FAKE NEWS DETECTION

Cognitive, Behavioral and Social Data
2018/2019

Tommaso Azzini, Damiano Clementel, Michal Kowal, Veniamin Pavlov, Diego Pilutti, Kanza Rasheed, Salvatore Romano

1. Abstract

In modern days, the problem of misinformation has risen due to the nature of online news publication. In this scenario, Fake News have become media weapon capable of heavily affecting society and our perception of not only facts and opinions, but also the media itself. Automatic Fake News Detection is a challenging problem, because of its real-world political and social impact.

In this report, we present an ML algorithm to detect Fake News, starting from a dataset of real-news related short statements (on which previous work have been done on automatic detection of fake news focused on fact-checking), explaining all the steps of the process. Finally, we test our results on a more consistent dataset.

2. Introduction

In the last years, social media has changed not only the way we retrieve information, but also the personal opinion about everyday issues. One of the most dangerous problems about this new way to approach information is the quantity, virality and veracity of such information. The massive spread of news in a day makes traditional fact-checking of potential deception information impossible. Fake News has arisen in last years to mislead and hit people and communities with fake informations. A study led by Ipsos Public Affairs for BuzzFeed News in 2016 showed how about 75% of the time American adults are misled by fake news [1].

The New York Times defines Fake News as: “a made-up story with an intention to deceive, often geared toward getting clicks” [2]. Again, in 2016 announced “post-truth” as 2016 World of the Year [3], defining it as: “relating to or denoting circumstances in which objective facts are less influential in shaping public opinion than appeals to emotion and personal belief”.

With respect to the relevance of the problem we started from some theoretical foundations of fake news, to better understand the context and the scope to which present an automatic model for Fake News Detection.

2.1. Psychological Foundations of Fake News

According to this branch of studies Fake news exploit individual psychological vulnerabilities, that can be summarized as:

a) Naive realism: people tend to see their own perception of reality as the only one and superior to others, and
b) Confirmation Bias: consumers prefer to receive information that confirms their existing views.

2.2. Social Foundations of the Fake News Ecosystem

Another perspective of the problem is given by considering the entire news consumption ecosystem and to focus on the of the social dynamics that contribute to the proliferation of fake news, specially in the contemporary scenario were social media and internet communities play a central role in news dissemination and validation. This concepts can be embodied in so called ‘prospect theories’. What is it? Decision making on relative gain and losses compare to current state in which social acceptance, considered as socially safe decisions with respect of shared belief in a community that can be fake, that play a strategic role in the decision-making process of validating or sharing a deceptive belief.

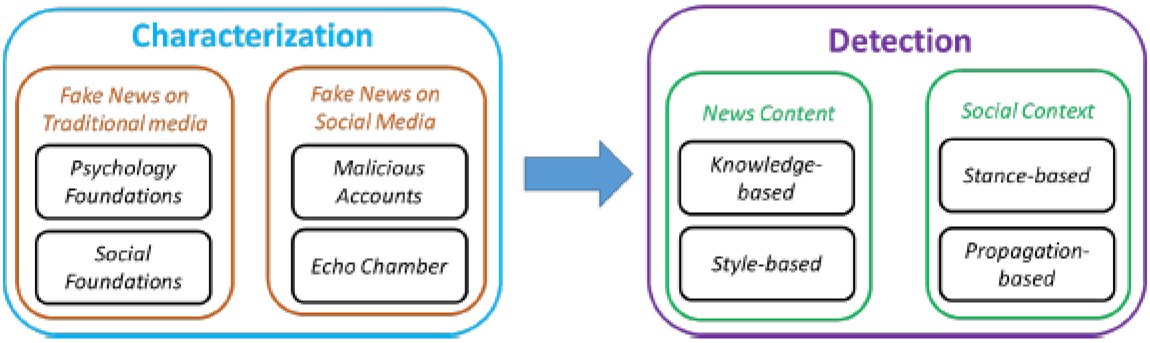


Figure 1: Fake news on social media: from characterization to detection.

3. Problem Description

The aim of this project is to provide a Fake News Detection systems able to assist in detecting and filtering potential deceptive news. In facts, we strongly believe that using state-of-art technologies to automatically detect fake media content can be leveraged to combat this social problem.

It is possible to build a system described above implementing technologies as machine learning and natural language processing. Basically, natural language processing is defined as the manipulation of natural language, in our case text, by software. In other words, it is the ability of a computer program to understand and process human languages.

So, with machine learning techniques we will be able to classify news as “fake” or “real”, that have already been labeled as such by humans. But is NLP the only way to detect Fake News?

