"Fake News Detector": An Automatic System for the Reliability Evaluation of Digital News

Claudio Cilli[a], Giulio Magnanini[b], Lorenzo Manduca[c] and Fabrizio Venettoni[d]

Department of Computer Science, La Sapienza, University of Rome, Rome, Italy

Keywords: Fake News, Misinformation, Disinformation, AI, Scraper, Deep Learning, NLP, Cybersecurity, 5W.

Abstract: Nowadays, information is taking on an increasingly central role in people's lives. With the rise of internet, the amount of information has grown exponentially as the ease of publishing content of all types has increased. At the same time, however, the risks deriving from the lack of checks on the truthfulness of these have also increased. In fact, a "Fake" information content can lead to serious reputational, economic or health damages. To overcome the problem of verification, several studies have been carried out, but none of them appears to have been the subject of a significant commercial implementation. In all the projects proposed so far there is a common thread, that is to act directly to the source of the news, examining, thanks to existing technologies, the possible truthfulness of the same. On the other hand, lacking a real involvement of the reader, these projects were not suitable to increase the awareness of the end user. This work implements a technological platform able to provide a reliability value of a digital news, thus measuring the level of impartiality of the author through the evaluation of a defined series of parameters.

1 INTRODUCTION: STATE OF ART

"Fake News" is a phenomenon that has always characterized the history of mankind and has a dramatic impact on the society. Today, Fake news are considered one of the biggest threats to democracy, justice, public trust, freedom of expression, journalism and economy. Nowadays there is a large body of scientific literature on the subject. Among the many proposals there are several articles on the subject, which, for the most part, propose the use of Machine Learning technologies, Neural Networks, Multimodal systems, BlockChain, Deep Learning. In all the projects proposed so far there is a common thread, that is to act directly to the source of the news, examining, thanks to existing technologies, the possible veracity of the same. Lacking the focus on the user, these projects are not suitable to increase the awareness of the end user, the primary objective of this work. However, a list of major works is given:

- **MVAE:** Multimodal Variational Autoencoder for

---

[a] https://orcid.org/0000-0001-9558-2565
[b] https://orcid.org/0000-0002-0006-5600
[c] https://orcid.org/0000-0002-8778-3006
[d] https://orcid.org/0000-0002-7592-6479

Fake News Detection that proposes, for the detection of fake news, a bimodal variational autoencoder coupled with a multimodal (textual + visual) binary classifier; (Khattar et al., 2019)

- **DeHiDe:** Hybrid model combining blockchain technology with an intelligent deep learning model to strengthen robustness and accuracy in combating fake news; (Agrawal et al., 2020)

- **dEFEND:** a fake news detection system that leverages user comments to verify whether the news is fake or real. (Shu et al., 2019a)

- **Fake News Early Detection: A Theory-driven Model:** In this paper, a theory-driven model for fake news detection is proposed. The proposed method aims to investigate news content at various levels: lexical, syntactic, semantic, and discourse. News is represented at each level, relying on established theories in social and forensic psychology. Fake news detection is then conducted within a supervised machine learning framework. This work explores potential fake news models, improving interpretability in engineering fake news features by studying various aspects of them. (AZhou et al., 2019)

In addition to the scientific literature, some tools and utilities are highlighted such as:

- **Fiskkit:** a platform created by John Pettus that aims to build a place to promote consistency and neutrality of information through participant submission of feedback to help people identify what is true, false, well-argued, or logically incorrect in articles or opinions. (Pettus, 2018)

- **TextThresher:** web interface that refines the social science practice of content analysis, making it more transparent and scalable to hundreds of thousands of documents. (Adams, 2016)

- **FakeNewsTracker:** a tool for collecting, detecting, and visualizing fake news, using datasets and ML models by extracting useful features. (Shu et al., 2019b)

- **FakeNewsNet:** a data repository with news content, social context, and spatiotemporal information for studying Fake News on social media. (Shu et al., 2019c)

