Milestone 1
Fake News Analyzer

Ana Clara Moreira Gadelho - up201806309
Flávia Carvalho Gavinha Pereira Carvalhido - up201806857

FEUP - MEIC - Information Processing and Retrieval

November 16, 2021

Abstract

This report aims to detail and complement the making of a project from the curricular unit Information Processing and Retrieval, in which the authors have chosen a dataset comprising data from various fake news articles and proceed to work on it in the ways described throughout this document.

1 Introduction

The project described in this report consists in performing a set of actions, as defined in a data pipeline that will later culminate in the making of an information system revolving over a chosen dataset, in this specific case, a fake news collection from 2016.

The report will start by discussing the dataset, it's source, trustworthiness, structure and consistency. After this, the report will focus on the data pipeline built to showcase the path the data will go through during this project.

Furthermore, all the steps described in the data pipeline will be explored in more detail in the following sections. All of these sections will in some way contribute towards the final objective of this project: having an information system which allows to explore and find patterns or stats in atrocious news containing fake information.

2 Fake News Dataset Concepts

Although fake news and the explored dataset seem pretty straightforward, there are a few concepts that should be taken into account before moving forward in this report, namely to better understand some columns present in the explored dataset.

Thread Title - Besides the news article title, there is a thread title regarding a thread, where that same article and some others are. We can consider the thread title the bigger picture in which the news article is inserted.

Spam Score - This is a metric calculated to evaluate the level of likelihood of a news article being spam, as it is present in many websites and platforms. It varies between 0 and 1, 0 not being spam and 1 being spam.

Domain Rank - Another metric that represents the authority of a certain domain, developed by Moz. It is described as such: *"a search engine ranking score developed by Moz that predicts how likely a website is to rank in search engine result pages (SERPs). "*[1]

Type - This is not a widely accepted classification system, but rather something used by the original author of the dataset when choosing and crawling the data. The news are classified as 'bs', 'junksci', 'bias', etc. by the BS Detector Chrome Extension by Daniel Sieradski, in a way which is further described in his repository, where the open source code used for the tool is present.[2] The original author of this dataset used this tool to classify both the news and the websites in which all the data was extracted from.

3 Dataset Research and Selection

The authors of this report focussed their research for datasets in the online platform Kaggle, where the NLP datasets were surely to be the ones with more text columns to process. This was also the most important aspect the dataset should accomplish, thus ruling the research direction. After visiting and comparing many datasets repositories, as well as looking at already carried out research on those same datasets, the chosen one ended up being "Getting Real about Fake News - Text metadata from fake biased news sources around the web" by Meg Risdal[3]. This dataset comprised 12999 news articles from 2016 from 244 websites, classified by the B.S. Detector[2] as fake. As it is of common knowledge, 2016 was the year of fake news, as a very important election - Trump vs. Hillary - was taking place, creating all sorts of rumors connected to all different things.

This dataset is classified by the platform Kaggle and its users as a 7.1/10 in terms of usability, which is a very good usability score, specially because the dataset is pretty complete and contains a lot of relevant information. The data was originally crawled from the "bs" classified websites[2] using the webhose.io API[4],

and because it's coming from their crawler[4], not all websites identified by the BS Detector are present in this dataset. Each website and article were labeled according to the BS Detector[2] as registered in column 'type'. Data sources that were missing a label were simply assigned a label of "bs" by the original dataset author. There are (ostensibly) no genuine, reliable, or trustworthy news sources represented in this dataset, so don't trust anything you read.

The tools further used for the following steps were Microsoft Excel, for the first analysis, and Jupiter Notebooks (Python) for the remaining processes.

3.1 Dataset structure

The dataset has got 12999 rows and 20 columns, some of which will be dropped later as described in the Data Cleaning section. The 20 columns each describe one aspect from the news article, namely: uuid, order in the thread, author, published date, title, text, language, crawled date, site url, country, domain rank, thread title, spam score, image url, replies count, participants count, likes, comments, shares and type.