Fake news is not just another type of content that circulates online, but that it is precisely the character of this online circulation and reception that makes something into fake news. In this sense, fake news may be considered not just in terms of the form or content of the message, but also in terms of the mediating infrastructures, platforms as and participatory cultures which facilitate its circulation. In addition, the significance of fake news cannot be fully understood apart from its circulation online. [4] In our project we focus on empirically investigate automatic fake news detection based on surface-level linguistic patterns, because of its simplicity of approach that could make it a good technique to assist and give consistent insights in broader fake news detection methods.

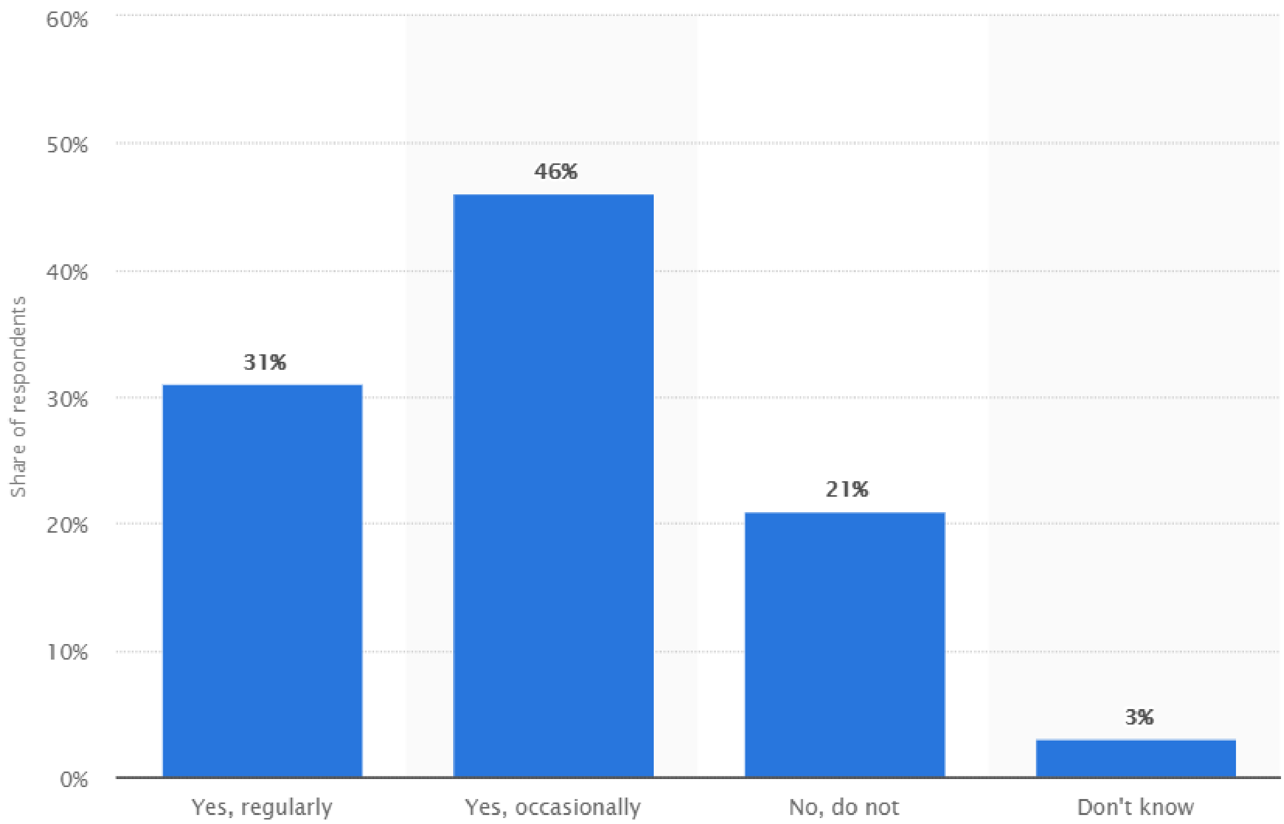


Fig 1:Perceived frequency of traditional major news sources reporting fake news stories in the United States as of March 2018[7]

4. Related Work

As discussed in the previous sections, generally previous works of the literature on fake news detection approach the problem of fake news detection in terms of these paradigms (fake news detection based on knowledge, on context, on style, semantic and network):

a. Knowledge-based fake news detection (also called "fact checking"), is tackled with methods borrowed from information retrieval, semantic web, and linked open data (LOD) research.
b. Context-based fake news detection employs methods from social network analysis where the spread of false information and rumors as well as their containment is studied.
c. Style-based fake news detection relies on computational linguistics and natural language processing, and, more specifically, on methods from deception detection to identify statements at the sentence-level that constitute falsehoods and lies.
d. Semantic analysis extends the n-gram plus syntax model by incorporating profile compatibility features, showing the addition significantly improves classification performance. (Feng & Hirst, 2013). The intuition is that a deceptive writer with no experience with an event or object

(e.g., never visited the hotel in question) may include contradictions or omission of facts present in profiles on similar topics.

e. Network-based approaches such as linked data and social network behaviour use network analysis of inherently structured data network and the use of metadata to analyze the patterns of behavior of questionable sources.