- **Detecting Fake News in Social Media Networks:** The purpose of the following work was to find a solution that can be used by users to identify and filter out sites that contain false and misleading information. Using simple and carefully selected features of the title and post, it is possible through the use of the tool to identify false posts. (Aldwairi and Alwahedi, 2018)

- It is worth mentioning, among the projects under development, **SocialTruth** (Demestichas, 2018). This European project wants to provide an innovative and distributed way, thanks to BlockChain and Machine Learning technology, to achieve both content and author credibility verification and fake news detection, in order to increase trust in Social Media. However, the project started in 2019, is still in the implementation phase, with no relevant results published so far.

2 FAKE NEWS DETECTOR: A TOOL TO ASSIGN RELIABILITY TO NEWS

2.1 Why "Fake News Detector"?

"Fighting fake news is like battling a many-headed Hydra while swimming in a tsunami of slime." (Govindraj Ethiraj)

As described in the previous paragraph, all the projects defined act directly to the source of the news to verify the reliability of the content. Lacking the focus on the user, these projects are not suitable to

increase the awareness of the end user, the primary objective of this work. Other limits arised from the existing projects are:

Lack of an Implemented Tool: there isn't any standard tool that verifies fake news;
Lack of a Tool that Implements the 5W's Theory: 5w's theory is one of the standard in the fact checking journalism, but there isn't any tool that implement it correctly;
Tool "No-comment based": a lot of tools often verify news asking for user comments;

Fake news Detector aims to overcome those limits through the implementation of a user-friendly tool that, using Web-Scraping and AI, tries to implement in a very simple way, the 5Ws theory to verify a fake news. This project is related to the creation of a Python application (**Lack of an implemented tool**) that uses a series of functions that assign a score in order to the 5ws parameters (**Lack of a tool that implements the 5W's theory**). Fake news detector also guarantees the principle that the reliability of a news is not determined by what other people think but it depends on a series of objective parameters derived by theory (**Tool "No-comment based"**).

3 INFORMATION, MISINFORMATION AND THE 5WS THEORY

3.1 Information vs. Misinformation

The difference between Fake News and other types of news is their intent. As you can see from the table below, a Fake News has a malicious intention, created with the aim to destabilize the environment and cause havoc. On the opposite, of a False News, we don't know if the intention is necessarily malicious or not as well as for Disinformation and Rumor.

Table 1: Classification of the different types of News.

	Authenticity	Intention
Fake News	False	Malicious
False News	False	Unknown
Satire	Unknown	Not Malicious
Disinformation	False	Malicious
Misinformation	False	Unknown
Rumor	Unknown	Unknown

3.2 The 5Ws Theory

The 5W theory is extremely useful and necessary in order to defend against Fake News. It represents questions whose answers are considered fundamental in gathering information or solving problems. Often used by journalists (Hart, 2012) (MediaSmarts, 2012) (Copy Editing, 2008) to write their articles correctly, these can also be used to assess whether a news item can be considered reliable or not. A news story can only be considered complete if it answers these questions:

- **WHO:** Is there an author of the news item?

- **WHAT:** Type of news item. Is it a News Fact, election propaganda, dogma, maxim, etc.?

- **WHEN:** When was the news story written? and when did the events reported in the news story happen?

- **WHERE:** Where is the location of the news story?

- **WHY:** What is the motivation for publishing the news? Chronicle, informational, economic, social/political, etc.

The 5W rule, in addition to the analysis of the parameters of presentation of the news, constitute the fundamental elements for the identification of the reliability of a piece of information.

4 APPLIED TECHNOLOGIES

4.1 Web-Scraping

Web scraping refers to the extraction of data from a website. This information is collected and then exported in a format useful for analysis.
The following web scrapers were used for the implementation of "Fake News Detector": Goose3, Scrapy, BeautifulSoap.