Some of these concepts are pretty straightforward and others were already explained before in the section *Fake News Dataset Concepts*. They all come together in this dataset to provide us more information about the classified news articles. Some of the data can be dropped, as it doesn't provide useful information for the information system, such as the whole 'main image url' column and other broken rows or null values, however this will be better specified in the next section.

This dataset has got mostly string columns, containing titles, text, names, etc. It also has got int columns for the counters and and a float column for the spam score, which is a value between 0 and 1.

The exploration process begins with some light analysis in a data visualizer, in which the main problems of the dataset were identified (broken rows, null values, etc). After identifying the next steps to be taken, the data processing begins, with actions further described and completely automated through the use of a Makefile.

4 Data Domain Conceptual Model

The data was modelled into a domain conceptual model in UML. The tables used were 'Thread', 'Article', 'Author' and 'Website', all connected to each other as described in the picture below. The 'Article' is the center piece of this model, containing most of the information and being our most essential table, as most search will occur in the information it comprises. There is also a 'Type' enumeration table, containing all the values that the column 'type' can have, as it is a classification of the Article itself. The table with the most entries will definetly be the 'Article' table, as all the other can have values that relate to the Article entries more than once. All the field types are also mentioned in this model and come into play when cleaning the dataset, as it will be mentioned in the Data Cleaning section.

UML.drawio.png

Figure 1: UML Class Diagram.

5 Data Processing Pipeline

After modelling the data, a pipeline construction was carried out in order to plan out all the different stages the data would go through before becoming completely usable and ready for the information system. First of all, the data was taken out of the Kaggle platform, in a .csv file. It then goes through the Data Cleaning and Data Analysis steps, using Python, more specifically using the libraries Pandas and MatPlotLib. This process allows us to prepare the data and get completely acquainted with the dataset. After doing this, the data was split into 4 different .csv files, according to the conceptual model. This step was made as a preparation to then store the data into a sql database. The raw data is now free of cluter and stored, both in a csv file and in a SQL database for further processing.

Pipeline.jpeg

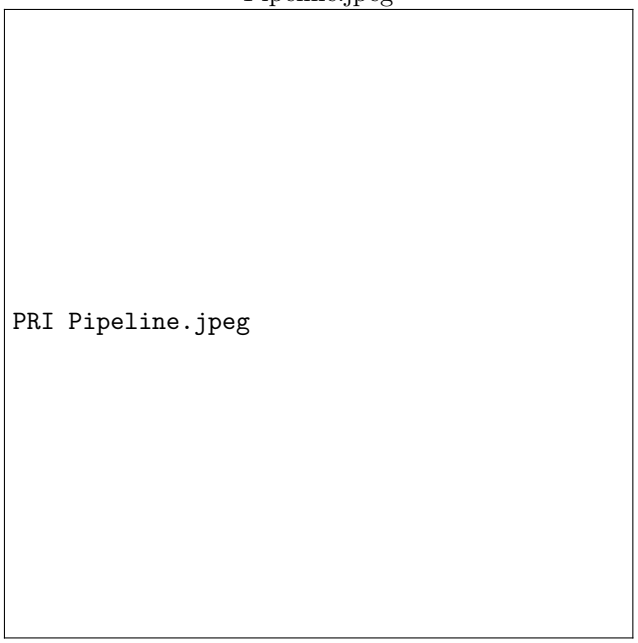

Figure 2: Pipeline diagram.

6 Dataset cleaning

For the cleaning process there were two main steps: fixing missing values and correcting broken or wrong values.

To fix missing values, firstly the *fake.csv* file containing all the data was loaded to the Jupiter Notebook and read using the Pandas Library with the argument to make every null value an empty string, so they are easily identifiable. After doing so, all the duplicate rows

were dropped and each column containing null values was identified and analysed, in order to evaluate how to deal with the missing data from each of these cases.

