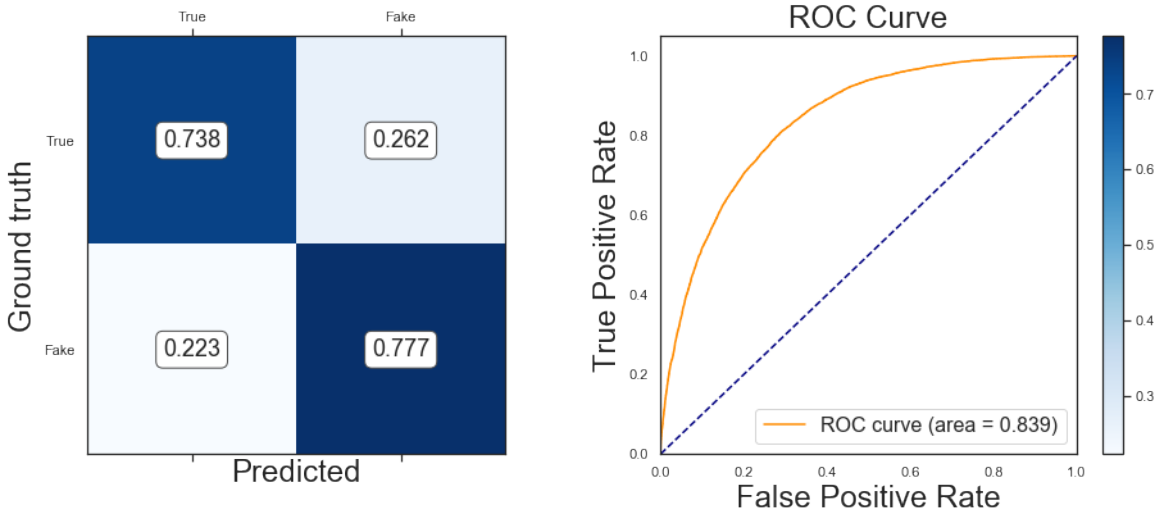
User tweet features

Now that a baseline is defined we can enrich the features by extending the analysis to the tweets that a user has sent. These new features focus on two different aspects, namely the type of content that the user shares and the sentiment that the user expresses towards that content. The content can be obtained as the category, extracted parsing the URL, of the article. Although this information is not available for all articles, this representation still provides useful knowledge about the actual interests of a user i, which can be represented as a vector $v_i = [x_{i1}, \ldots, x_{i8}]$ where x_{ij} is the number of articles of category j that the user i has sent. At the same time, to measure the sentiment we resort to TextBlob, which provides a simple interface to perform sentiment analysis on text. The library uses two indicators, namely polarity and subjectivity. The former score is a float within the range $[-1.0, 1.0]$. The latter is a float within the range $[0.0, 1.0]$ where 0.0 is very objective and 1.0 is very

https://textblob.readthedocs.io/en/dev/



(a)