4.2 NLP and Feature Elaboration

Thanks to NLP technologies, the prototype built for the detection of fake news is able, given an input text, to extract the locations present in it, to allow us to calculate the distance between the place of occurrence of the facts of the news and the place of the person who is making the search. To do this, some libraries and mathematical formulas have been used such as: spaCy, Geopy.

4.3 RPA

Robotic process automation (RPA) is the technology that allows anyone today to configure computer software, or a "robot" to emulate and integrate the actions of a human interacting within digital systems to execute a business process. RPA robots utilize the user interface to capture data and manipulate applications just like humans do. They interpret, trigger responses and communicate with other systems in order to perform on a vast variety of repetitive tasks. Only substantially better: an RPA software robot never sleeps and makes zero mistakes. (UiPath, 2020)
For this tool is used Selenium WebDriver for RPA.

5 THE PROJECT: FAKE NEWS DETECTOR

5.1 Requirement Specification

Fake News Detector has as objective the definition of an automatic tool for the detection of Fake News. Unlike the projects mentioned above, we tried to focus and involve the user as much as possible without working only on the source of the news, trying to develop his awareness. The strong point of the tool is the automation of the 5W theory, to which two How are added. This can be summarized in the following points:

- **WHO:** this point aims to verify the presence or absence of the author of the news. It wants to answer the question: "Is there an author for the news item to be analyzed?". Based on the answer, the tool assigns a 0/1 score as follows:

  a. [**0**] The author of the news item is not present.

  b. [**1**] The news item is attributable to an explicit author.

  Generally, a Fake news, does not present an author.

- **WHAT:** this point aims to search for the topic of the news. Based on the type of topic, the tool assigns a score from 0 to 1 as follows:

  a. [**0**] It is not possible to classify the topic of the news item.

  b. [**0.5**] Dogma, maxim, speech, political propaganda, commercial propaganda, satire, provocation

  c. [**1**] Fact, scientific publication, news story.

  A news story considered reliable is most often attributable to a well-defined topic.

- **WHEN:** this point aims to identify the date of occurrence of the events that took place in the news story. The tool assigns a score from 0 to 1 in the following way:

  a. [**0**] If there is no reference date of the news item
  b. [**1**] If present in explicit form

  A fake news story generally does not have precise dates. Therefore, it is only possible to give a high score to a news item if the date is present in an explicit form.

  In addition to the above points, the temporal proximity to the publication date is also added as a factor in determining the When. The following score is assigned:

  a. [**0**] If the facts in the news item refer to a period prior to 180 days (approximately 6 months) from submission to the tool.
  b. [**0.15**] If the facts of the news item refer to a period before 2 years since submission to the tool
  c. [**0.35**] If the facts in the news item refer to a period earlier than 2 years after submission to the tool

  Another very important feature when evaluating a news item is the date of occurrence of the reported facts. Tendentially, more the date of occurrence of the facts is close to today's date, more increases the probability that it is a fake news constructed ad hoc to deceive the reader. The reader of the news tends to attribute to it a greater credibility if the facts present in it have happened in a relatively close period, since the subject is current; while, the more we move away from today's date, the more the probability that the news is a fake news decreases (but even if it were, it would no longer have the same media effect of a recent news story).

- **WHERE:** this point aims to identify the place where the events took place and attribute the proximity to the place where the person who intends to carry out the verification resides. The tool assigns a score from 0 to 1 in the following way:

  a. [**0**] If it does not exist or is not possible to identify the place of occurrence of the facts
  b. [**1**] If it is possible to attribute the place of occurrence of the facts

  A news considered reliable will have a well defined place of occurrence of the facts

  In addition, an additional score will be assigned according to the following parameters:

  a. [**0**] If the news is distant less than 1650 KM

b. [**0.15**] If the news is at medium distance from the place where the verifier resides (distance between 1650 KM and 6000 KM)
c. [**0.35**] If the news is at long range from the place where the verifier resides (distance greater than 6000 KM).

