International Journal of Latest Trends in Engineering and Technology
Vol.(15)Issue(4), pp.001-007
DOI: http://dx.doi.org/10.21172/1.154.01
e-ISSN:2278-621X

ANDES A NEW FAKE NEWS DETECTION SYSTEM

Giacomo Abbattista[1], Vito Nicola Convertini[2], Vincenzo Gattulli[3], Lucia Sarcinella[4]

Abstract- In this study we present a software, called ANDES (fAke News DEtection System) able to distinguish fake from true news. ANDES is based on the idea of using two independent models for classification, based on different perspectives: domains and news. For the validation of the news is used a matrix term-documents with function of weight TF- IDF, Support Vector Machine for the classification and the Stochastic Descent of the Gradient to form the model. In the classification with respect to the second perspective, a Bayesian classifier will be used, on a set of characteristics taken from the domain. The data used in the tests, have been obtained through a new scraping system, which uses instances of the browser Google Chrome.
Keywords – Fake news, fake news detection, weight TF- IDF, Support Vector Machine, Stochastic Descent of the Gradient

1. INTRODUCTION

In the last decade the problem of misinformation has become so important that it has become necessary to coin the phrase 'fake news'; designating that information, which in part or in full, does not correspond to the truth, divulged mainly through the Web. Web sites that publish fake news try to be perceived as legitimate and reliable. Fake content is usually distributed with the intent to deceive to harm an agency, entity or person or for profit-making purposes only. The topic has gained particular interest especially in recent years, after the Cambridge Analytics scandal with the misuse of data from about 85,000 Facebook accounts, used to convey false news for political purposes [1][2].

At the moment there is no standard or any software system that acts as a filter to effectively counteract the proliferation of fake news.

The main objective of this study is to lay the foundations for a software system capable of distinguishing reliable from false news. The idea is to exploit two independent classification models, based on different perspectives: domain and news. Such a software can count on the evaluation of an entire domain or of the single news, also allowing to combine the two evaluations and increase the reliability of the final result, according to the "philosophy": a news coming from a site considered not reliable will most likely be fake or vice versa, a site with a high number of news (considered) fake, probably will not be reliable. In this work is been evaluated the ML algorithms and the characteristics to be evaluated necessary to the two classifications, paying particular attention to the number and type of features to be considered. For the validation of the news, referring to a study present in the literature, will be used a matrix terminal-documents with function of weight TF-IDF, Support Vector Machine for the classification and the Stochastic Descent of the Gradient to form the model. In the classification with respect to the second perspective, a Bayesian classifier has been used, on a set of characteristics taken
from the domain, such as: date of creation of the site, number of published news, average of published news in a year, number of authors, number of topics etc.

The data used in the tests were obtained through a scraping system that uses instances of the browser Google Chrome.

2. THE STATE OF THE ART

Despite the technological progress, we have more and more performing machines and more and more valid artificial intelligence algorithms at our disposal, today we have no technological solution, without the periodic human intervention, that can be considered reliable to counteract the diffusion of fake news. In fact, the best solutions are still the fact-checking sites, where specialized human operators, called debunkers, check the information. Despite this strategy having a relatively optimal level of reliability, the end result is often disappointing: once the fake news is denied, it has already been read and shared, frustrating part of the efforts to refute them.

TrustProject [5], an initiative born at the University of Santa Clara (California), is a consortium of journalistic organizations that work together to define a standard of transparency in journalism with the aim of creating a more reliable and transparent freedom of the press. This project also involves companies such as Google, Facebook and Microsoft. The guiding principles on which the project is based, as reported on the official website [6].

In the future this project will most likely offer great advantages in assessing the truthfulness of the information, but currently it is not yet complete and little widespread, which greatly reduces the current scope of the project. Also, the Italian State, in view of the 2018 parliamentary elections, has developed, with the joint work of the Ministry of the Interior and the Postal Police, a tool on the website https://www.commissariatodips.it/ that allows users to report fake news directly to law enforcement agencies. The tool, called Red Button [7], has been developed by the Centro anticrimine informatico to try to