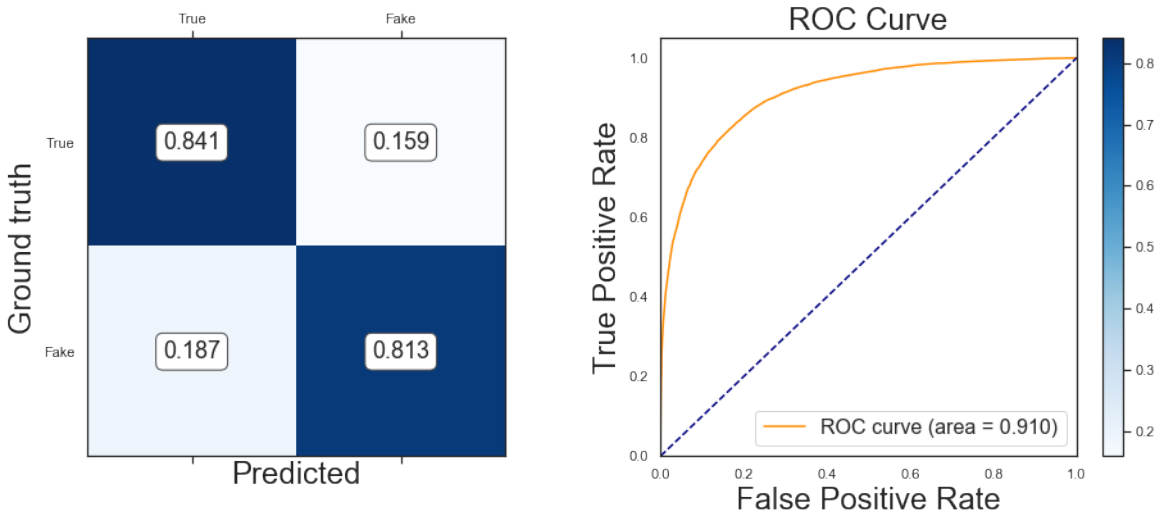


(b)

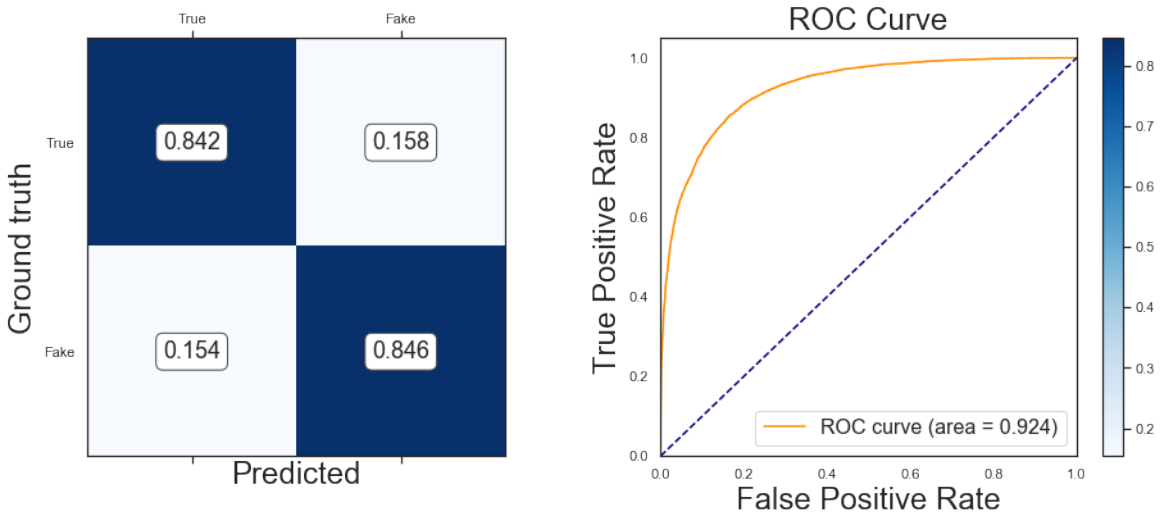
Figure 6.16 Confusion Matrix and ROC curve plots of Random Forest **(a)** and CatBoost **(b)** trained using only user features

subjective. We can therefore define and average polarity and subjectivity score for a user analyzing the text that is associated with each tweet. As before, the confusion matrix and ROC curve for the two models are illustrated in figure 6.17.

In this case, the 10-fold cross validation returns an accuracy of 0.83 ± 0.07 for Random Forest and 0.85 ± 0.01 for CatBoost, while the accuracy of Random Forest and CatBoost on pure retweeter increases to 0.81 and 0.84, respectively. From this results we can observe an increase in accuracy of more than 10% for both models over the previously defined baseline.



(a)



(b)

Figure 6.17 Confusion Matrix and ROC curve plots obtained from Random Forest **(a)** and CatBoost **(b)** using enriched content and sentiment features

These results evidence how the combination of news content and social engagement give rise to many latent information that news content alone is not able to capture.