is important to analyze the distance between the person searching for the news and the place where the news occurred. In general, the closer the distance of occurrence of the facts is to the place of residence of the person doing the research, the greater the probability that it is fake news. The person reading it will have more interest in giving importance to a news story that takes place in a place relatively close to his own

- **WHY:** this point aims to research the purpose for which the news story was published. It intends to answer the question, "Why did the person who published or shared the news do so? What purpose did it achieve? What emotions did it evoke in me?" Based on the type of answer, the tool assigns a score from 0 to 1 as follows:

  a. [**0**] If the purpose of the person who published the news item is to arouse emotions in the reader or to induce them to buy a good/do an activity
  b. [**1**] If the purpose of the publisher is to inform in a disinterested manner

If a news item arouses strong emotions in the reader (fear, anger, dismay, etc.) or causes him to change his idea/opinion or to perform an action that he would not have done before, then the news item will most likely be a fake news item created with a very specific purpose. Generally, a real news that has the sole purpose of informing the reader, should arouse in the one who reads it a neutral feeling.

The automation of checks follows the following solution:

- **WHO:** A web scraper is used to verify the existence of the author with the goal of extracting the same from the xml code of the page.

- **WHAT:** To understand the topic of the news, the tool asks the user to select one of the options from a specially designed drop-down menu.

- **WHEN:** Through an algorithm of "String Matching" it searches for the date within the news; if this is not explicitly present, through a semantic analyzer it searches for names related to known periods of the year (e.g. Christmas, New Year, Winter, Summer, etc.). Once determined the date, the proximity is calculated and the parameters are assigned as defined above.

- **WHERE:** Using a semantic analyzer trained on a dataset of geographic locations, the presence of the location is extracted. Once extracted, through the Nominatim API, the distance in km from the address declared by the searcher is calculated.

- **WHY:** In order to resolve this point, a specially designed drop-down menu is proposed.

In addition to the parameters provided by the 5W, two other parameters defined as follows will be studied:

- **HOW 1 - Analysis of the relationship between uppercase and lowercase letters.** Through a syntactic analyzer we analyze the relationship between capital and small letters in the title in order to assign a value to be used as an additional parameter for the evaluation of the news. The tool assigns a 0/1 score in the following way:

  a. [**0**] If the news item has a capital/lowercase ratio higher than 10 percent

  b. [**1**] If the news item has a case-sensitive ratio of less than 10 percent

  This is the case of the so-called "Screamed Headline", which has the sole purpose of drawing the reader's attention. A real news does not need to include in the title text a high percentage of capital letters

- **HOW 2 - Analysis of external references to the facts described in the news.** Through the automation of Google search, Fake News Detector proposes to the user a series of links referring to the facts of the news. If the user recognizes a correspondence with the facts in the news itself, then the tool assigns 1 to the news, otherwise it assigns 0.

  a. [**0**] If the news doesn't have any reference besides the site itself

  b. [**1**] If the news has a reference on a different source than the site itself

  A fake news story is generally not reported by other sources but only by the publishing site.

- **HOW3 - Presence of Misleading Images.** Through the automation of the reversa image search of Google it is possible to analyze the images present in the news in order to verify if they are coherent with the news itself. The tool assigns a value included in the interval [0,1] in the following way:

  $$scoreImages = \left(\frac{\sum_{i=1}^{n} scorePar(i)}{n} \right) \text{ where}$$

  – n = total number of images

- $scorPar(i)$ = score assigned to image i

There are many times in which in the news are included images that are not relevant or modified; all this would not make any sense if the news were real.

6 IMPLEMENTATION

In order to proceed with the implementation of the tool "Fake News Detector" it was necessary to set up the development environment as first thing. For the realization of the prototype was used Python version 3.7 because of its compatibility with the libraries described above. Spyder was also used as the primary IDE for writing code; this is because the Spyder Python IDE is provided as the default implementation along with the Anaconda Python distribution.

