Dr.Watson

Project 5 - Overcoming the «fake news»

Digital Business Innovation Lab - A.Y. 2017/18
Group: *GestIT - massimo.perini@asp-poli.it*

Demo: https://fakenews-polimi.herokuapp.com/

Alessio Baccelli 898514 *Gioele Bigini* 898741 *Filippo Calzavara* 898526

Luca Comoretto 906086 *Harilal Orunkara Poyil* 898231 *Massimo Perini* 898302 Hichame Yessou 898275

Contents

Fake News Analysis

1.1 Introduction

Nowadays people retrieve pieces of information from several sources: websites, newspapers, but also social media. The purpose of this project is to explain whether the latter is a risky policy. We'll soon see why it is and we will propose an approach to tackle the issue.

Social media sites like Facebook and Twitter have become predominant tools for online users to share contents that go from simple text to rich media. The rate at which stories are getting propagated through social media is enormous. For example, on an average, around 6000 tweets are posted on Twitter per second which corresponds to 500 million tweets per day [Sta] and this number grows in an incremental fashion day by day. In recent years social media evolved as significant tool for opinion formation. It's often noticed that many incidents and stories appears in social media well before they hit on mainstream media. For example, they played a major role in the anti-government demonstrations in Turkey [Hut13] to US presidential election, 2016 [Kap16]. It helps in humanitarian and disaster relief efforts and other law and order situations to a good extent. It also employs key role in the field of marketing, healthcare, politics, entertainment, etc. At the same time, the misuse of social media also gets observed frequently since several individuals are using it for promoting controversial, if not simply populist and false topics, such as anti-nationalism, human trafficking or terrorism.

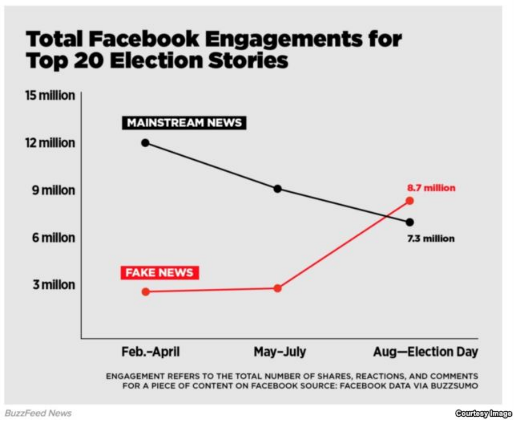


Figure 1.1: Facebook Engagements over News

To tackle the problem we first need a definition of it: fake news are false stories that appear to be news, spread on the Internet or using other media, usually created to influence political views or as a joke.

More extensively, we can define them as fake news when an inaccurate, sometimes sensationalist report is created to gain attention, mislead, deceive or damage a reputation. Unlike misinformation, which is inaccurate because a reporter has confused facts, fake news is created with the intent to manipulate someone or something. Fake news can spread quickly when it provides disinformation that is aligned with the audience's point of view because such content is not likely to be questioned or discounted.

We can certainly say that social networks are very suitable to spread news seen the features of the formers: they are free and they rapidly spread information around the world, much faster than more traditional means of communication. Furthermore, they are not checked and balanced by any reliable authority. People are therefore passively fed with uncertified news which might be of several types: poorly written and therefore confusing, striking and easy to contradict or misleading and made to induce them to subtly believe in something that actually hasn't happened.

This extensive spread of false reports has a deeply negative impact on the society and even single individuals, who can be led to overreact impulsively, determining dramatical and dangerous situations. There are, for example, reported episodes indeed of people shooting somebody else due to fake news Considering all these factors, a fake news detector looking after social media has recently become an important tool to distinguish and guide people in the forest of the lies. Several research groups and companies are nowadays addressing this issue investing considerable resources in order to deliver an automatized software to the market.

The challenges are multiple due to the unique features that the problem presents and which make traditional algorithms rather ineffective. On one hand, the posts are written to mislead readers, recalling articles patterns so that content analysis results difficult: to address this issue is necessary to analyze further information on the person who published the news such as his relations on social media. On the other hand, the usage of this auxiliary information is not trivial at all, since they are huge and noisy: determining whether they are useful or not becomes very tricky. Additional issues are related to the human behavior: when you listen to somebody talking it's possible to understand, most of the times, the use of sarcasm or irony, in general, the tone of the speech. When it comes to reading something, all these information are lost. Wrapping up, our software has the goal to face all these issues and solve them using an ensemble of Data Mining techniques, capable of overcoming the problems.

1.2 Management Analysis

Initially, companies had the power to control all the informations belonging to the general public through their communication channels. With the advent of social media the old communication changed, forcing companies to look at the power of the internet, creating adequate company structures to reach the final consumer.

For example, without the power to govern informations, people began to be more informed about products, avoiding the high bias introduced by companies through their informations. This brought to the birth of review platforms, which escape even more from companies and mean that people are more able to make the right choice by looking not only to their personal observations but also to unbiased opinions on the web (instead of listening to companies' advertising).

As previously said, to overcome the problem companies began to create adequate company structures with the following goal: building their own image. In order to do this, some of them thought to use review platforms as showcases with the final goal to be easily in contact with consumers.

But the direct contact with consumers is not always that good from a company point of view. From the companies point of view, the final goal is to acquire new satisfied consumers. Allowing people to give a negative feedback on a product could bring to a promising product to not be sold and turn away potential new consumers. Even if this behavior should be normal in general, the problem resides on feedback spread. Since an opinion is highly biased by who pronounces it, It may not reflect reality and could affect also the opinions of others. On the internet this is highly amplified.

Finally, we can relate these cases as "Fake News". The Fake News are very difficult to control and they damage the company in a critical way. For this reason the image of a company must be strong to face them in the best way.

1.2.1 From Fake Reviews to Fake News

Opinions like online reviews are one of the main sources of information for customers to help gain informations into the products they would like to buy. Customers write reviews to give feedback by sharing their bad or good experience with others. Their experiences, of course, impact long term businesses either positively or negatively. Of course, this fact generate new opportunities for manipulating customers' decisions thanks to the generation of fake reviews. This practice is called opinion spamming and people who write false opinions to influence others are called spammers. Opinions/Reviews may be written either to improve or damage the reputation of a business or a product. Recently we noticed that opinion spam does not exist in product reviews and customer feedback only: fake news and misleading articles is just another type of opinion spam. One of the biggest sources of spreading fake news is, of course, social media websites, like Facebook, Twitter, Sina Weibo and so on. Recognizing fake news is a complex task and probably much harder than detecting fake product reviews. The open nature of the web and social media and the recent advance in computer technologies simplified the process of Fake News generation and spread. While it is easier to understand intention and impact of fake reviews, the intention and the impact of propaganda generation by the spread fake news can't be easily understood. Fake reviews and Fake news, however, may be reputational risks for a company.

1.2.2 Reputational risks

Reputational risk issue has always attracted much attention from academics and practitioners, since reputation is usually considered a critical asset for a company. It is able to influence the behavior of the company's stakeholders in several ways. A "good" reputation draws staff attention, helps the company in retaining customers, allows to legitimate company operations with public authorities and encourages the shareholders to invest in a certain company. Any damage to corporate reputation, instead, could have very bad consequences, such as loss of current or future customers, loss of employees or managers within the organization, reduction in number of business partners and increase in financial funding cost. Because of this, reputation has to be managed as other company's assets and companies have the need of dealing with the problem of reputational risk. In recent years, indeed, the issue of the reputational risk management moved high on top managers' agenda. A survey done by Economist Intelligence Unit with 269 senior executives confirms what we stated, showing that a bad reputation is considered one of the most significant threat to business success [The05]. In general, reputation can be defined as the system of stakeholders' perceptions and expectations towards the corporation, and reputational risk can be described as the risk of having this system of perceptions/expectations altered or damaged. From this perspective, the ability of a company of handling a crisis depends not only on its decision, but also on how its actions are perceived by its stakeholders. Recent years growing sensitivity of a broader range of stakeholders, including the public opinion, towards corporate behavior, is challenging even more the companies.

1.2.3 Patents research

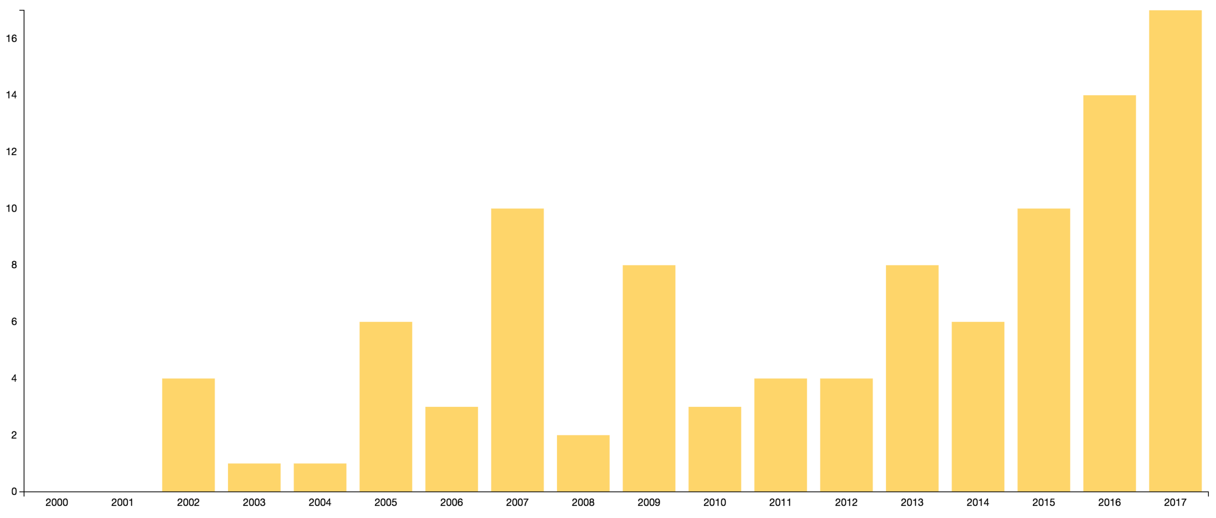
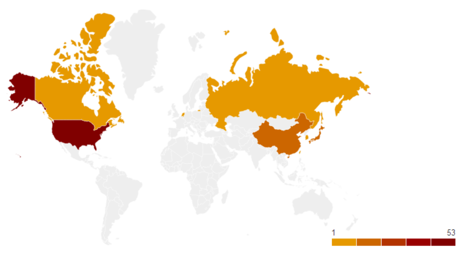
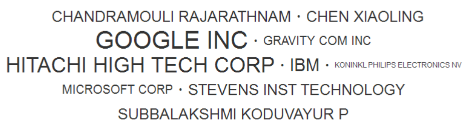