Articles Topic

From previous results we have concluded that the article's category provide discriminative information for the task at hand. For this reason, we use statistical modeling techniques to further enhance the set of features and extract additional content information from all articles.

Topic modeling is the process of identifying topics in a set of documents, it provides a different approach to extract numerical features from textual content. For our analysis, we use Latent Dirichlet Allocation (LDA), which is a form of unsupervised learning that views documents as bags of words, and represents them as a mixture of topics, which in turn are represented as a mixture of words. This representation can be then used to represent the content of an article and can be combined with other features.

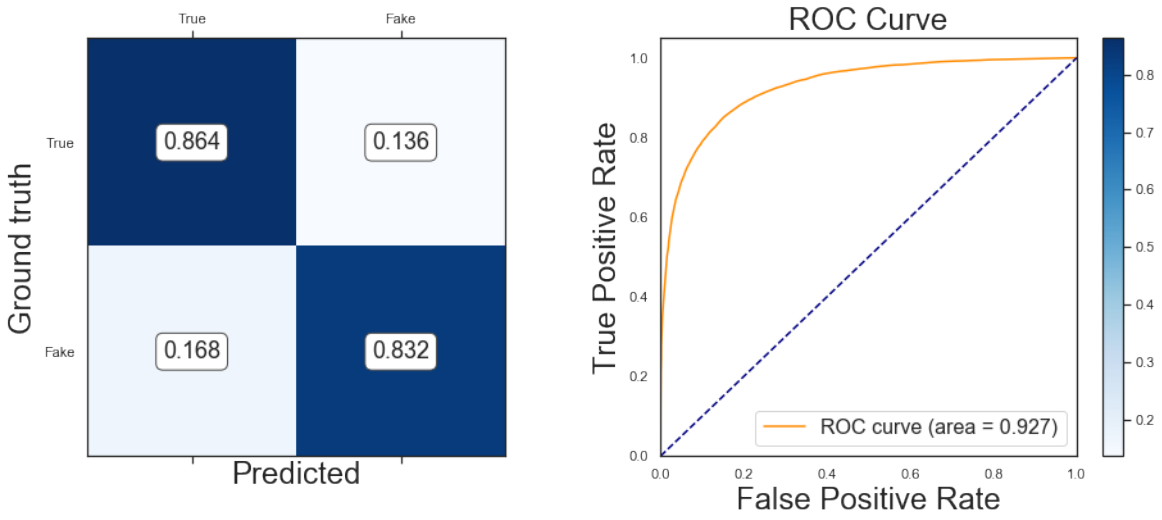
In order to extract the topics from the documents we first perform pre-processing on them, this step consists in stopwords removal, bigram identification, e.g. *White House* and *Donald Trump*, and lemmatization, keeping only words that are associated with a POS tag that is either noun, adjective, adverb or verb. Since LDA is a parametric topic modeling method, we have to define a priori the number of topics k to be extracted, in our case $k = 50$. In table 6.10 we show an example of three topics, along with their top 5 associated keywords, obtained after training the model for 20 passes.

Topic ID	Top-5 keywords
0	(0.021, border) (0.015, trump) (0.014, immigration) (0.012, migrant) (0.012, wall)
22	(0.015, uk) (0.015, government) (0.011, brexit) (0.010, deal) (0.010, britain)
41	(0.045, climate_change) (0.034, ocasio_cortez) (0.025, deal) (0.025, climate) (0.012, green)