The columns with missing data from the initial dataset were: domain_rank: 4223 n/a rows; main_img_url: 3643 n/a rows; author: 2424 n/a rows; title: 680 n/a rows; country: 176 n/a rows; text: 46 n/a rows; and, thread_title: 12 n/a rows.

Starting with domain rank column, the decision was to fill all the nonexistent domain rank values with 0, assuming that the website would rank 0 (minimum value) or wasn't good or visited enough to even get a ranking.

The main img url column was dropped completely, was it wouldn't provide any relevant information to be explored using text search, for example.

The author column was a case of null value substitution, as some authors were already called "Anonymous" and not having an author is a very similar situation. So all the null values in this column were substituted by "Anonymous".

As for the title column, it was noticed that some titles were the same as thread-titles. Initially, it was thought that only the first news article from a thread would get a title and the remaining ones wouldn't and that the title was always the same as the thread-title. It was later proved that it wasn't the case and that the missing values were just missing because the crawler only associated the title with the first news article in the page (ord_in_thread equal to 0). It was clear that it was possible to assume that all the news present were connected to the thread-title anyway, allowing it to fill the title column missing values with the respective thread title values, since the only missing title values were groups from the same thread, just not the first ones in the thread (ord_in_thread bigger than 0).

The missing country values, text values and thread_title values were all dropped (as in, the row was all dropped), because the number of rows wasn't very significant and not having that data wouldn't allow the rows to be processed and evaluated properly. It was also impossible to find replacement for that missing data.

Last but not least, during this step, it was also noticed that the uuid column didn't add anything to the information system, as it was a mere hash value and the whole column was dropped as well.

Moving on to the next step of the cleaning process: correcting broken or wrong values. There were several broken columns in the dataset and it is a bit hard to identify them using a general formula. Some columns had the text all scattered out through them because they weren't properly crawled, there were random 0, 1 and values with just white spaces and sometimes even emojis. The way this was all cleaned was tackling the bigger problems first (cleaning all the 0 and 1) and later, confirming that all the columns didn't have any white space string or wrong type values, so all the remaining columns went through a type verification and throughout cleaning process, using regex and match patterns, adapted to each case. After all the columns were clean, the clean dataset was registered in a csv file called fake_clean.csv.

7 Dataset analysis

After the cleaning process, the dataset was ready to be analysed, having 12650 rows and 18 columns. Several plots were made to better understand the data we're dealing with.

7.1 Country Distribution

The country of origin of each news article was analysed in the graph of the Figure 3, from which we can observe that the big majority of the articles come from the US. This goes in line with the year in which these articles were written (2016) as the elections were going on in the US and most of the misinformation was being produced for the American people.

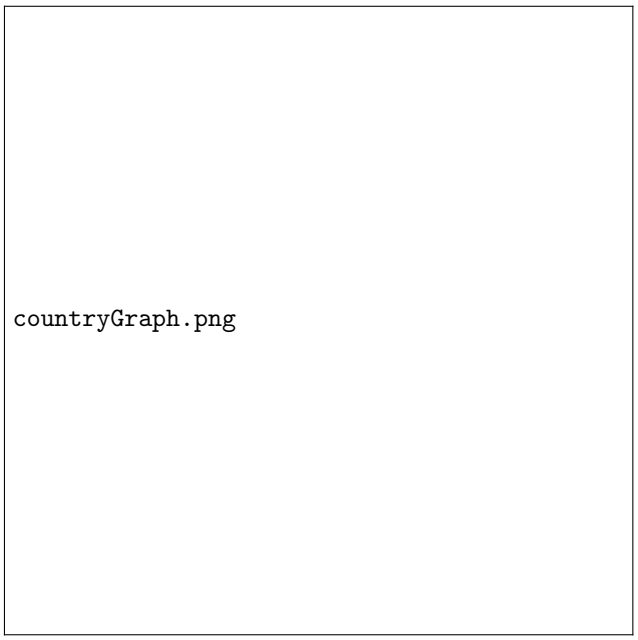

Figure 3: Country Distribution Chart.