Figure 1.2: Patents related to fake text detection systems in different years



(a) Patents applicants by country


CHANDRAMOULI RAJARATHNAM · CHEN XIAOLING
GOOGLE INC · GRAVITY COM INC
HITACHI HIGH TECH CORP · IBM · KONINKL PHILIPS ELECTRONICS NV
MICROSOFT CORP · STEVENS INST TECHNOLOGY
SUBBALAKSHMI KODUVAYUR P

(b) Top 10 applicants

Figure 1.3: Patent applicants

We investigated patents assignments trends in relation to fake or noisy text detection systems. Even if patents are just one of the ways a company can protect an IP, we can have an idea of their actions in this area. As Fig.1.2 shows, this subject is quite new since there were a very small number of patents in the first years of this century. This topic start becoming slightly more popular in the years 2007-2009 and its popularity grown even more in the last years. This can probably be caused by the spread of Internet around the world and events like Brexit and US Elections, in which Fake News played a key role and big-tech giants promised to deal with this problem. Fig.1.3a shows the countries most active in this topics: they are U.S. and China since the biggest IT industries like Google or Tencent are located in these countries. Fig.1.3b shows the companies with the highest number of patents: Google is the leader in this field, followed by other companies like IBM and Microsoft. However, since the number of patents released in this topic is very low, companies that leads this rank probably don't have a strong advantage towards their competitors for what concerns this topic. It's interesting the fact that famous companies like Facebook or Twitter don't appear in this list: probably most of the IP of these corporations in this field are protected in other ways or are not available on Google Patents.

1.3 Examples

Deceiving the enemies, misleading the population or simply lying to somebody are always been part of the history of the world, with the purpose of accomplishing some selfish goals.

In this section you will find a couple of important examples of the last two decades.

1.3.1 Autism and vaccines

It's about 20 years that the most famous health fake news blew up, bringing disconcert and confusion in the world: on the prestigious magazine 'The Lancet', Andrew Wakefield published an article claiming the correlation between trivalent vaccine and diseases insurgence. Deceiving the referees of the magazine, he succeeded in publishing something deliberately false in order to exploit his own monovalent patent [Adn18]. The ideas spread since then are still haunting the public opinion, supported by populist parties and through internet, such that there are incredibly low vaccine rates in the population. The harm caused by a single news can be evidently huge and very difficult to contrast: decades of efforts have been nullified by few years of ignorance.

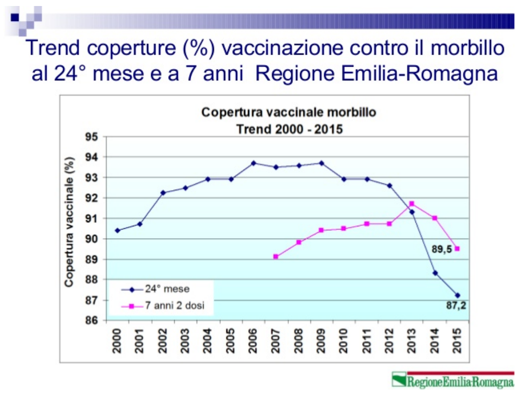


Figure 1.4: Measles vaccine rate

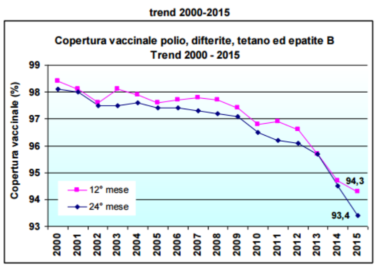


Figure 1.5: Other vaccines rates

1.3.2 US Election 2016

Much of today's political debate starts on social media, and the medium often amplifies vitriol and slants information. Social media have been used since their creation for political purposes, but the 2016 presidential election is the first one which has been tremendously influenced by their use and especially their misuse. Clinton and Trump both used social to their advantages, but overall the surrogates, supporters and influencers were the ones who shaped perceptions, according to whatever views served their preferred candidate's interests.

Social networks democratize information spreading and sharing, and it helps individuals with very limited interests rapidly and easily connect with others who share their thoughts. Sensational headlines and memorable memes give rapid-fire politics momentum on social media. The fact is that what is mainly shared is also the content which is the most polarized, most interesting, entertaining and engaging, much more than the stuff normally posted, therefore news are narrowed and we enter in a vicious circle. What happened during the electoral period (they tend to be also in general) was that the most sensational news were actually the fake ones, which were posted and re-tweeted multiple times with breaking effects.

WikiLeaks CONFIRMS Hillary Sold Weapons to ISIS... Then Drops Another BOMBSHELL!
From The Web

Figure 1.6: Fake news example

1.4 Our solution

The solution we elaborated to solve the problem of discovering the fake news is articulated in three classifiers put together in an ensemble, a machine learning tool that uses the information given by its components and combines them to obtain a rather better result.

In particular, we developed a multilayer recurrent neural network that exploits the timely disposition of the interactions with the tweets, an IBM Watson based Natural Language Understanding analyzer and a FastText developed text classifier.

To be able to interface the complex system, a dynamic and interactive website was developed and published on the IBM Cloud App Service. A friendly interface classifies selected random tweets and shows precise statistics about them. In the following chapters the classifiers and the website will be explained in technical details nevertheless a brief idea of how they work will always be given. A simple analysis of the external environment and a thorough brand identity explanation have been conducted too.

Brand Identity

2.1 Brand Identity and Design

An important aspect of the project development concerned the design of a Brand Identity that characterized in a unique fashion the final product delivered. The inspirational colours and shapes of the whole front-end and presentation stack resembles the undeniable suggestion of confidence that IBM first-rate services offer. The choice to include a Brand Identity chapter in this document stems from the fact that, due to the high complexity of the evaluation algorithm, a good visualization mask needed to be developed in order to convey the robustness and sophistication that the technical part presents while enabling not experts of the Machine Learning field to understand the potential of this tool.

Certainly, a future possible development is production (thus public distribution) of the platform as an example of combined AI technologies applied to the real life.

2.2 Colors shades

Figure 2.1: Palette of colors used on the front-end/presentation layer

The first choice that the UX unit has taken was to generate a palette of colors on which would be later drawn the template and the presentation layout.

As described before, the original colors were influenced by IBM Corporate Identity and re-adapted for the project's purposes

Color	**Name**	**HTML HEX**	**RGB**
■	Daintree	#022231	2,34,49
■	Green Vogue	#022b48	2,43,72
■	Tarawera	#0c3d5f	12,61,96
■	Chathams Blue	#0e486f	14,72,111
■	Blue Chill	#0d6797	13,103,151

2.3 Brand creation

Inspired by the name of IBM's utilized service "Watson", we appeal to our project the name "Dr.Watson", a fictional character in the Sherlock Holmes stories by Sir Arthur Conan Doyle, while *GestIT* remains the group formal name.

The brand creation has begun with logo's designing. Such process included a thorough research of the main font later found as *Merriweather Black* and the application of the Palette colours.

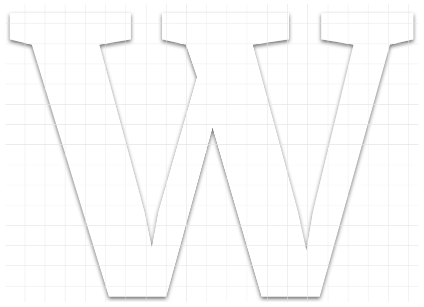


Figure 2.2: First sketch of "W" which stands for Watson of "Dr.Watson"

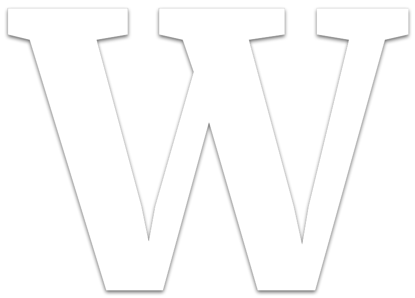


Figure 2.3: Final "W" with transparent background

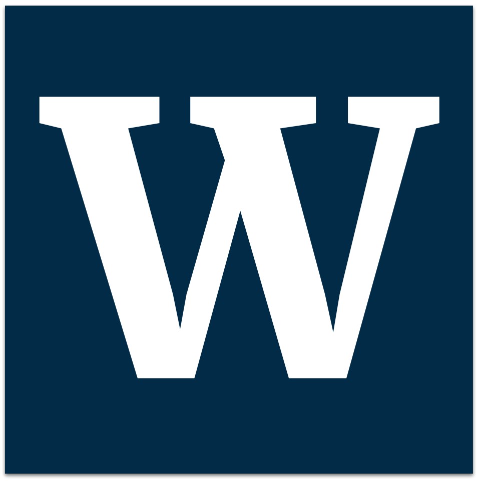


Figure 2.4: Final "W" with dark blue background

Front End

The front-end is the application layer that deals with the presentation of the website. Its goal is to facilitate the end user's interaction with the system, composed of algorithms, through a more concrete and closer to reality interface.