[1,2,3,4] University of Bari Aldo Moro, Bari, Italy

limit fake news during the election campaign period. The function of this tool is very simple: the user who claims to have found a fake news, can connect to the site and report it through the appropriate form, then the experts of the National Anti-Crime Center for the Protection of Critical Infrastructures (CNAIPIC), will analyze the information and sources to be able to assess the truthfulness. In the case of fake news, the Commissariat of PS Online will publish the verdict on the official channels and will also report the content to social media.
In addition, on the web it's possible to find solutions produced by professionals or technology enthusiasts, proposed as add-ons for the browser. Once added to your browser, their operation is simple enough: if the extension should detect a visit to a news page, will be used a visual message to indicate to the user if the news is true, false or not verificable. Here are the most relevant solutions:

[1] Fake News Blocker [8]; its strong point is the user, who is made an active component in the evaluation of information. The feedback obtained is used to draw up lists of unreliable sites/news;
[2] Official Media Bias Fact Check Icon [9]; as the first solution, only that the feedback is acquired from fact-checking sites;
[3] Unpartial Truthiness Analyzer (Fake News) [10]; tool designed by Recognant to evaluate articles using Natural Language Under-standing algorithms. The evaluation takes into account the syntactic and semantic structure of the article. This includes an assessment of grammatical correctness, analysis of prejudices/affirmations of the author and density of facts. The same vendor states that the solution is not very effective in the case of fake news written using journalistic conventions.

In addition, in the literature there are a significant number of studies on the effectiveness of machine learning algorithms in identifying fa-ke news. A study published during the SCOReD (Student Conference on Research and Development) [11] is particularly interesting. The author using a corpus of 11000 news, pre-labeled (reliable/not at-tendible) and published by Signal Media, studies and analyses the performance of various models trained on three series of features:

- bigram tf-idf (term frequency-inverse document frequency)
- frequency of syntactic dependencies
- union of the two

In the study, the best performance is provided by the SGD, applied on a set of features extracted with TF-IDF. ANDES is based on a multi-level validation system for webnews, which allows you to classify both individual news and associated domains on the basis of the data collected by them. Given the excellent results found in the literature, the classification process associated with webnews will be entrusted to the SGD/TF-IDF pair. ANDES aims to evaluate the basis for an autonomous validation system that results:

- Rapid: a news item must be able to be evaluated almost immediately, so that the user can be notified in time.
- Reliable: the evaluation process must yield results with a relatively small degree of uncertainty.
- Usable: the instrument must be quick to consult and, above all, easy to use. A relatively easy way. To achieve this, it is simple to administer the evaluations through a REST API service, in order to facilitate their implementation on multiple platforms.

The reliability and speed of the system could be two characteristics in contrast with each other. In order to increase the first, additional feedback systems should be integrated that allow the network to straighten the shot, which would imply more workload and a possible decrease in the speed of the evaluation process.

3. SYSTEM ARCHITECTURE

3.1 Language and db

ANDES has been fully developed in Python3[12]. Of particular interest is Cython, a superset of Python in order to convert code written in Python into extension modules compiled in C.
To reduce the complexity of the software system, ANDES is divided into two sub-systems: one for data collection and one for data processing and classification. The first sub-system, the crawler, has been designed and implemented in order to obtain a navigation similar to that of a user. At the domain level, a unique SQLite3 dabatase has been associated, in order to simplify debugging operations and have greater control over the extracted content.

3.2 The data collection system

The libraries used in the scraping process have been: Selenium [15], is a framework that provides tools for testing web applications, Newspaper [16] a python library for the extraction and manipulation of webnews, AdblockParser [17]: a python library to filter the advertisements present in a web page. Whois[18]: is a network protocol that allows, through queries, to obtain information about a given IP address or a specific url, Apsw (Another Python SQLite wrapper)[19]: is a SQLite3 wrapper for Python. Compared to the standard language library, it provides more control over database operations.

In particular:

- Crawling_tools is the central part and manages and synchronizes all parts of the system. At the forefront is the CrawlerManager, which is the class that handles multiple instances of the Crawler class simultaneously. The CrawlerManager assigns a single domain to visit and process to each Crawler. In particular, the crawler object visits (using the browsing_tools package) each url of the domain, from which it extracts the necessary information (using the extraction_tools and newspaperlite packages) and then stores it on disk (storage_tools). The visit and data collection, in each crawler, are performed in parallel. In particular, each crawler is assigned a maximum number of instances that can be used simultaneously to visit the web pages. Each crawler is assigned a maximum number of instances that can be used simultaneously to visit the web pages.
- browsing_tools: the package offers the tools to visit the pages of a domain. The package is based on selenium and chromedriver in headless mode. The data collected by the module are the same as those found in a modern web browser:
- Verification of the http response code
- Canonicalurl
- Check for redirections and retrieval of the url.
- Requests made by the page during loading. This information is mainly retrieved in order to identify the uploaded advertisements.
- Errors returned from the page
- HTML page

For each instance of the Crawler class, one or more objects of the CustomWebBrowser class are instantiated and managed, which is the one that handles the download of the page given a url.