6.1 Structure of the Tool

The tool is divided into several python files. As you can see in Figure 2, the tool has been split into several Python files so that there are separate functions for each task. The main file of the whole program that inherits the various functions of the tool is called "main.py".

6.2 Main.py

The main.py is called when there is the necessity to start the tool. Once this file is called, the various modules that lead to the implementation of the various use cases are executed in cascade. This file also allows the calculation of the total score of the various news and the probabilistic value of truthfulness, associated with these. To start the tool it's necessary to invoke the main.py from the terminal and enter the following command line: *python main.py*.

6.3 WHO

The purpose of this function is to extract author, title, text and images present in a news item. After the insertion of a url by the user, this is passed to a function named checkURL which checks the format and notifies the user verifier of any problems. After this check, the URL is passed to a scraper named "Goose", which, through the analysis of html and xml code, allows with the functions *"cleaned_text"*, *"title"*, the extraction of text and title in string format.

After the extraction of text and title, all the images in the page of the defined url were downloaded. To
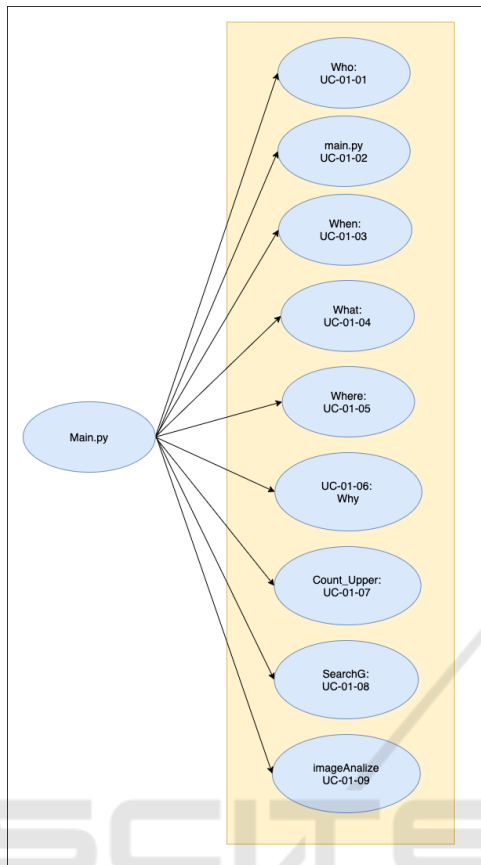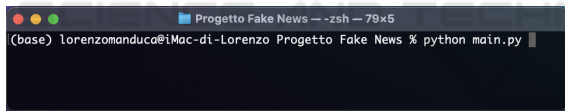
Figure 1: Tree structure of *main.py*.

Figure 2: How to start the tool: *main.py*.

do this we integrated the web scraper *BeautifulSoup* to download images and the Python module *"os"* to manage directories. This part of the code checks if there is a folder named "articleImage" in the program path and if it is empty. If the folder does not exist it is created; otherwise the folder is emptied of files, deleted, and then created again. Once the folder is created, the images present in the "img" tags of the html code relative to the url passed as input are downloaded into it.

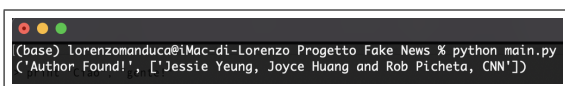The function ends with the extraction of the author and assigns the relative score.

Figure 3: Who - Author detection.

6.4 WHAT and WHY

The goal of this function is to allow the user to select the most suitable topic for the article he submitted to the tool previously. Thanks to the Pandas module, a dataframe containing the news topics is created. Through the inquirerer module, the user is asked to select the most appropriate topic for the facts contained in the article. Following the user's choice, the function ends with the assignment of the score related to the What.

Figure 4: What - Topic Selection.

Also with regard to Why the function aims to allow, thanks to the "inquirer" module, to select an answer to the question shown to the verifying user: *"Does the article would like to convince you to buy some product or to change your opinion about some fact/person?"* In base to the type of answer selected, the function finishes with the assignment of the relative score to the Why

Figure 5: Why - Question.