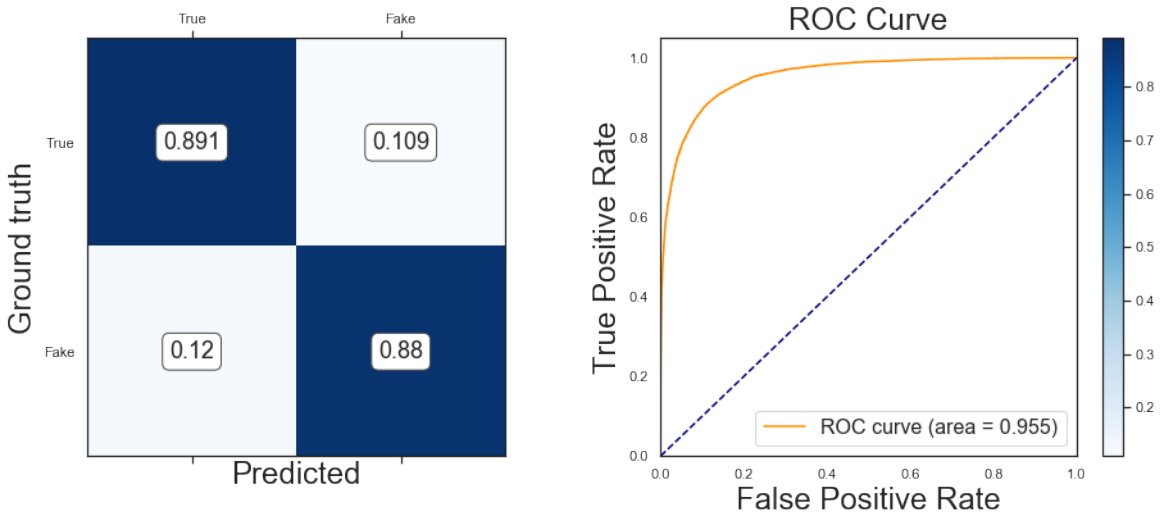
Table 6.10 Representation of three topics with their top-5 associated keywords and weights

With this news extend set of features we repeat the training and evaluation procedure and obtain the confusion matrix and ROC curve visible in figure 6.18. In particular, we notice that the accuracies measured with 10-fold cross validation increase again to 0.85 ± 0.07, for Random Forest, and 0.89 ± 0.00, for CatBoost. These results further confirm the idea that knowing the interest of a user, in terms of topics shared, can give powerful insights on its reliability as a content sharer.

For what concerns users whose behavior is only to retweet someone else content, the measured accuracy for CatBoost is still 0.89 although, as it visible in figure 6.19, there is a noticeable difference in performance when classifying users, and a possible explanation



(a)



(b)

Figure 6.18 Confusion Matrix and ROC curve plots obtained from Random Forest **(a)** and CatBoost **(b)** using LDA to extract 50 topics and enrich content features

for this may be due to the fact that the two classes are unbalanced, nonetheless the general accuracy is reasonable.

To understand how the different features contribute to the classification task we decide to perform the permutation importance analysis on the last version of the CatBoost model. As it is reported in table 6.11 the most important features to discriminate a reliable user from an unreliable one are a mix of content and social features that are either provided by Twitter API, articles URLs or the LDA model. As it could be easily expected, the most discriminative