7.2 Language Distribution

In terms of the language in which each article is written, it can be seen that almost all are written in English, but a total of 15 languages are included in this dataset.

7.3 Type of article

In the graph below, we can see the distribution of the articles in terms of the type of fake news that they represent.

7.4 Domain Rank

As explained above in this report, the domain_rank variable evaluates the score of a website when searched for. It can be seen in the chart below that the majority

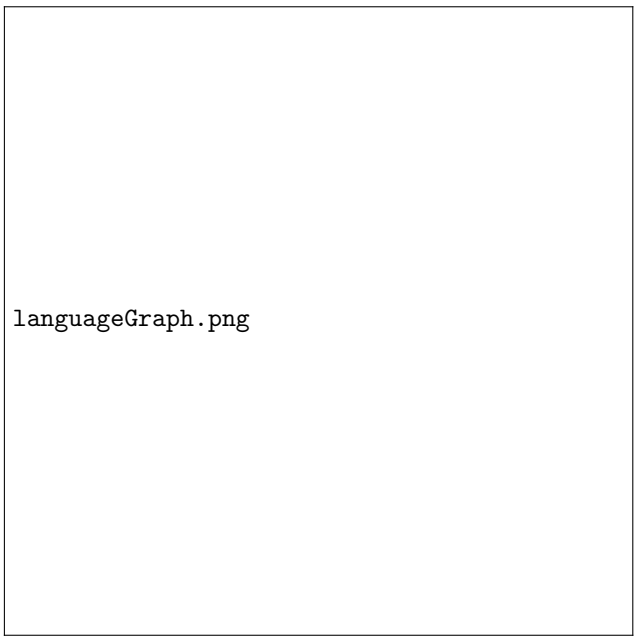

Figure 4: Language Distribution Chart.

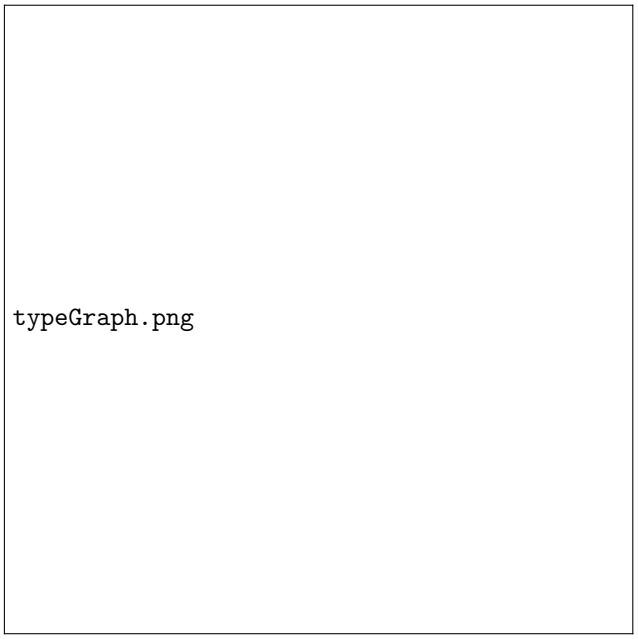

Figure 5: Type Distribution Chart.

of websites present in the dataset have a scored around 0, which makes sense since they are websites with fake news.

7.5 Domain Rank

A analysis of the articles' comments, shares and likes was carried out, resulting in the graph below, that has the averages of each value. It can be seen that the articles of this dataset don't have a significant amount of likes or shares and in average have almost no comments.