6.5 WHEN

The objective of this function is to extract dates and eventual time periods from the text extracted in the "main.py" function. First of all, the function eliminates the special characters in the text and then extracts the temporal periods from the text through an NLP classifier based on the temporal expressions contained in the dataset of the JodaTime library and a dataset in the "data" folder of the library containing the main temporal expressions taken from English texts and divided into categories retrievable through settings. This library is built on:

- joda-time: Library for Java date and time classes

- opencsv: Parser library

- JUnit: testing framework

- Log4j: Logging service

- Gson: Json Library for Serialization / Deserialization

The algorithm exploits a dataset of known temporal expressions (included in the library used) and ad-hoc formatted regular expressions for date recognition. Since the extraction of time periods also includes expressions such as "By 20 minutes", "From 12 to 20". etc. it was necessary to set some settings (previously set) that would exclude these time periods for the extraction of only expressions containing dates. These settings, which allowed the exclusion of these time periods, are:

- .excludeRules("durationRule"): excludes time periods like "last two days", "last 30 minutes", etc.

- .excludeRules("repeatedRule"): excludes time periods with repeated event expressions such as "every Sunday at 5 pm", "weekly", etc.

- .excludeRules("timeIntervalRule"): excludes time periods such as "from 19 till 20" and "2:00 pm and 4:00 pm".

After the extraction, thanks to the Pandas module, the extracted dates are inserted into a dataframe. The dates inserted in the dataframe are passed in output to the user through the inquirer module. If more than one date is present within the text, the user is asked to select the most correct date corresponding to the facts of the article; if no date is detected, the user is instead asked to select *"No time Identified"* After the user's choice, the function ends with the assignment of the score related to the When

Figure 6: When - Detecting dates.

6.6 WHERE

The goal of this function is to extract locations or places of interest from the text extracted in "main.py"; this is done thanks to the "Spacy" module, which allows NLP on the text to extract locations or places. SpaCy extracts location features from text documents using Named Entity Recognition (NER). The library applies a classifier based on a dictionary of known locations included in it. Also in this function, the extracted locations are inserted into a dataframe along with their coordinates. To extract the coordinates, the "geopy" module is used to geocode the extracted addresses, cities, places or locations into geographic coordinates. Afterwards is asked to the user, thanks to the "inquirer" module, to select the most appropriate place where the facts of the article took place among

those extracted by the function; if it is not possible to assign a place of occurrence of the facts of the news, the user selects the item "No Location Identified". In addition to the choice input, selected above, the user is asked to enter the address or place from which he intends to verify the facts of the news and geocoded by the tool. Thanks to Haversine's function, the distance between the place where the facts of the news happened and the place of the verifying user is calculated; the calculation is made thanks to python's "math" library. The function ends with the assignment of the score related to the Where.

Figure 7: Where - Select extracted location and insert address.

6.7 HOW 1

The goal of this function is to calculate the ratio between the number of capital letters and the total number of letters in the title extracted and normalized in the "main.py". The function counts the total letters and the capital letters and calculates the ratio. The function finally ends with the assignment of the score related to the count_upper.

Figure 8: Count Upper - Score.

6.8 HOW 2

The objective of this function is to simulate a search on the Google search engine, using the title extracted at the beginning, to establish whether or not there are results similar to the news previously given as input. To do this, the "googlesearch" module is used, to which the title is passed as input; the "BeautifulSoup" web scraper is then used to extract the results (title and link) from the search page. Thanks to the Pandas module a dataset is created containing the links and the titles previously extracted by the function. Also in this case the user is asked to select the title and the link most similar to the news submitted by him in the use case UC-01-01; in case no news is found or not similar to the one submitted by the verifying user, he can select the item "No Link identified", "No Title Identified". The function ends with the assignment of the score related to searchGoogle.

Figure 9: SearchGoogle.