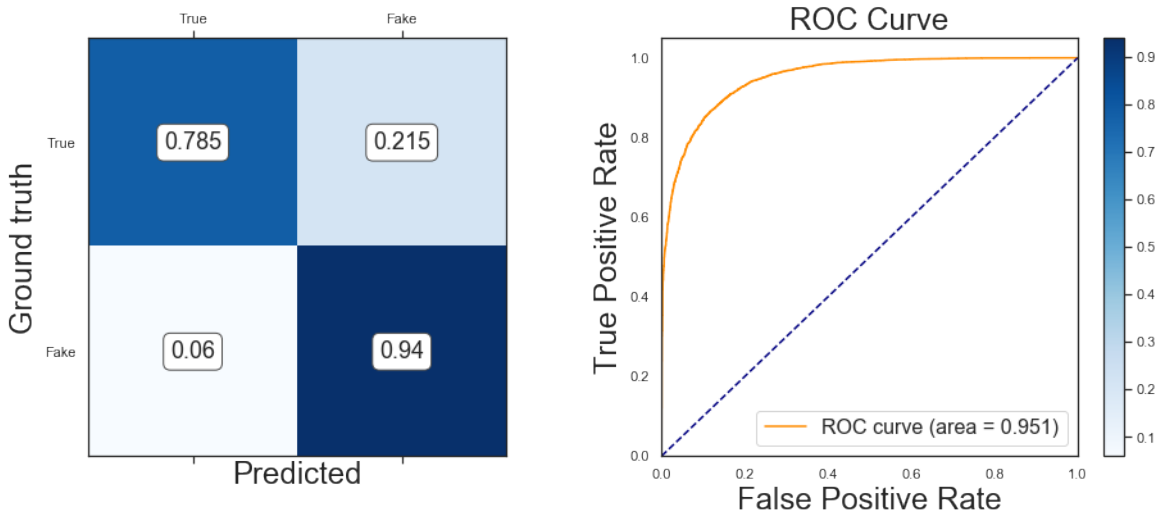


Figure 6.19 Confusion Matrix and ROC curve plots obtained from CatBoost when evaluating users that only share other people content

feature to distinguish between user's type is whether a user tweets mostly about politics or not and eventually how many of those tweets are not original ones.

Weight	Feature name
0.0579 ± 0.0020	cat_politics
0.0239 ± 0.0016	num_retweets
0.0215 ± 0.0028	cat_science/technology/health
0.0203 ± 0.0019	followers_count
0.0186 ± 0.0005	cat_business
0.0181 ± 0.0016	cat_national/local
0.0146 ± 0.0008	topic_25 (Trump)
0.0135 ± 0.0016	cat_world
0.0125 ± 0.0020	topic_3 (Police shootings)
0.0118 ± 0.0022	num_tweets
0.0118 ± 0.0014	topic_24 (Brexit)

Table 6.11 Permutation importance feature weights obtained using CatBoost as candidate model

Looking at the table we find that three of the topics found in the documents are important features for the model, and by considering the terms that define those topics we find that: topic 25 focuses mostly on articles that involve or talk about Trump, topic 3 is mostly related to shootings and kills that involve police forces, while topic 24 focuses on the Brexit deal and campaign. The importance of these topics is consistent with the user distribution, since

most of them are from English speaking countries and in particular from USA and UK, and with the most relevant political topics at the time of writing.

6.5.3 Results

In this section we summarize the results obtained in the two different case studies. We start by highlighting the fact that in both cases the use of auxiliary information introduced with the social context significantly increased performances. In fact, for the article classification task the best result is achieved when combining news content, in the form of Doc2Vec embeddings, with news diffusion using a graph representation, as it is shown in table 6.12. This last result highlights two main keypoints: 1) Social features are extremely useful to expose latent information in the context of news diffusion 2) Graph neural network provide a powerful, yet unexplored, tool for fake news detection.

Features type	Accuracy	AUROC
Doc2Vec text embedding	71%	0.78
Twitter statistics (tweet)	68%	0.73
Graph representation + LDP	67%	0.72
Graph representation + Doc2Vec text embedding	**86.6%**	**0.93**

Table 6.12 Performance summary for the different feature-based configurations evaluated in case of articles classification

For what concerns user reliability estimation, as it is shown in table 6.13, we found that Twitter statistics, in combination with user stance and topic preferences, provide highly discriminative features. In particular, knowing the type of news that a user shares, either the category or the topic, along with the size of its social networks, i.e. number of friends and followers, proves to be an important information to predict its reliability.

Features type	Accuracy	AUROC
Twitter statistics (user profile)	76%	0.84
Twitter statistics + Category vector + Sentiment	85%	0.92
Twitter statistics + Category vector + Sentiment + LDA	**89%**	**0.96**

Table 6.13 Performance summary for the different feature-based configurations evaluated in case of user reliability estimation

Chapter 7

Conclusion

The main objective of this thesis was to investigate the diffusion of fake news on Twitter, and to highlight the importance of the latent information carried by the different set of features, which were extracted from both the news content and the social context using a fully-automatic data collection pipeline. In the end, throughout the exploration of a large set of news related tweets and enriched users profile, we were able to identify the most discriminative features for fake news classification. In addition, we also present a promising formulation using graph neural network that frames the problem of fake news detection as a node classification one.