3.1 Functionalities

The web application mainly allows visitors to check whether a news contained in a tweet is fake and visualize statistics about tweets that may or may not contain fake news. To do so, the platform was built to be an easy-to-use application that requires only few steps to assess tweets' news and offers the possibility to have the same user experience across desktop and mobile browsers. The application is able to: classify a sample of test data that the user can choose, getting a confidence level of the prediction and getting the sentiment detected by IBM Watson. For transparency, we added some sections that will allow the reader to have a general overview of how Dr.Watson works and the features implemented for the project.

We here split the web application into six main sections.

3.2 Website Organization

Dr.Watson was designed on a single web page in order to require the lowest possible number of steps from the user and avoid mazes that could distort the user experience on the web application.

3.2.1 Sidebar

From the left side a sidebar is available. Here it is possible to shift quickly to the preferred section available. The six horizontal lines below the section names are lighted up when selected. If no selection is made, the current section is lighted up by default.

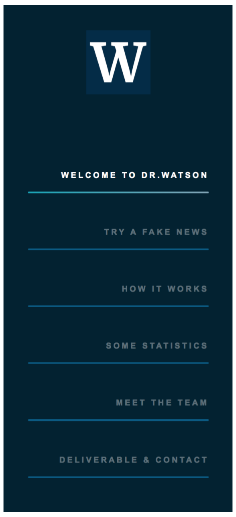


Figure 3.1: The sidebar of Dr. Watson

3.2.2 Sections

Landing Page: This is the main section when the user open the web page. From this section the user will be scrolled down to the second section, where he/she is allowed to try the demo console.

Try Fake News This section allows checking Fake News by simply clicking on the sample shown. This will check the veracity of the tweet and will show the outcome of the analysis carried out by Dr.Watson. In order to keep a high simplicity level across different devices, we chose to show only three tweets per time, while allowing the user to shuffle across the demo dataset.

How it works Hereby we are showing a high-level overview of how the system is structured. As stated on the website, Dr.Watson is a complex application that was built with several programming languages. The front-end interface is developed in Node.js that is a JavaScript runtime application server. The web-scraper used to retrieve, build and enrich the dataset has been written in Java interfacing with the Twitter APIs. The Machine Learning model was written in Python in order to exploit libraries like Pandas where we the group excels.

Statistics Visualization is a big part of the Data Science field, allowing the developer and the user to have a look at mistakes, biases and other unexpected problems related to the data. The growing availability of data has led to an increasing reliance on the data visualization providing a powerful way to communicate data-driven findings, motivating analyses and detecting flaws. In our case, we have a massive amount of data so we needed a simple way to explain the distribution and its analysis. We opted for 4 graphs, 3 of which describes the best the tweets, news and users in our dataset. The sentiment graph instead shows the distribution of the feelings contained in the tweets. All the charts are interactive, so that the visitor can click and move through the timeline and the sentiments.

Meet the team This section describe the whole team and development organization that we are glad to quote again:

- Alessio Baccelli - Artificial Intelligence development
- Gioele Bigini - Front-end and Data Visualization
- Filippo Calzavara - User Experience and Front-end development
- Luca Comoretto - Artificial Intelligence developer
- Harilal Orunkara Poyil - Artificial Intelligence and Domain Expert
- Massimo Perini - Artificial Intelligence development
- Hichame Yessou - Front-end and Data Visualization

Deliverable & Contact From this section the user is able to download the document, view the mid check presentation, go to the GitHub Repository (permissions may be needed) or contact the team.

3.3 Development

3.3.1 Strategy

The chosen development strategy for Front-end development is the Top-Down strategy. The Top-Down approach generally involves identifying the final product and then solving the sub-problems deriving from it, in a divide-et-impera style.
Down below the followed procedure for the project development is summarized:

1. Generation of the HTML structure starting from an open-source template
2. Welcome page and footer development
3. Meet the team section development
4. Contact section development
5. Test section development
6. Statistics section development

The design of the web page has been developed concurrently to other tasks.

3.3.2 Programming Languages

The three main technologies used for web development are:

- **HTML**
- **CSS**
- **Javascript**

HTML (Hyper Text Markup Language) is the standard markup language for creating Web pages. It is mainly used to build the general structure of the website.

CSS (Cascading Style Sheets) is a style sheet language used for describing the presentation of a document written in a markup language like HTML. Briefly, it is used to display contents correctly. The nature of HTML does not allow to describe the design of the contents like layouts, colors and fonts. HTML and CSS combination constitutes the User Interface (UI), that part that can improve content accessibility, provide more flexibility and control.

Javascript (JS) enables interactive web pages for the correct interaction between the end user and website. Although HTML provides basic functions for interactions with the end user, Javascript extends the potential of HTML and CSS through animations and interactions in real time. This part constitutes the User Experience (UX) that tries to improve the overall experience of a user using the product.

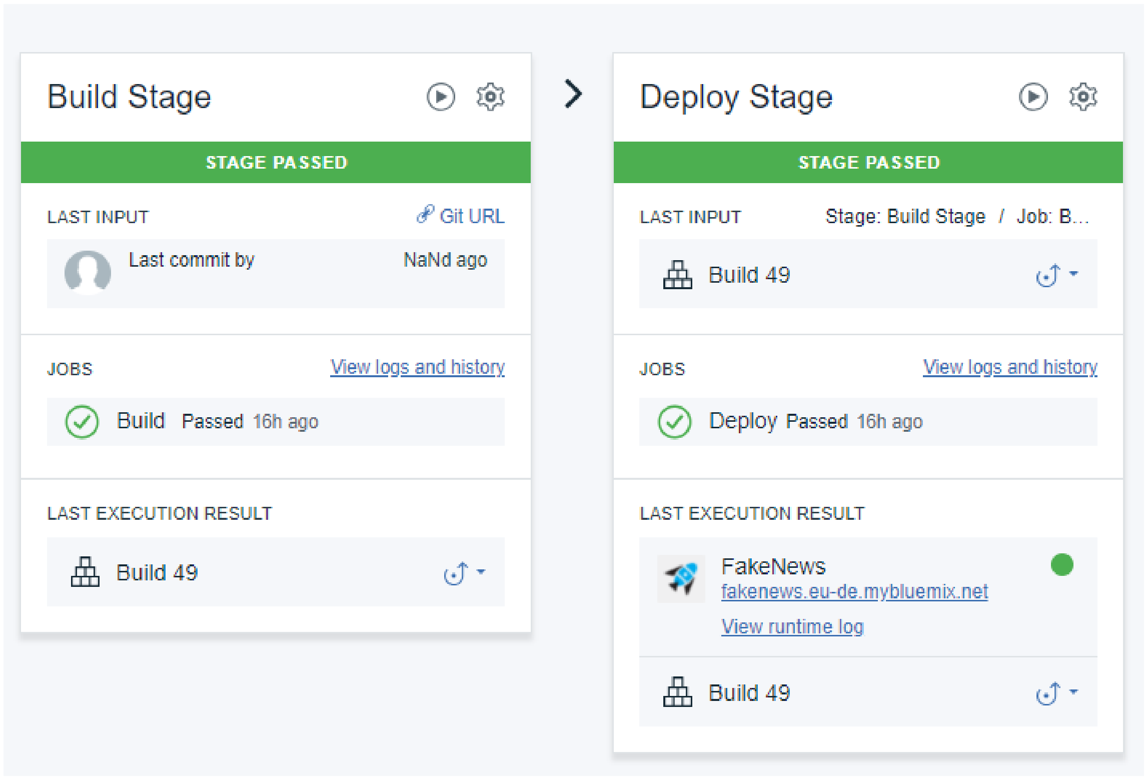
3.3.3 Client-Server Patterns adopted

The application is based on a Client-Server communication model. The clients are thin, in order to let the application run on low-resources devices and in order to easily add new features without always require an update of the application. This approach has been chosen for different reasons:

- Its practical and easy to implement.
- Data synchronization: there is only a central server that keeps and manage the data.
- Having a cluster of servers improves the availability, reliability, safety and mantainability of our system.
- Improves the security between clients, that known only the server endpoint but not other clients.

3.3.4 Automatic Deployment

In projects where several engineers work together is a crucial point to have a centralized codebase and a unique deployment point in order to be coherent and updated with the features integrated. Since structurally the project is developed in two different layers (front-end, back-end) we considered necessary to split the work in different repositories while leaving a link between them. This was done by setting up the main repository to have a submodule towards the front-end project in order to have only commits concerning the web application. The second part has been accomplished by having an automatic deployment system that runs after every commit in the repository. We used the **IBM Delivery Pipeline** in order to trigger the building process and the deployment of an instance of the **Node.js Cloud Foundry** applications. This provides us with the possibility to commit a new functionality and within minutes to have it deployed on the domain of our web application with the advantage of testing it on several devices such as desktops and smartphones, while keeping a high level of integrity.



3.3.5 Tools

Visualizing dataset insights is not that easy in general. A powerful tool to do it has been Highcharts, a Javascript library that easily allow developers to set up interactive charts in a web page. It is fully HTML5 compatible and dynamic, which actually means that is hardly customizable.

3.4 Code structure

Our front-end has been developed with Node.js and Express in order to have a minimal and robust framework that we could have shaped with our needs and integrated with the several APIs which Dr. Watson needs. We chose to use Handlebars as a template engine since we wanted to keep the majority of the business logic away from the representational layer of our web application. Every component and section of Dr. Watson has been build in order to be completely responsive and to be correctly displayed on the majority of the display sizes.

The web application is structured around the landing page, which allows us to have a very simple routing structure, with very few routes used for interactions with Twitter APIs. This was needed in order to allow the integration of multiple features from several engineers at different times. Having a strong and neat separation of concerns have been a key factor in the early decisions regarding the architecture of the web application and the structure of work-flow.