6.9 HOW 3

The goal of this function is to simulate an image search on the Google search engine to check if any images in the article or news are relevant or not to the facts contained in it. The images taken as input are those downloaded in the folder "articleImage" from the "main.py" As first step it is checked if in the folder "articleImage" there are images or not; this is done thanks to the "os" module of Python. Next, the images are encoded and the url is generated for searching. Thanks to the library "Selenium", which instantiates a web browser "Chrome" in the background, is passed the url of the previous point and are extracted the information about the search by url of the images contained in the article. Also in this function the "Pandas" module is used to create a dataset containing links and titles related to the image search. For each image the information about the related Google search is extracted using the tag "fKDtNb". This is because Google uses its artificial intelligence algorithms that try to recognize the content of images and associate certain keywords with them for searches on their content. If results are found, the user is asked to indicate whether or not the topic of the article or news item relates to the related search value extracted from the image. If the answer is yes, the function updates the partial score of the images by adding the value 1 and adds the value 1 also to the image count. Moreover, in case of negative answer, the function extracts the links related to the image from the search and asks the user to select one if related to the facts of the news; if there is no related link, the user will select "No Link identified". If a link is selected, the function updates the partial image score by summing the value 1 and adds the value 1 also to the image count; if no link is selected, the function updates the partial image score by summing the value 0 and adds the value 1 to the image count. This process is repeated for each image in the "articleImage" folder. The function ends with the assignment of the score relative to imageAnalize

6.10 Total Score

As described at the beginning of the chapter, the calculation of the total score of the various news items and the probabilistic value of the truthfulness associated with them is carried out in the "main.py" function. At the end of each function that is called by the

method, it adds the score obtained by the single function and updates the variable called *score_tot* which contains the value of the total score obtained by the news.

7 TEST

In order to test the tool and calibrate a threshold of accuracy for Fake News identification, a dataset containing 1120 news links of different topics was created and labeled as described below:

- 1-80 News item Fake
- 81-160 Scientific Publication Fake
- 161-240 Dogma Fake
- 241-320 Maximum Phrase Fake
- 321-400 Political Propaganda Fake
- 401-480 Commercial propaganda Fake
- 481-560 Satire Fake
- 561-640 News item Real
- 641-720 Scientific Publication Real
- 721-800 Dogma Real
- 801-880 Maximum Phrase Real
- 881-960 Political Propaganda Real
- 961-1040 Commercial propaganda Real
- 1041-1120 Satire Real

Following the application of the tool on the links the tool detected the following values for the news:
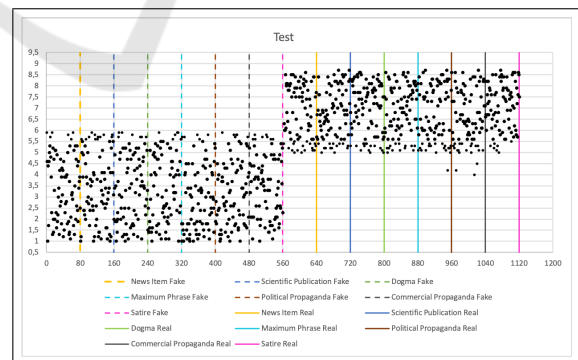

Figure 10: Initial Evaluation of the sample.

Analyzing the graph obtained, the maximum accuracy is obtained calibrating the Threshold value at 5,5 as in the following picture:

As we can see from the graph in Figure 11 we can see that even changing the topic of the news the proposed threshold remains the same without impacting the accuracy of the system. After the calibration the tool

Figure 11: Results from the Test.