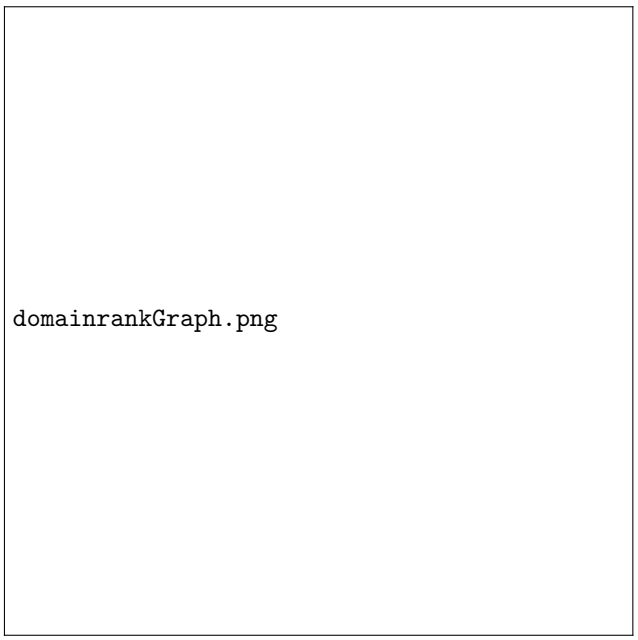

Figure 6: Domain Rank Box Chart.

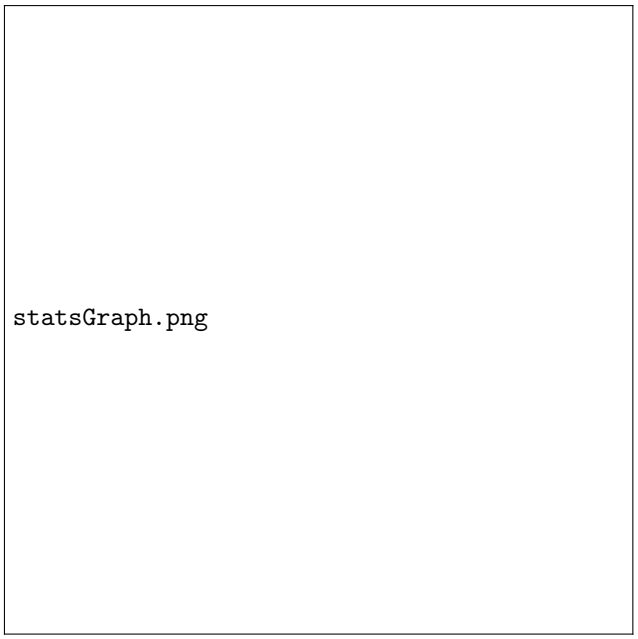

Figure 7: Article Statistics Chart.

8 Possible Search Queries

As this is a dataset to be applied in an information system, this information system will allow the data to be used in many ways. It is possible that some queries are more common than others, so in this section some common and more useful search queries will be described.

8.1 Theme search

It is very possible for this tool to be used to search for fake news related to more controversial themes, using text inputs such as "2016 elections", "Trump", "corruption", "Russia", "conspiracy theories", "nuclear war", etc. This information system will have to

be prepared to withstand these text queries and search for relevant articles or threads to show as search results.

8.2 Author, website and thread stats

Additionally, users might want to figure out some stats per website, author and thread, such as the number of articles of each, the most common type of articles they have, average number of likes and shares per article, etc. Therefore, it is important for the tool to have these stats for each one of these based on the articles associated with them.

[1] - “What is Domain Authority and why is it important?” Moz, https://moz.com/learn/seo/domain-authority.

[2] - ”B.S. Detector” selfagency/bs-detector, https://github.com/selfagency/bs-detector

[3] - ”Getting Real about Fake News -Text metadata from fake biased news sources around the web, by Meg Risdal” Getting Real about Fake News, Kaggle, https://www.kaggle.com/mrisdal/fake-news

[4] - ”Webhose.io” webhose.io, https://webhose.io/auth/login?redirect=/web-content-api