The entry point of our application is the `/bin/www` that initialize the low-level settings like port usage, HTTP server and the various listeners that it needs. Our core component is the app.js which set up the routes and the templating engine of our web application. From here we render all the sections of the landing page through the index route, while the tweet route is left to interact with the Twitter APIs. We mainly have used 2 Twitter APIs which are the `statuses/show/:id` and the `statuses/oembed` that have been used to retrieve the JSON data of the tweets and the HTML attribute in order to embed the Tweets.

3.4.1 Fake News Evaluation

Figure 3.2: The icon indicates that a Fake News has been found

The network speed is an important bottleneck to take into account. Since the computation of the models have been done through the IBM Watson API we opted for an Offline-based computation where few results have been pre-computed and available to be shown to the end-user. To evaluate if a news is Fake or Real the user can easily use the tool available inside the "Try a Fake News" clicking on the preferred tweet. The Offline-computation will start showing the results.

3.4.2 Statistics

We had several technologies to choose from to visualize our data, but we opted for Highcharts because it seemed to be the most responsive and with the highest number of features to be personalized.

The Javascript code of the Highcharts implementation is in the public folder and they are fed with JSON, originating from the same folder, in order to enhance and speed up the loading time.

The JSON files have been created in the pre-processing phase. This phase has been useful to correctly reshape data without incurring in problems such invalid values or outliers. Furthermore, since the whole dataset is too large, the pre-processing phase permitted to focus over those data useful to represent the charts in every specific case.

Four are the visualization created, corresponding to four different statistics:

- Dataset Insights

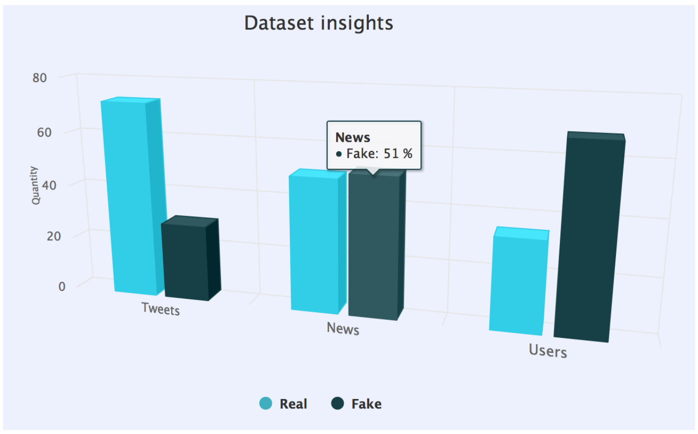


Figure 3.3: The histogram with the main features of a tweet

The chart above shows that even if the analyzed data seems to be unbalanced (in terms of fake/real tweets), the news considered are more or less the same percentage. This means that tweets are usually relating to the same news.

What emerges from the last information related to users (posting news) is that the fake news are more popular and with a wider diffusion than real news. The reason could lie to the fact that those news tries to hit the sentiments of a user, causing him to share it again.

- Distribution of the Tweets over time

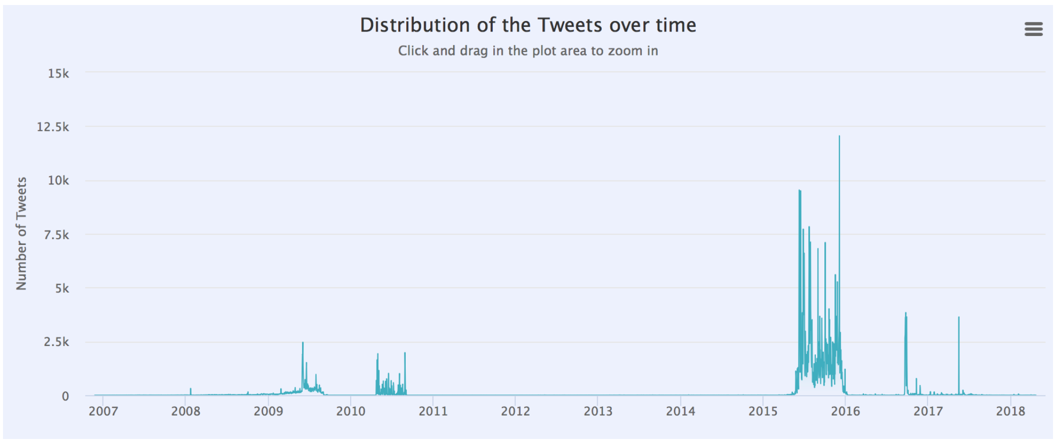


Figure 3.4: The time band considered

The representation shows the time band considered in the project. As we can see, the majority of the tweets comes from 2015 to 2017.

- Tweets Spread over the week:

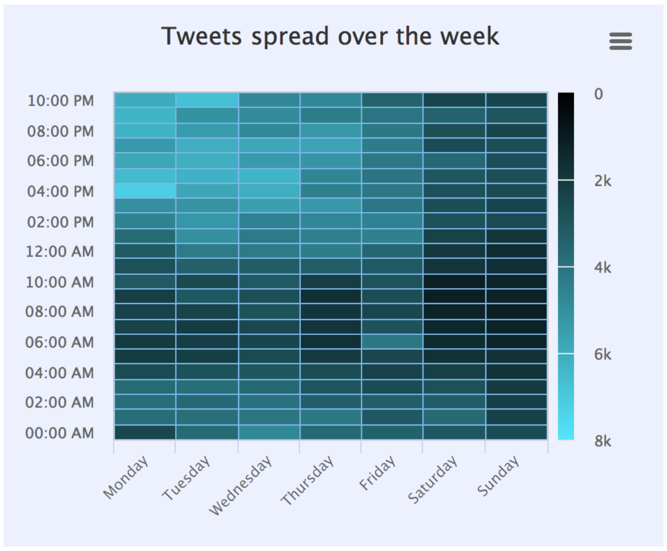


Figure 3.5: The heat map representing news spread

The heat map is really helpful to show 3-Dimension data. The color from "black" to "light-blue" represent the quantity of tweets in that specific period of time. The nearer to black is the color of a cell, the less tweets have been created and obviously the nearer to light-blue is the color of a cell, the more tweets have been created.

- Sentiment Distribution over Tweets:

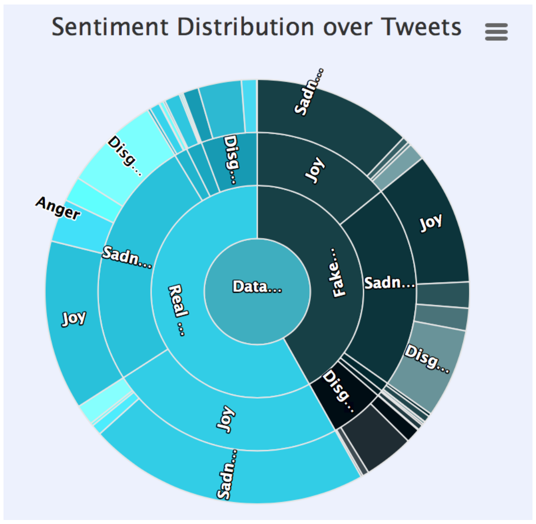


Figure 3.6: The sunburst chart representing sentiment over tweets

Through the Sentimental Analysis has been possible to relate up to 5 dominant sentiments for each news, classified in range from 1 to 5. For creating the chart only the first and the second sentiment for each news has been chosen.

The pie chart shows the relation between sentiments and tweets (that could be real or fake). The last ring is strictly related to the previous one. It represents the second dominant sentiment after the first one in the previous ring.

Fake News Detection using Social Media

4.1 Introduction

Detecting fake news on social media is an important but challenging problem. There are several difficulties related to this task: one of these is that even the human eye cannot accurately distinguish true from false news. For example, a study found that 80% of high school students had a hard time determining whether an article was fake or not [Edk16]. Here, we are gonna propose our solution to tackle the problem.

4.2 Model overview

Our model is based on three different algorithms, which are trained on the same dataset but on different features, obtained through different analysis, explained in the following sections.

4.3 Data collection

For this project, we make use of two real world publicly available datasets: the one used in [MGM+16] paper and [Shu17], which contains the set of tweet ids for different fake news articles. We crawled corresponding tweet contents, using Twitter streaming API [Twi18]. We then extracted the URLs mentioned in each tweets and crawled the text content of the URLs. For this we used Jsoup [Hed10], a Java based library for HTML parsing.

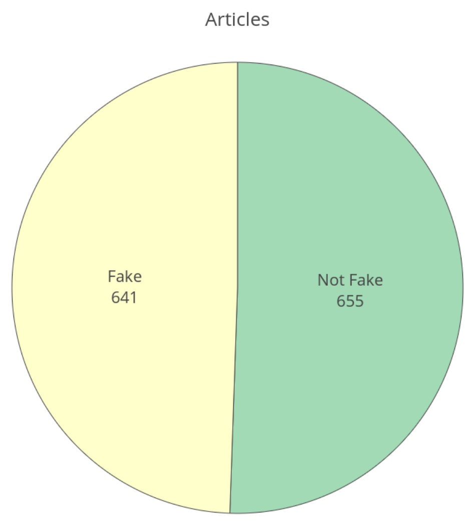
From the response JSONs obtained using crawling phase we filtered the fields used in analysis. The fields we have identified for analysis and the corresponding descriptions are given in 4.1. The overall statistics of the final dataset is given in **??**.

4.3.1 Training data for analyses

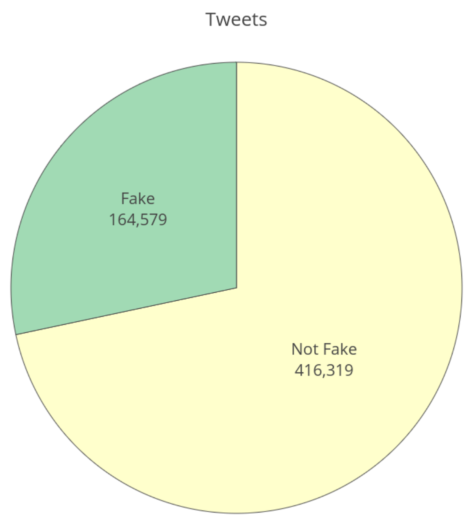
We created 4 different datasets for building analysis models. These datasets are different based on the number of articles (crawled using the links specified in the tweets) per event. The distributions are 5, 10, 15 and 20 articles per event.