7.1 Discussion

As social media increasingly become the prime source of information in everyday life, the need of automatic systems that can help human fact checkers in discovering and debunking news content is crucial. In this work we found that, similar to [37], falsehood diffused significantly farther and more broadly than the truth in all categories of information.

In particular, this behavior is more prominent in topics such as politics rather than science, technology or health. Moreover, the users who spread false news have not only a tendency to retweet others people status, rather than writing their own, but they also have significant less followers and friends.

In addition, we argue that the method we propose to identify deceptive content, using state of the art graph neural network, can stand as a powerful tool in this sense, as it is shown from the AUROC of 0.92, achieved using an, easy to get, combination of features related to news proliferation and articles content. Our method can also be extended and improved in several ways, but the results we obtained, especially compared to similar works, are encouraging and suggest that this could be a significant starting point.

Furthermore, despite our tests were focused on a single social platform, i.e. Twitter, the approach could be extended to any other social media, with the proper modifications, that has the following characteristics: 1) the users share news related contents on the social platform and 2) users establish social relations and communicate with each other.

7.2 Future work

If we consider this work as a starting point, there are many directions in which it could be extended.

First of all, we chose to focus on English speaking news sources, but to prove the generality of this approach other studies should be conducted on other languages, for example in Italian, in order to find consistency across the importance of different features and the category distribution of fake news.

Another important option could be to further extend the set of features, by adding the analysis of not textual contents (such as the emoticons and images) or even sentiment analysis, which could give a significative contribute to understand whether a user is sharing a news by supporting or denying its content.

Finally, an additional improvement in the performances of graph neural network could be obtained by extending the graph representation with the introduction of edge features that describe the type of users that share similar articles. In doing so, graph convolution operation would also consider edge related features alongside nodes ones.

Bibliography

[1] Allcott, H. and Gentzkow, M. (2017). Social media and fake news in the 2016 election. *Journal of economic perspectives*, 31(2):211–36.

[2] Bakshy, E., Rosenn, I., Marlow, C., and Adamic, L. (2012). The role of social networks in information diffusion. In *Proceedings of the 21st international conference on World Wide Web*, pages 519–528. ACM.

[3] Bandari, R., Asur, S., and Huberman, B. A. (2012). The pulse of news in social media: Forecasting popularity. In *Sixth International AAAI Conference on Weblogs and Social Media*.

[4] Bastos, M. T. and Mercea, D. (2019). The brexit botnet and user-generated hyperpartisan news. *Social Science Computer Review*, 37(1):38–54.

[5] Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

[6] Bovet, A. and Makse, H. A. (2019). Influence of fake news in twitter during the 2016 us presidential election. *Nature communications*, 10(1):7.

[7] Bronstein, M. M., Bruna, J., LeCun, Y., Szlam, A., and Vandergheynst, P. (2017). Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42.

[8] Cai, C. and Wang, Y. (2018). A simple yet effective baseline for non-attribute graph classification. *arXiv preprint arXiv:1811.03508*.

[9] 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. ACM.

[10] Chakraborty, A., Paranjape, B., Kakarla, S., and Ganguly, N. (2016). Stop clickbait: Detecting and preventing clickbaits in online news media. In *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 9–16. IEEE.

[11] Colleoni, E., Rozza, A., and Arvidsson, A. (2014). Echo chamber or public sphere? predicting political orientation and measuring political homophily in twitter using big data. *Journal of communication*, 64(2):317–332.

[12] Flaxman, S., Goel, S., and Rao, J. M. (2016). Filter bubbles, echo chambers, and online news consumption. *Public opinion quarterly*, 80(S1):298–320.

[13] Garimella, K., De Francisci Morales, G., Gionis, A., and Mathioudakis, M. (2018). Political discourse on social media: Echo chambers, gatekeepers, and the price of bipartisanship. In *Proceedings of the 2018 World Wide Web Conference*, pages 913–922. International World Wide Web Conferences Steering Committee.