was tested on a further sample of 50 random news and reported an accuracy of 97.73%. It is worth remembering once again that the objective of our project is not to distinguish with certainty a fake news from a real one, but to signal to the user the degree of "dangerousness" of the news itself in order to solicit a critical examination. From a careful analysis of the results it is possible to notice that the score relative to the Who presents some zero values on news that in reality present an author. This happens because the author of the news is present in the article but is not correctly inserted in the "author" field of the html code relative to the url. As far as the extraction of the locations is concerned, the Where function sometimes presents difficulties in identifying the correct location because, when translating into English, the name of the location could be translated incorrectly. For example, if one considers locations with compound names such as "Testa di Lepre" and "Porte di Roma", they may be translated as "Hare Head" and "Gates of Rome", which prevents the tool from finding the correct location. If consider the score related to the search for similar news on other online sources, this often takes the value 1 as the news is found on social networks or other sites that represent a different source from the original one but publish the same content. While for the image score this is closely related to Google's artificial intelligence algorithm which is often a source of ambiguity. For example, a photo of a character dressed in a smart suit (which is consistent with the content of the news) is understood and analyzed by Google as "Formal Wear" creating ambiguity in the recognition of the image and therefore lowering the score of the news. In any case, for the purposes of the system, these uncertainties are perfectly acceptable, and could be the subject of further developments and improvements of the software in the future.

8 CONCLUSION AND FUTURE DEVELOPMENTS

The next release of the tool will verify the credibility of the author by searching in the main search engines for further works of the same author in order to evaluate his reliability. It is necessary to underline that an author with a "pen name" is not necessarily a writer of fake news. In fact, many authors considered reliable use pen names (e.g. Joseph Conrad) and these should be credited by the tool as authors with high reputation. Currently, reader interaction is almost completely automated as they only have to answer a few multiple choice questions. In the next version of the tool, which will be released soon, a user-friendly GUI will be created and also the disambiguation, which is currently manual, will be automated.

The following are additional criteria/developments that would improve the accuracy of our tool:

Improving Instrument Accuracy:

- Studying the difference in points between the font sizes of the title and the text adding another parameter to identify the reliability of a news item. (A fake news generally has a much larger font in the title than in the text).

- Dictionary of emphatic lexical forms for the analysis of the presence of these within a news story. Generally a fake news, to convince, uses a lot of them.

- Web scraper to automate the reverse image of Bing that seems to have a higher degree of accuracy than Google in identifying subjects in images.

- Study of an algorithm that notifies the number of special characters in a news headline. A fake news, generally presents a very high number of special characters in the title.

Browser Plug-In:

- An extension for Web Browser that allows the user to calculate the degree of reliability of the news directly within the visited page, at each own request.

REFERENCES

Adams, N. (2016). Textthresher software.

Agrawal, P., Parwat Singh, A., and Peri, S. (2020). *DeHiDe: Deep Learning-based Hybrid Model to Detect Fake News using Blockchain.*

Aldwairi, M. and Alwahedi, A. (2018). *Detecting Fake News in Social Media Networks.*

AZhou, X., Jain, A., Phoha, V., and Zafarani, R. (2019). *Fake News Early Detection: A Theory-driven Model*.

Copy Editing, I. (2008). *Five More Ws for Good Journalism*.

Demestichas, K. (2018). *SocialTruth*.

Hart, G. (2012). *The Five Ws of Online Help*.

Khattar, D., Singh, J., Gupta, M., and Varma, V. (2019). *MVAE: Multimodal Variational Autoencoder for Fake News Detection*.

MediaSmarts (2012). *Deconstructing Web Pages of Cyberspace*.

Pettus, J. (2018). Fiskkit.

Shu, K., Cui, L., Wang, S., Lee, D., and Liu, H. (2019a). *DEFEND: Explainable Fake News Detection*. Association for Computing Machinery, New York, NY, USA,.

Shu, K., Mahudeswaran, D., and Liu, H. (2019b). Fakenewstracker: a tool for fake news collection, detection, and visualization.

Shu, K., Mahudeswaran, D., Wang, S., Lee, D., and Liu, H. (2019c). *akeNewsNet: A Data Repository with News Content, Social Context and Spatialtemporal Information for Studying Fake News on Social Media*.

UiPath (2020).