- extraction_tools: once the html of the page (browsing_tools) is obtained, this module is used to extract the requested information. In particular we find the class ContentExctrator, which deals with:
- Complete the relative urls inside the DOM
- Extract and save the urls referring to other pages of the same domain
- Removal of the navigation content, using the implementation of an algorithm found in the literature (NavigationContentHunter)
- Check, using OpenGraph [20] and the schemes provided by Schema.org [21], if a certain page is a webnew
- extract the contents of the web news present in the page

In addition, the module provides classes to extract rss flows from a domain (RssFinder) and to identify advertising within the page (AdsExtractor).

- newspaperlite: is a fork of newspaper (available on Github), Python library for the extraction and the care of webnews. The library has been modified for reasons of optimization. In fact, some tasks are already managed previously by the extraction_tools module, such as downloading the DOM and converting it into an inspectable object (using the lxml library) to perform searches and cleaning. The following classes are defined in the fork:
- Article: coordinates the extraction tasks inside the library. Once all the elements have been extracted (Extractor), the class, before returning an object containing all the information obtained, performs a cleaning task on the text of the article (OutputFormatter).
- Extractor: the class provides methods to extract all the contents of a news item from the DOM. Among the extracted data we have title, text, authors, main image, video and publication date. For each information to be extracted a method is provided. In addition, each method offers more than one extraction strategy, so that if one fails, the next can be used. The position of a strategy is given by its accuracy and consequently the results of the first strategies are expected to be better than those of the following ones. Usually the optimal results are had when a domain/page uses the standards OpenGraph or microdata.
- OutputFormatter: the class deals with formatting the text for a correct visualization. In addition, the last cleaning on the text to remove possible html tags, special characters or excessive spaces.
- storage_tools: the module not only stores the extracted data, but also manages the list of urls to be forwarded to the crawler. In fact, the crawler requires a new list of urls, only once has finished processing the previously requested ones. In addition, the extracted data is passed from the Crawler to the DataCollector, which places it in a queue, waiting to be saved on disk, to be stored. The module provides two classes:
- AbstractDataCollector: as the name suggests, it is an abstract class. The class manages the whole part concerning the preparation of queries and data, leaving not implemented all that concerns the connection to the database, transactions and commits.
- DataCollector: subclass of AbstactDataCollector. The class uses a Sqlite3 database. Data extracted from the Crawler is passed to an instance of this class which inserts it into a queue object. For a matter of time optimization, inside it there is also an object (of the class AdsExctractor) that analyzes the requests of each page processed to be able to

identify the advertisements. This last task is executed on a different thread from the one that manages the DataCollector object.

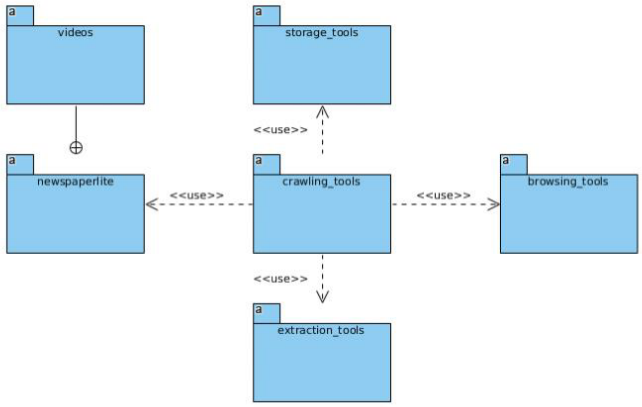

Figure 1 Packages diagram

4. UNITS

Use either SI (MKS) or CGS as primary units. (SI units are strongly encouraged.) English units may be used as secondary units (in parentheses). This applies to papers in data storage. For example, write “15 Gb/cm2 (100 Gb/in2).” An exception is when English units are used as identifiers in trade, such as “3½-in disk drive.” Avoid combining SI and CGS units, such as current in amperes and magnetic field in oersteds. This often leads to confusion because equations do not balance dimensionally. If you must use mixed units, clearly state the units for each quantity in an equation.