Table 4.1: Descriptions of data fields

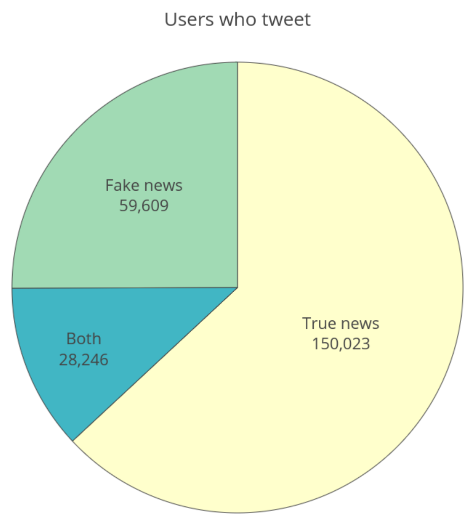
Field	Description
id	The integer representation of the unique identifier for one Tweet
tweet_content	The actual UTF-8 text of the status update
url_content	The content(if available) of any URL mentioned
event_id	The unique identifier for the event(fake news) this tweet belongs to
created_at	UTC time when this Tweet was created. Example: "created_at":"Wed Aug 27 13:08:45 +0000 2008"
user_id	The integer representation of the unique identifier for the the user who posted this Tweet
is_fake	1 and 0 for the fake and real respectively



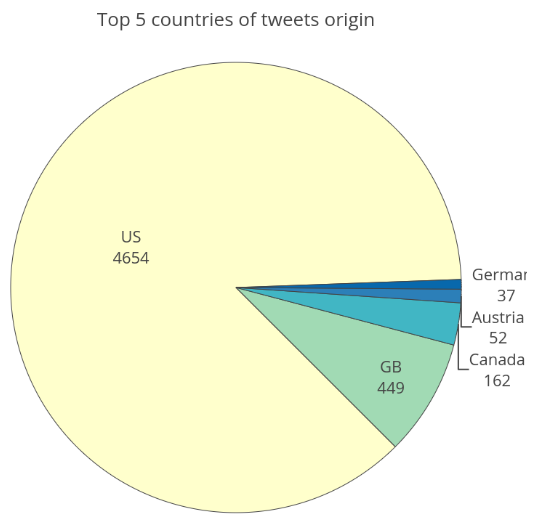
(a) Articles distribution



(b) Tweets distribution



(c) Users for each kind of tweet



(d) Top 5 countries of tweets origin

Figure 4.1: Dataset insights

4.4 Fake News Detection using temporal patterns

4.4.1 Introduction

In this section, a model able to detect fake news using only data related to social media (in our case Twitter) will be outlined. Models that use the article texts will probably be more accurate than this one, but the one here explained relies only on data owned by social network companies.

This model classifies news using the response that a news article generates in the users: how a certain article makes them feel? Efforts to automate response detection typically model the spread of news in the social graph or use social-network dependent features. For example, [TBD+17] shows that the number and frequency of Facebook likes can be a predictor of a post with a fake article. Unfortunately, access to a social graph is not feasible in practice and manual selection of features is labor intensive. Please notice that the model here discussed is very general: it won't be tailored on features that are social-network dependent, such as the number of Facebook likes.

4.4.2 General description

We assume that a tweet is related to an article if this tweet includes the article web address (URL). This means that every article spreading on Twitter will have at least one tweet that includes its URL. The goal of this model is classify this article as Fake or Not Fake according tweets features distribution across time. To do so, our idea is grouping tweets according the article they are related. Then, each group of tweets will be temporally sorted and split in several sub-groups according to the time at which it has been written. For each sub-groups, then, we will compute several features. Given this sequence of features and information about the user, our model will be able to predict if the article is fake or not.

4.4.3 Model description

We consider a series of temporal interactions (in the specific case, an interaction is a Tweet) that occurred between n users with m news-articles over time [1,T]. Each interaction between a user u_i and an article a_j at time t is represented as $e_{ijt} = (u_i, a_j, t)$. This model adopt three main features:

- Textual information: features related to the text of the interaction
- Response of users: features related to frequency and distribution of the interactions
- Source: features that represent the author of the interaction

Response features

In order to fully track the response to an article with high efficiency, we generate different partitions p for each event (a partition is a subdivision of the timeline in disjoint subsets or groups). Each partition groups interactions according to their normalized creation time (the first interaction will be at time 0) and their number is a hyperparameter of this model. For each subset we have several features. The first two are η and Δt. η is the number of interactions in it, Δt is the time between the first interaction of the partition group and the last interaction of the last not-empty group. The usage of Δt allows to remove empty subsets since they don't have anymore useful informations. The partitions are then padded in order to have the same number of subgroups for each article. Of course, Δt of the first subset will be 0 since there are no previous groups.

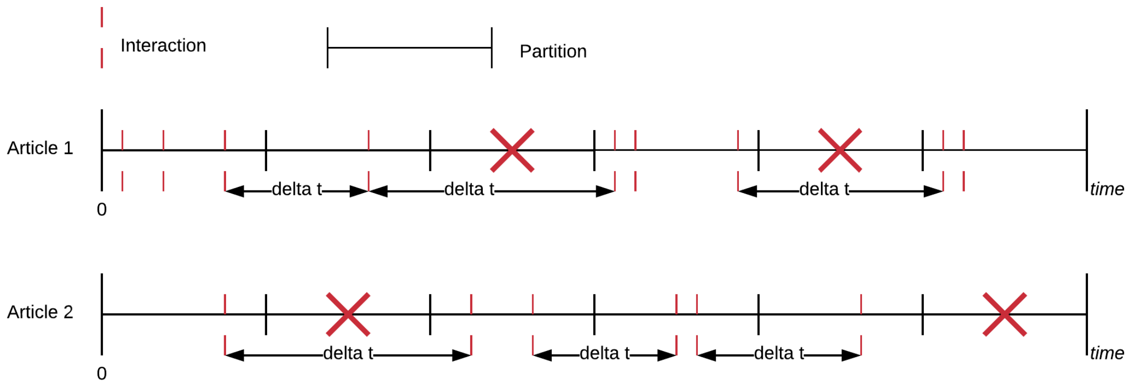


Figure 4.2: Response features

Text features

Text feature vector is related to the text available in each interaction with a given article a_j. The first step has been the generation of clean texts, so we removed the punctuation symbols from the original interactions. Then we choose the Natural Language Processing doc2vec [LM14] Neural Network model to avoid hand-crafted textual feature selection. In this way we automatically extracted a compressed and less noisy representation of the text of each interaction. Doc2vec Hyperparameters will be part of parameters of this model.

Source features

Source feature vector is a specific vector for each user. We want represent users by the articles each one interacted with. To do so, in line with existing literature on information retrieval and recommender systems, we construct the binary incidence matrix of which articles a user engaged with. This matrix is a (n, m) matrix in which

$$\begin{cases} x_{i,j} = 1, & \text{if } \exists t \mid \exists e_{ijt} \\ x_{i,j} = 0, & \text{otherwise} \end{cases}$$

Since the number of 0 will be much higher than the number of 1, in order to be more efficient and avoid memory waste, we implemented it as a sparse matrix. We therefore applied the Singular Value Decomposition (SVD) to extract a lower-dimensional representation for each u_i. Then, each tweet has been linked with the right user feature vector.

Prediction model

We can formalize some points we discussed so far: A is the set of articles, U is the set of users, T is the discrete set of time the articles have been written. P is the set of partitions and $P_{ji} \subset Pj \subset P$ is the $i - th$ partition linked to event j. e_{ijt} is an interaction of user $i \in U$ to article $j \in A$ at time $t \in T$. L is the length of each partition. F_{ijk} is the value of the $i - th$ feature of the partition $j - th$ of event k

$$\forall j \in A, \exists i \in U, \exists t \in T \mid \exists e_{ijt}$$
$$e_{ijt} \in Psu \leftrightarrow (j = s \wedge t \geq (L \cdot u) \wedge t < L \cdot (u + 1))$$
$$\forall i \forall j \forall k \nexists e_{abc} \in P_{jk} \implies F_{ijk} = 0$$

From the above conditions is clear that each article can now be seen as a multivariate time serie (a sequence of partitions, each one linked with several features). Each partition group of each article is characterized by the following variables:

- η
- Δt
- The averages of the Tweets text features, computed over all the Tweets in a partition subset
- The averages of the source features, computed over users whose Tweets are in the partition

Table 4.2: Example of multivariate time series

article	partition	feature η	feature Δt	avg. sources feature 0	avg. sources feature 1 ...
1	0	3	0.0	5.066724e-03	6.042654e-05
1	1	1	50643000.0	1.487121e-05	2.883997e-08
1	2	2	122161000.0	2.265753e-09	2.081532e-10
2	0	1	0.0	3.168470e+00	1.224494e-03
2	1	2	43371000.0	2.625489e-03	1.898865e-05

This table shows a sample of time series. For each article there are a number of partitions and for each partition there are features previously discussed

The problem now is related to the classification of multivariate time series (a timeline characterized by the aforementioned) features. We implemented with Tensorflow + Keras, two popular open source neural network frameworks, the state-of-the-art Deep Learning MLSTM-FCN model described in [FSHS18]. The input to this Neural Network will be a 3-dimensional matrix $(articles, features, partitions)$ and the output will be a sequence of binary values (for each article - so for each sequence of partitions - if it is Fake or not).