[14] Grinberg, N., Joseph, K., Friedland, L., Swire-Thompson, B., and Lazer, D. (2019). Fake news on twitter during the 2016 us presidential election. *Science*, 363(6425):374–378.

[15] Ha, L. and James, E. L. (1998). Interactivity reexamined: A baseline analysis of early business web sites. *Journal of broadcasting & electronic media*, 42(4):457–474.

[16] Hand, D. J. and Till, R. J. (2001). A simple generalisation of the area under the roc curve for multiple class classification problems. *Machine learning*, 45(2):171–186.

[17] Hermida, A., Fletcher, F., Korell, D., and Logan, D. (2012). Share, like, recommend: Decoding the social media news consumer. *Journalism studies*, 13(5-6):815–824.

[18] Jin, Z., Cao, J., Zhang, Y., and Luo, J. (2016). News verification by exploiting conflicting social viewpoints in microblogs. In *Thirtieth AAAI Conference on Artificial Intelligence*.

[19] Kaplan, A. M. and Haenlein, M. (2010). Users of the world, unite! the challenges and opportunities of social media. *Business horizons*, 53(1):59–68.

[20] Kipf, T. N. and Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

[21] Lee, C. S. and Ma, L. (2012). News sharing in social media: The effect of gratifications and prior experience. *Computers in human behavior*, 28(2):331–339.

[22] Liaw, A., Wiener, M., et al. (2002). Classification and regression by randomforest. *R news*, 2(3):18–22.

[23] Livingstone, S. (2005). On the relation between audiences and publics.

[24] Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

[25] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

[26] Monti, F., Frasca, F., Eynard, D., Mannion, D., and Bronstein, M. M. (2019). Fake news detection on social media using geometric deep learning. *arXiv preprint arXiv:1902.06673*.

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

[28] Pariser, E. (2011). *The filter bubble: What the Internet is hiding from you*. Penguin UK.

[29] Persily, N. (2017). The 2016 us election: Can democracy survive the internet? *Journal of democracy*, 28(2):63–76.

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

[31] Shu, K., Sliva, A., Wang, S., Tang, J., and Liu, H. (2017). Fake news detection on social media: A data mining perspective. *ACM SIGKDD Explorations Newsletter*, 19(1):22–36.

[32] Shu, K., Wang, S., and Liu, H. (2019). Beyond news contents: The role of social context for fake news detection. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 312–320. ACM.

[33] Song, Y.-Y. and Ying, L. (2015). Decision tree methods: applications for classification and prediction. *Shanghai archives of psychiatry*, 27(2):130.

[34] Susarla, A., Oh, J.-H., and Tan, Y. (2012). Social networks and the diffusion of user-generated content: Evidence from youtube. *Information Systems Research*, 23(1):23–41.

[35] Tang, J., Chang, Y., and Liu, H. (2014). Mining social media with social theories: a survey. *ACM Sigkdd Explorations Newsletter*, 15(2):20–29.

[36] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., and Bengio, Y. (2017). Graph attention networks. *arXiv preprint arXiv:1710.10903*.

[37] Vosoughi, S., Roy, D., and Aral, S. (2018). The spread of true and false news online. *Science*, 359(6380):1146–1151.

[38] Ward, A., Ross, L., Reed, E., Turiel, E., and Brown, T. (1997). Naive realism in everyday life: Implications for social conflict and misunderstanding. *Values and knowledge*, pages 103–135.

[39] Wu, L., Li, J., Hu, X., and Liu, H. (2017). Gleaning wisdom from the past: Early detection of emerging rumors in social media. In *Proceedings of the 2017 SIAM International Conference on Data Mining*, pages 99–107. SIAM.

[40] Zhang, H. (2004). The optimality of naive bayes. *AA*, 1(2):3.

[41] Zubiaga, A., Aker, A., Bontcheva, K., Liakata, M., and Procter, R. (2018). Detection and resolution of rumours in social media: A survey. *ACM Computing Surveys (CSUR)*, 51(2):32.