The SI unit for magnetic field strength H is A/m. However, if you wish to use units of T, either refer to magnetic flux density B or magnetic field strength symbolized as μ0H. Use the center dot to separate compound units, e.g., “A·m2.”.

5. EXPERIMENTATION

The data used in the tests have been retrieved from the system described previously, from a list of known/untrusted domains. In total the dataset will consist of 1,424,333 web pages from 36 different domains, of which 803,587 are news with at least 500 characters in the text. Below is the list of sites used for testing: Trusted sites: www.lagazzettadelmezzogiorno.it, www.ilfattoquotidiano.it, www.ansa.it, www.corriere.it, www.tgcom24.mediaset.it, www.ilsole24ore.com, www.ilmessaggero.it; www.agi.it; www.lastampa.it; www.repubblica.it, www.ilmattino.it, www.lavocedimanduria.it, www.italiaoggi.it; www.ilgiorno.it, www.ilgiornale.it, www.liberoquotidiano.it, www.iltempo.it, www.ilmanifesto.it. Unreliable sites: googl.al, newsepericolose.blogspot.it, www.corrieredelcorsaro.it, www.ilcorriere.cloud, devinformarti.blogspot.it, www.gazzettadellasera.com; lastella.altervista.org, www.ilfattoquotidaino.it, www.direttanews.it, italianosveglia.com; autismovaccini.org, informatitalia.blogspot.it, internapoli.it, pianetablunews.it, www.segnidalcielo.it, vacciniinforma.it, www.ilmessaggio.it, www.news24europa.com; The data obtained from these domains are subjected to a process of processing/cleaning before it can be used in the classification. The information thus obtained is used to construct two types of datasets necessary for the two types of classification. In both cases, the dataset is divided into two subsets, the training set and the testing set, which are respectively used to find patterns in the data and verify the accuracy of the generated model in estimating the reliability of a site/domain.

5.1 Libraries used in the cleaning/evaluation process

The main libraries that will be used in the evaluation process are:

- Scikit-learn: open source library for automatic learning. It contains regression algorithms, classification and clustering. The library is implemented so that it can interact with other libraries for Data Science.
- TextBlob: text processing library (NLP). The library, in the system, is used exclusively for the identification of the language in the text.
- Pandas: open source library that provides data structures for data manipulation and analysis (dataset). The functionalities offered are:
    - an object that encapsulates a dataset and allows its manipulation and indexing, called DataFrame.
    - Methods for reading and writing data on different data structures and file types.
    - Possibility to interact directly on the missing data of the collection.
    - Merge and data structure split. It is also possible to group the data with respect to a field of the dataset.
    - Indexing with respect to a field. This allows to operate on large datasets through a subset of the same.

Moreover, the library has been partially rewritten in Cython for optimization purposes.

- Numpy: library for efficient manipulation of large multidimensional matrices. There are also basic functions for working with matrix data.
- NLTK: is a suite of libraries for Natural Language Processing (NLP). It includes algorithms for: classification, tokenisation, stemming, lemmatisation, tagging. It is widely used for teaching and research.

5.2 Features

The news classification has been performend on a set of characteristics extracted from the matrixdocuments with TF-IDF weight function; the process has been by the TfidfVectorizer class of the Scikitlearn library. In this case a document means the text of a news item and consequently we will have on the lines of the matrix all the news recovered from the scraping process and on the lines the extracted terms. In this phase the news will be considered reliable if such is the domain from which they come. Of course, this choice is not always valid, since it is well known that fake content sites tend to copy-paste the news of recognized newspapers.

In the domain classification, the characteristics that will be considered are:

- Age of the dns record: the creation of the DNS record associated with the domain is recovered and a difference is made to get the days passed.
- Time left: using the information associated with the DNS record is evaluated how many days are left until the end of the record.
- Foreign country: Flag that indicates if a domain is associated with a DNS record registered outside Italy.
- Published news: number of published news from the creation of the domain to the day in which the domain is executed.
- Average of the news published daily.
- Average of the news published annually.
- Average of advertisements uploaded per page.
- Number of RSS feeds present in the domain.
- Number of authors associated with the domain.
- Number of sections/topics associated with the domain.
- Number of words identified in the news.
- Number of vulgar words found within the news.