This model comprise of a fully convolutional block and a LSTM block, as shown in 4.3. The fully convolutional block contains three temporal convolutional blocks, used as a feature extractor. Each convolutional block contains a convolutional layer, with filter size of 128 or 256, and is followed by a batch normalization. The batch normalization layer is then followed by the ReLU activation function. In addition, in each of the two convolutional blocks, a squeeze and excite block completes the work. The final temporal convolutional block is followed by a global average pooling layer. Detailed Neural Network description is available in Appendix B.

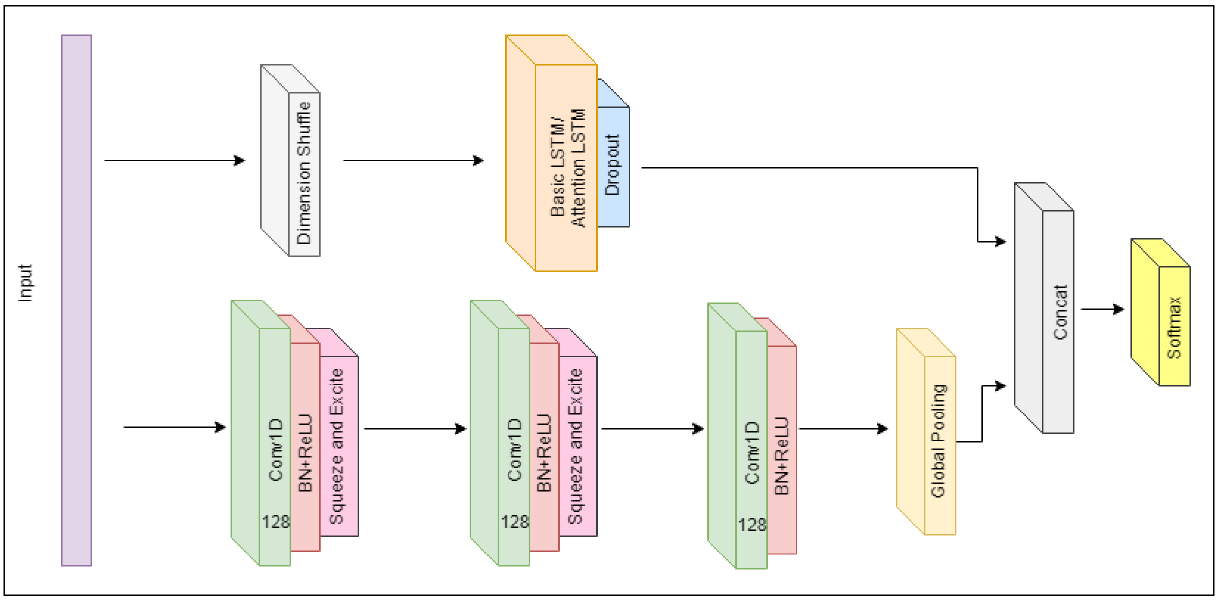


Figure 4.3: The MLSTM-FCN architecture [FSHS18]. LSTM cells can be replaced by Attention LSTM cells to construct the MALSTM-FCN architecture [FSHS18]

4.4.4 Results

We performed an evaluation of this model with K-Fold Cross Validation over 10 folds. This evaluation method consists on training and testing the model K (in our case 10) times with different portions of the dataset. In this way, it helps avoiding overfitting and asymmetric sampling. Full results are reported in the table. We can notice a quite good value in the average accuracy: more than 73% of the times this model predicted the article right label. Range of Accuracy values is quite reduced and we can notice peaks of more than 78% of accuracy. Variance is pretty low, less than 0.00163 for accuracy and less than 0.00035 for the Mean Square Error. The Normalized Average Confusion Matrix can help us understanding more about the accuracy. It has two independent variables:

- True Label: This variable tells us if the input article is a Fake Article or not
- Predicted Label: This variable gives us the result of our model for the same input article (Fake or Not Fake)

Therefore we have 4 different values: A Fake News is usually spotted 77% of the times, instead a True news received the right prediction 69% of the times. This can be caused by the fact that True articles may have a more range of temporal features pattern. Therefore, the probability of Fake article classified as Not Fake is slightly less than a True article classified as Fake. In Appendix A Confusion matrixes for each fold are reported.

Fold	MSE	Accuracy
1.0	0.185110	0.776923
2.0	0.238600	0.700000
3.0	0.208551	0.715385
4.0	0.189748	0.776923
5.0	0.197346	0.738462
6.0	0.227610	0.646154
7.0	0.218941	0.705426
8.0	0.175922	0.782946
9.0	0.203156	0.744186
10.0	0.213187	0.736434
Average	**0.205817**	**0.732284**

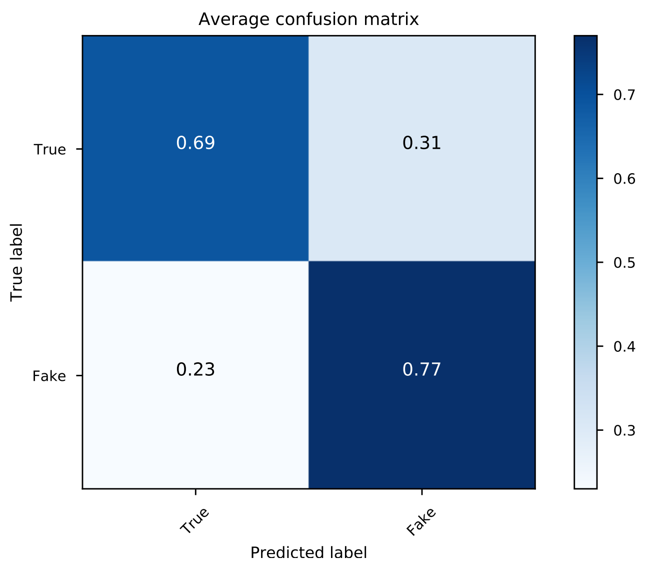


Figure 4.4: Normalized Average Confusion matrix

4.5 FastText

FastText is a library for learning of word embeddings and sentence classification created by Facebook's AI Research (FAIR) lab. The model is an unsupervised learning algorithm for obtaining vector representations for words [JGBM16]. Following are the some of the tuning parameters of we used in training phase.

- Epoch : By default, FastText take each training sample 5 times. It's very small given a comparatively small training data. So, we tried different values and came up with optimum value of 20.
- Learning rate : This represents how much the model changes after processing of each training sample. The values ranges from 0 to 1. If we choose 0 the model won't change after each step; means it won't learn anything. We have choosen learning rate as 1.0.
- Word n-grams : The order of words is really important in text classification problem. If we are taking the individual words as independent entities, it results to poor performance compared to n-grams where n $>$ 1; such as bigrams, tigrams etc. In this model, we are taking trigrams as it yeilds better accuracy.

By tuning the aforementioned parameters we were able to improve the accuracy in the range of 8% to 12% for training data contain different number of tweets per event. The confusion matrices of FastText models for different datasets(different number of articles per event) are given in 4.3. The average accuracy of these models is 68.56956%.

Table 4.3: Confusion matrices of FastText models

Number of test samples : 1248 (Training data contain 5 posts per event) Accuracy : 70.03205%			Number of test samples : 2275 (Training data contain 10 posts per event) Accuracy : 71.82417%		
	Fake(Predicted)	Real(Predicted)		Fake(Predicted)	Real(Predicted)
Fake(Actual)	320	197	Fake(Actual)	605	309
Real(Actual)	172	559	Real(Actual)	326	1035
Number of test samples : 3242 (Training data contain 15 posts per event) Accuracy : 69.95682%			Number of test samples : 3951 (Training data contain 20 posts per event) Accuracy : 62.4652%		
	Fake(Actual)	Real(Predicted)		Fake(Predicted)	Real(Predicted)
Fake(Actual)	1053	523	Fake(Actual)	419	1318
Real(Actual)	441	1225	Real(Actual)	157	2057

4.6 Sentiment Analysis

4.6.1 IBM Watson NLU

The IBM Watson Natural Language Understanding (NLU) is a really powerful service, provided by IBM in cloud, that allows the user to analyze a sentence, a paragraph or even an entire article. The functionalities provided by this service are many:

- Sentiment analysis
- Emotion analysis
- Identify keywords, entities, concepts and their relations

After trying different combinations and settings, we've decided to analyze the texts of the articles linked to every tweet with IBM Watson NLU in order to extract keywords and the attributes associated to each of them. The outputs of the NLU for each keyword are: the score for the general sentiment, the score for each emotion (joy, sadness, disgust, fear, anger) and the score for the relevance of the keyword in the sentence.

4.6.2 Text extraction, data cleaning and features

As mentioned above, the analysis was done over the articles contained in the tweets. Specifically, the text of the articles was retrieved by using an existing API [Hed10]. After the analysis performed by the NLU, we've cleaned the data by removing all the tweets which had a "Null" score in the output of the sentiment and of the emotional analysis. Indeed, the URL contained in these kind of tweets led to non-articles pages, like links to YouTube videos and to other tweets and therefore could not be analyzed by the NLU. Moreover, since the project was considering only English tweets and articles, all the non-English text was not analyzed and the corresponding tweet was removed from the dataset. As features, to compute a single value for each emotion and for the general sentiment, we've decided to perform an average of the scores of the target emotion (or sentiment) of all the keywords identified by the NLU, weighted with the relevance in the phrase of the keyword itself.

4.6.3 Training and testing

For computational issues and because of the limit to the request that can be done to an IBM Watson API, the model was trained and tested with a small fraction of the total dataset. In particular, the dataset was divided by event and 80% of it was taken for training and the remaining part for testing. Moreover, because of the aforementioned reasons, only 5 tweets per event were taken to build the new dataset.
After different tests and evaluation, the method with the best performance over the dataset was a Quadratic Discriminant Analysis (QDA), with an accuracy of 69% on the test set.