All the characteristics mentioned above will be evaluated for each domain, compared to the pages associated with it.

5.3 Data transformation

Before the desired characteristics can be extracted from the collected data and used for classification, they must be processed/processed. For textual data, the first step is to identify the language used, as the cleaning processes are language-dependent. Once the language is obtained, the following steps are performed on the texts:

- removal of escape characters;
- transformation of the text from uppercase to lowercase; 3) division of the text into tokens (tokenization);
- removal of punctuation;
- removal of stop-words from the text;
- reduction of the flexed form of a word to its root form (stemming)
- Removal of words directly related to the domain, such as: authors' emails, site name.

In addition, during the cleaning phase, for a matter of efficiency, will be calculated the overall vocabulary and the number of vulgar words, whose values will be used in the classification with respect to the domain. Once all the cleaning steps have been completed, the resulting text will be associated with the corresponding news in the dataset, and then used in the next steps. Instead, for numerical data, a normalization (minmax) is made.

5.4 Tests performed on the news

To carry out the tests, all news written in a foreign language and those with repeated text in more than one news item will be discarded: in fact, two or more urls from the same domain, which "carry" identical information content, could be the result of a failure in the extraction process.

The filtered dataset count on 582,450 news from the initial dataset. The termina-document matrix has been built on the "clean" text associated with the news present in the filtered dataset, using the TfidfVectorizer class of the Scikit-learn library. Once the matrix was obtained, it was divided into two parts: training-set and test-set, using respectively 80% and 20% of the elements of the starting matrix. Classification will be done using a Support Vector Machine, combined with a model generated on the training set using the Gradient Stochastic Descent. The creation of the model and the classification will be entrusted to the SGDClassifier class of the Scikit-learn library.

The tests is been performed using the elements present in the test-set with different configurations for the extraction of features from the corpus. The parameters to be tested will be:

- The minimum and maximum threshold of the frequency of occurrence of a term within the corpus (TfidfVectorizer): a term must appear at least in N (min-df) documents and in less than M (max-df), where N and M are percentage

values with respect to the number of documents in the corpus. Changing the threshold values it is possible to exclude the rare terms, which could correspond to typing errors, and the very frequent ones that have n added value in the classification.

- Length n-gram (TfidfVectorizer): the length of the subsequence will be bound by a minimum and a maximum. To for example if the threshold values are given by 1 and 3 (respectively min and max), in the matrix terms documents the lines will not be indexed exclusively by the unigram (ngram composed of 1 element), but also by subsequences composed of 2 and 3 elements, respectively digramma and trigram. If the thresholds are both equal to 1, we obtain the classic matrix termina documents.
- The above-mentioned procedure has been conceived starting from another study present in the literature, in which we analyze the average performances of some algorithms of machine learning for the detection of fake news [11].

In the configurations used will be indicated in advance. Moreover, min_df (max_df) will be used to denote the minimum (maximum) frequency of occurrence in the generation of the terminologydocuments matrix. Similarly, ngram_min (ngram_max) will denote the minimum (maximum) length of the anagrams to be considered in the text.
In the following table there is the comparation of configurations results:

Table -1 Comparation of configuration results.

	Precision	Recall	f1-score
Configuration 1	75%	74%	73%
Configuration 2	84%	84%	83%
Configuration 3	87%	87%	87%
Configuration 4	87%	87%	87%
Configuration 5	87%	87%	87%
Configuration 6	76%	75%	73%
Configuration 7	85%	84%	84%
Configuration 8	87%	87%	87%
Configuration 9	87%	87%	87%
Configuration 10	87%	87%	87%
Configuration 11	74%	70%	64%
Configuration 12	74%	70%	64%
Configuration 13	80%	78%	77%
Configuration 14	80%	78%	77%
Configuration 15	80%	78%	77%

The results obtained on the different configurations showed that the parameter that most affects the performance of the models is the minimum frequency of occurrence of a term in the documents (min_df). On the other hand, the maximum frequency of occurrence (max_df) has almost no effect on performance, even if a low value drastically reduces the workload. Moreover, with the same reliability of the model, the configurations with unit-length anagrams are more convenient than configurations that include digrams.