Table 4.4: Confusion matrix for QDA using 5 tweets per event

	True (Predicted)	False (Predicted)
True (actual)	530	71
Fake (actual)	220	129

4.7 Final Ensemble

As explained at the beginning of this chapter, the three models previously presented were put together in an ensemble to generate a final prediction. In particular, the final output will be based on a Majority Voting algorithm, where a simple majority vote of decision made by the models is done. The final model has shown better performance on the dataset available with respect to every other single classifier tested, reaching a precision of 78%.

Conclusion

In this document, we have presented our approach for detecting fake news in social media data. After proper preprocessing of data, we employed an ensemble based approach, where we used three different sophisticated machine learning approaches followed by majority voting mechanism to better capturing the uncertainity of data social media possess normally.

One of the main problems faced during the analysis is the quality of the data, that affects a lot the overall performance. Generally speaking, the data quality is a critical aspect that is usually much more important than the algorithm itself. In this specific case, both FastText and NLU used the text articles obtained from tweets that were retrieved through an external API [Hed10] which sometimes didn't perform a perfect retrieval. The student version of IBM Watson does not allow to make unlimited use of their API, therefore we could not analyze all the dataset available. It is legit to consider that, especially for the sentiment analysis, a greater amount of data would have been better and would have led to better results.

In the first model we could try partitioning tweets in more complex ways, for example adopting density-based clustering algorithms like "Density-Based Spatial Clustering of Applications with Noise" (DBSCAN) [EKSX96] and adding a feature that stores the length of each partition. Another improvement could be training other algorithms and ensemble them in more sophisticated and complex ways, which may lead to better results. With IBM Watson and other tools could be also possible to perform other kind of analysis of the text, like semantic analysis or classification of the text, that could help to better classify the veracity of a news. Something that would definitely boost up the performance of the algorithm would be the implementation a system to perform cross-checking of a certain news, by looking for the content of the article on trustworthy websites.

Bibliography

[Adn18] Adnkronos, Ed. (2018). Vaccini e autismo, la regina delle fake news compie 20 anni, [Online]. Available: `http://www.adnkronos.com/salute/sanita/2018/02/22/vaccini-autismo-regina-delle-fake-news-compie-anni_Td4gMZuSL888mu3YX1CFRK.html?refresh_ce` (visited on 06/02/2018).

[Edk16] B. Edkins. (2016). Americans believe they can detect fake news. studies show they can't. Forbes, Ed., [Online]. Available: `https://www.forbes.com/sites/brettedkins/2016/12/20/americans-believe-they-can-detect-fake-news-studies-show-they-cant/#5798f89e4022` (visited on 05/20/2018).

[EKSX96] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise", in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, ser. KDD'96, Portland, Oregon: AAAI Press, 1996, pp. 226–231. [Online]. Available: `http://dl.acm.org/citation.cfm?id=3001460.3001507`.

[FSHS18] K. Fazle, M. Somshubra, D. Houshang, and H. Samuel, *Multivariate lstm-fcns for time series classification*, 2018. eprint: `arXiv:1801.04503`.

[Hed10] J. Hedley. (2010). Jsoup java html parser, [Online]. Available: `https://jsoup.org/` (visited on 04/15/2018).

[Hut13] S. Hutchinson. (2013). Social media plays major role in turkey protests. BBC, Ed., [Online]. Available: `http://www.bbc.com/news/world-europe-22772352` (visited on 06/02/2018).

[JGBM16] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification", Jul. 2016.

[Kap16] M. Kapko. (2016). How social media is shaping the 2016 presidential election. IDG, Ed., [Online]. Available: `http://www.cio.com/article/3125120/social-networking/how-social-media-is-shaping-the-2016-presidential-election.html` (visited on 06/02/2018).

[LM14] Q. Le and T. Mikolov, "Distributed representations of sentences and documents", in *Proceedings of the 31st International Conference on Machine Learning*, E. P. Xing and T. Jebara, Eds., ser. Proceedings of Machine Learning Research, vol. 32, Bejing, China: PMLR, 22–24 Jun 2014, pp. 1188–1196.

[MGM+16] J. Ma, W. Gao, P. Mitra, S. Kwon, B. J. Jansen, K.-F. Wong, and M. Cha. (2016). Detecting rumors from microblogs with recurrent neural networks, [Online]. Available: http://alt.qcri.org/~wgao/data (visited on 05/01/2018).

[Shu17] K. Shu. (2017). Kaidmml dataset. ACM, Ed., [Online]. Available: https://github.com/KaiDMML/FakeNewsNet (visited on 04/17/2018).

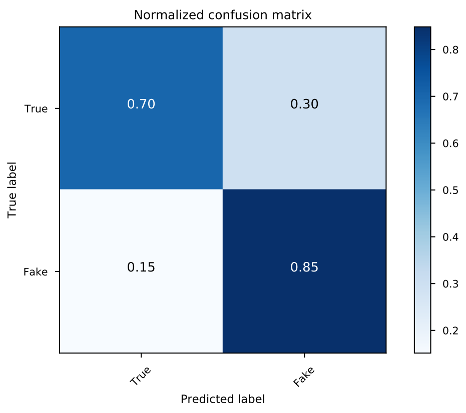
[Sta] I. L. Stats, Ed. (), [Online]. Available: http://www.internetlivestats.com/twitter-statistics (visited on 06/02/2018).

[TBD+17] E. Tacchini, G. Ballarin, M. Della Vedova, S. Moret, and L. de Alfaro, "Some like it hoax: Automated fake news detection in social networks", Sep. 2017.

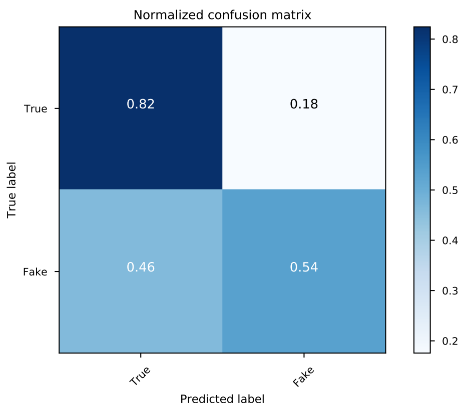
[The05] The Economist Intelligence Unit, "Reputation: Risk of risks", *The Economist*, 2005.

[Twi18] Twitter. (2018). Consuming streaming data, [Online]. Available: https://developer.twitter.com/en/docs/tutorials/consuming-streaming-data.html.