5.5 Tests performed on domains

The tests performed on domains are used to investigate if the features Having a small number of domains to be used to assess the goodness of the classification, was used the k-fold cross-validation repeated 3 times with 3 different values of k: 3, 5, 8.

Table -2 Results of k-fold cross-validation, with k=3

Sample	Precision	Recall	F1-score	ROC AUC
Sub-sample 1	86%	100%	92%	92%
Sub-sample 2	100%	100%	100%	100%
Sub-sample 3	62%	83%	71%	92%
Average	83%	94%	88%	94%

Table -3 Results of k-fold cross-validation, with k=5

Sample	Precision	Recall	F1-score	ROC AUC
Sub-sample 1	80%	100%	89%	100%
Sub-sample 2	75%	75%	75%	84%

Sub-sample 3	100%	100%	100%	100%
Sub-sample 4	75%	100%	86%	100%
Sub-sample 5	100%	67%	80%	78%
Average	86%	88%	86%	92%

Table -4 Results of k-fold cross-validation, with k=8

Sample	Precision	Recall	F1-score	ROC AUC
Sub-sample 1	100%	100%	100%	100%
Sub-sample 2	75%	100%	86%	83%
Sub-sample 3	100%	100%	100%	100%
Sub-sample 4	100%	100%	100%	100%
Sub-sample 5	100%	100%	100%	100%
Sub-sample 6	100%	100%	100%	100%
Sub-sample 7	67%	100%	80%	100%
Sub-sample 8	100%	50%	67%	50%
Average	93%	94%	91%	92%

6. CONCLUSION AND FUTURE WORK

The test results are encouraging. A possible future development, in ANDES, is the integration of a plagiarism detection system. For textual classification one could consider introducing a system for extracting semantic content from the news, so as to be able to validate a news with respect to all the reliable members of the network.

7. REFERENCES

[1] Hilary Osborne, Hannah Jane Parkinson, Cambridge Analytica scandal: the biggest revelations so far, 2018
[2] Editorial, Reuters (20 March 2018). "Factbox: Who is Cambridge Analytica and what did it do?". U.S. Retrieved 23 March 2018.
[3] Vosoughi S, Roy D, Aral S (2018) The spread of true and false news online. Science 359:1146-1151
[4] Illing, Sean (16 October 2017). "Cambridge Analytica, the shady data firm that might be a key Trump-Russia link, explained". Vox. Retrieved 24 March 2018.
[5] Trust Project, 2018, https://thetrustproject.org/
[6] Trust Project Faq, 2018, https://thetrustproject.org/faq/
[7] Polizia postale e delle comunicazioni, Red Button, 2018, https://www.commissariatodips.it/collabora/segnala-una-fake-news.html
[8] Floris de Bijl, Fake News Blocker, 2017, https://github.com/Fdebijl/FakeNewsBlocker/
[9] Jeffrey Carl Faden, Official Media Bias Fact Check Icon, 2016, https://jeffreyatw.com/blog/2016/11/check-news-bias-with-a-simple-browser icon/
[10] Unpartial, Unpartial Truthiness Analyzer (Fake News), 2016, http://www.unpartial.com/
[11] Shlok Gilda, Evaluating Machine Learning Algorithms for Fake News Detection, 2017
[12] Guido van Rossum , Python, , https://www.python.org/ 111-115
[13] Bing Liu, Web Data Mining,
[14] Stanford University, Stanford's PCFG, https://nlp.stanford.edu/software/lex- parser.html
[15] SeleniumHQ, Selium Browser Automation, 2018, https://www.seleniumhq.org
[16] Lucas Ou-Yang, newspaper, 2018, https://github.com/codelucas/newspaper
[17] Scrapinghub, Adblock Parser, 2018, https://github.com/scrapinghub/adblockparser
[18] Richard Penman, Python-whois, 2018, https://bitbucket.org/richardpenman/pywhois
[19] Roger Binns, APSW, 2018, https://github.com/rogerbinns/apsw
[20] Facebook, The Open Graph protocol, 2018, http://ogp.me/
[21] Google, Microsoft, Yahoo, Yandex, Schema.org, 2018, https://schema.org/