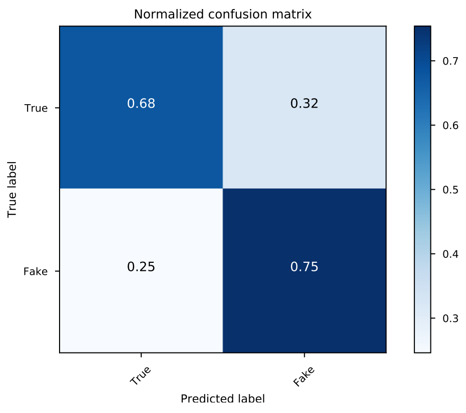
K Fold Confusion Matrices



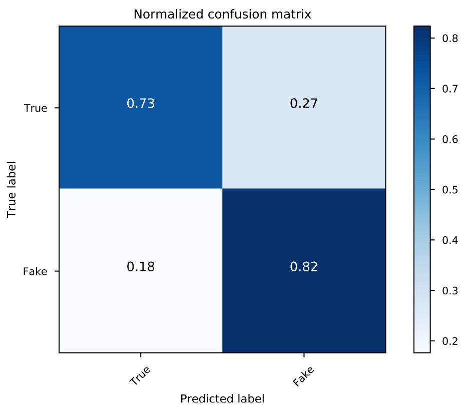
(a) Confusion matrix fold 1



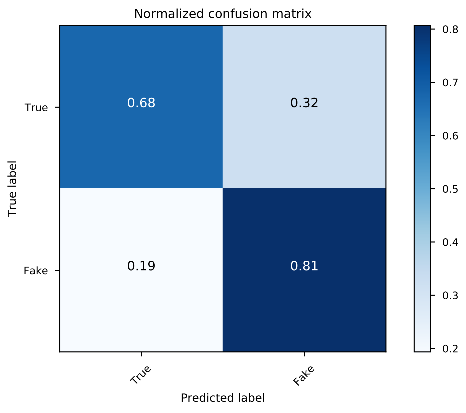
(b) Confusion matrix fold 2



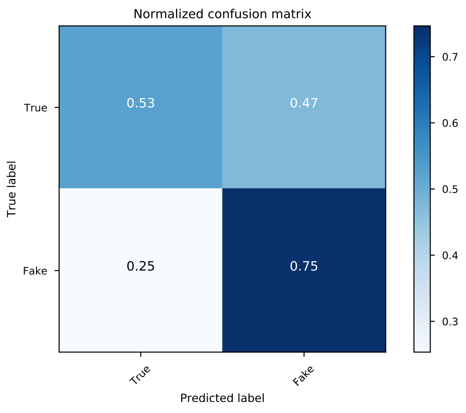
(c) Confusion matrix fold 3



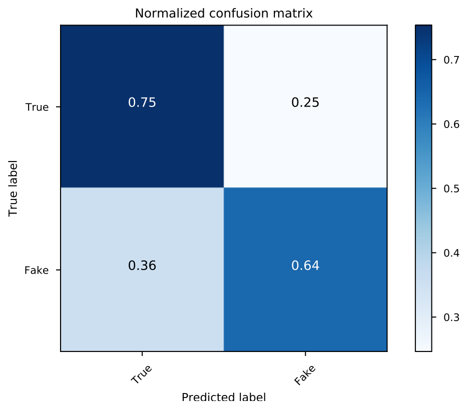
(d) Confusion matrix fold 4



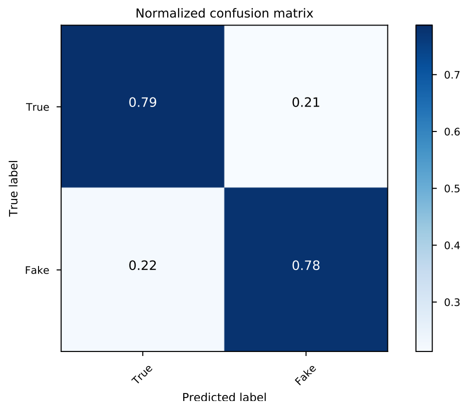
(e) Confusion matrix fold 5



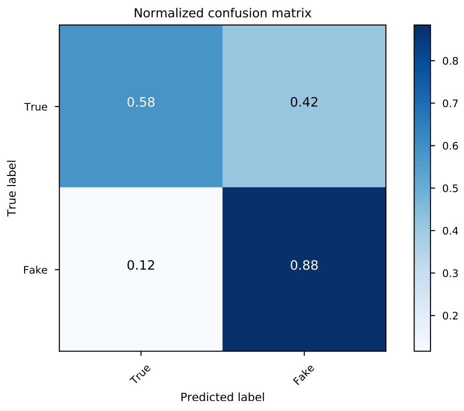
(f) Confusion matrix fold 6



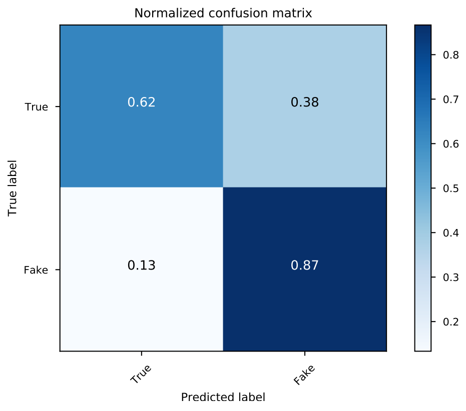
(a) Confusion matrix fold 7



(b) Confusion matrix fold 8

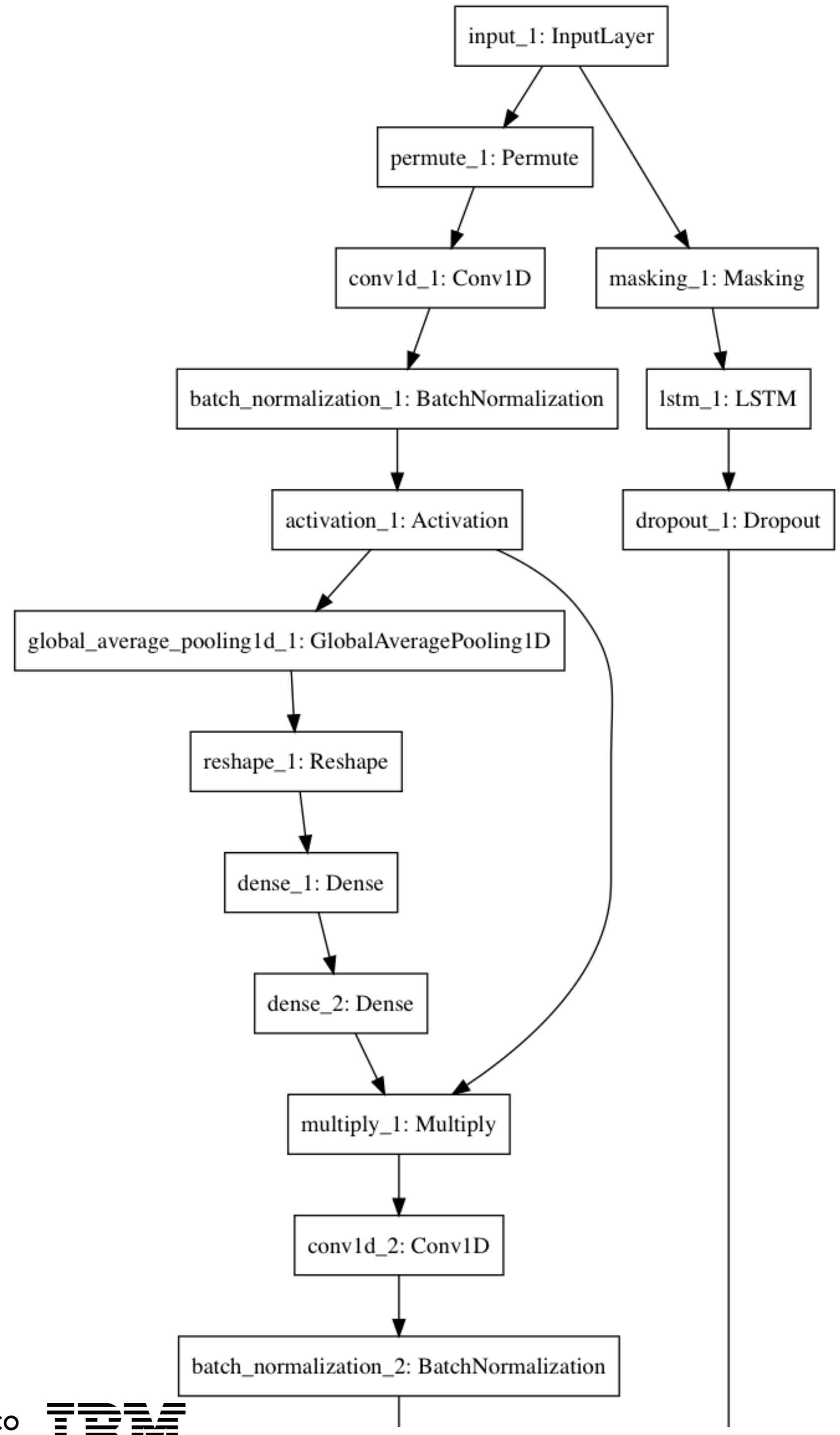


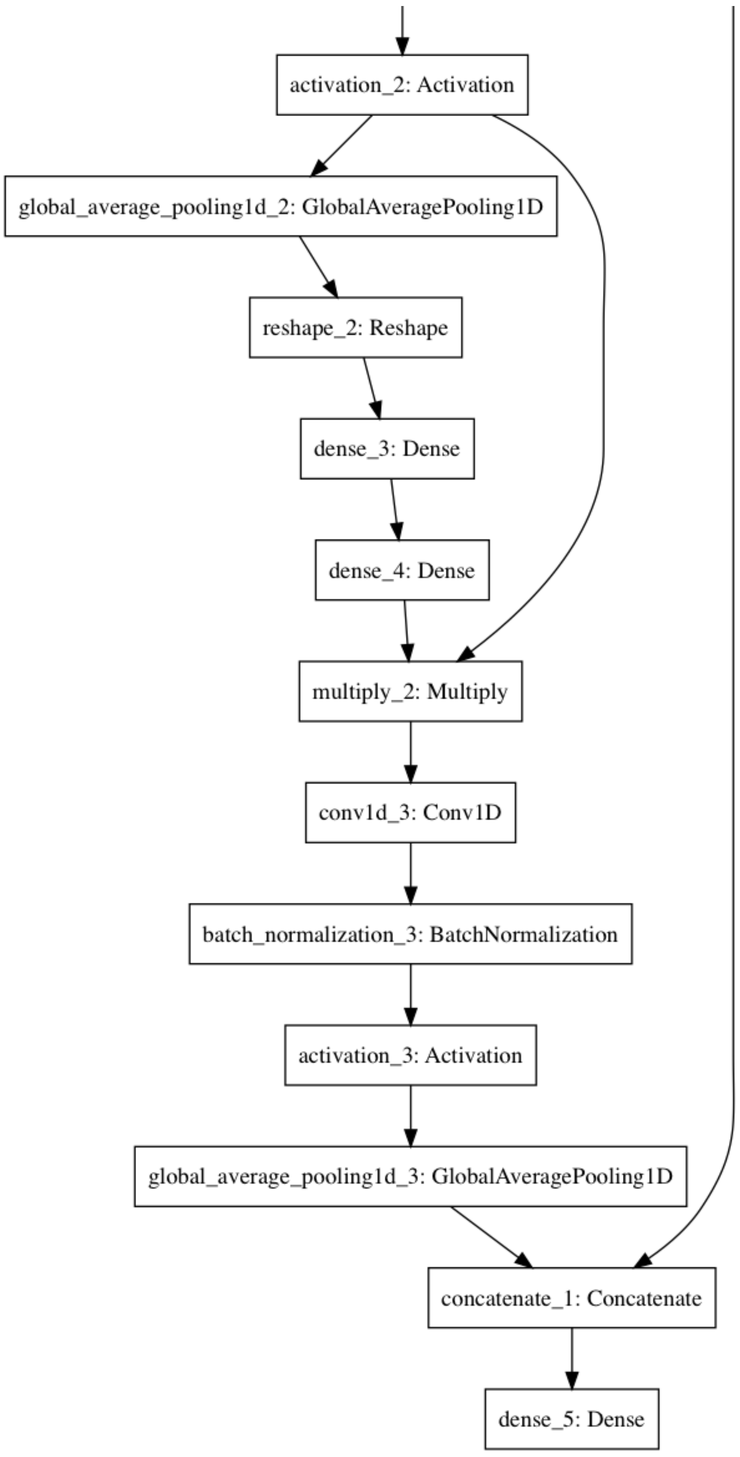
(c) Confusion matrix fold 9



(d) Confusion matrix fold 10

Detailed classification model




activation_2: Activation
global_average_pooling1d_2: GlobalAveragePooling1D
reshape_2: Reshape
dense_3: Dense
dense_4: Dense
multiply_2: Multiply
conv1d_3: Conv1D
batch_normalization_3: BatchNormalization
activation_3: Activation
global_average_pooling1d_3: GlobalAveragePooling1D
concatenate_1: Concatenate
dense_5: Dense