With respect to the scope of our project it is useful to go deeper into the definition of the style-based fake news detection approach given by the literature. Formally the style-based approach has been deployed in different works related to fake news automatic detection and it specifically addresses the problem of capturing the manipulation in writing style of news content:

a) **Deception-oriented**: stylometric methods to capture the deceptive statements, which deployed methods like Advanced Natural Language Processing models, Deep syntax, Rhetorical structure and Deep Network Models.
b) **Objectivity-oriented**: capture style signals that can indicate a decreased objectivity of news content and thus the potential to mislead consumers, such as hyper-partisan styles.

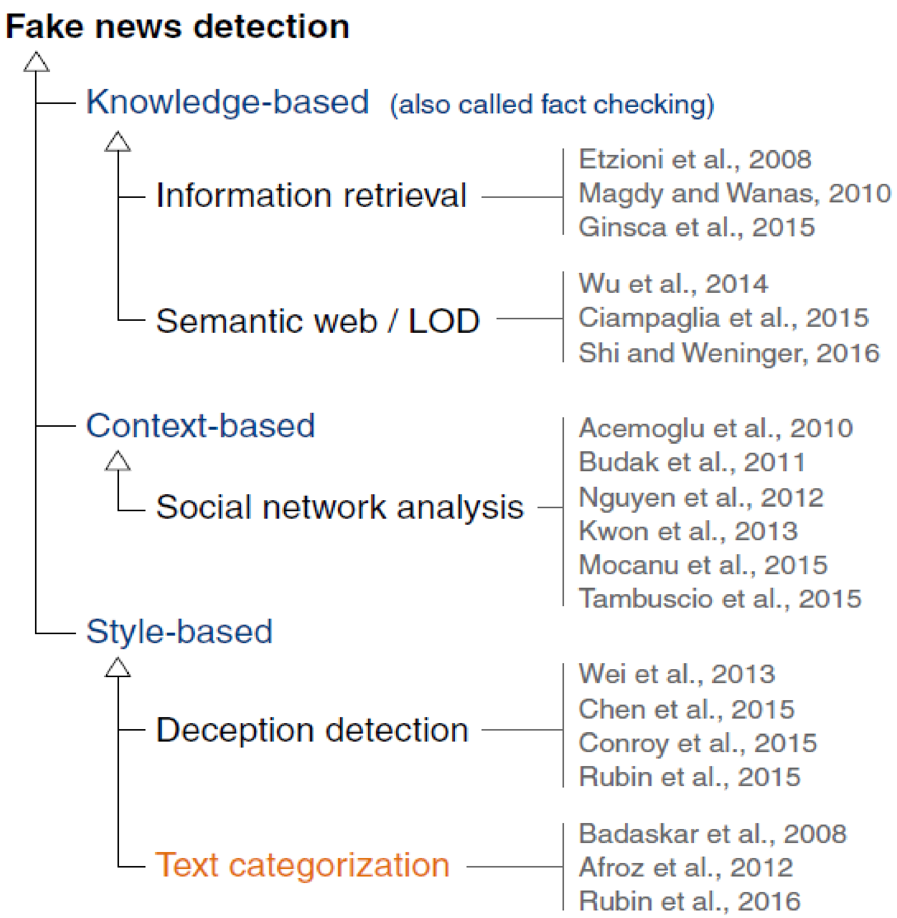


Figure 1: Taxonomy of paradigms for fake news detection alongside a selection of relevant work.

This approaches nevertheless focus on detecting from the writing style of the author the actual intention to produce and disseminate misleading information.

On the other side with respect to this approach we wanted to investigate the correlation between the text features of a heterogeneous corpus of news collected from different internet publishers and its truthfulness, considering linguistic and text features.

This kind of approach have been already used in automatic detection of fake news, usually in correlation to other news-related information, such as content, author, source, and social network feedback and dissemination. From the literature, we can define the linguistic-based features adopted by previous works as:

a. **Lexical features**: total words, characters per word, frequency of large words, unique words.
b. **Syntactic features:** sentence-level features, frequency of function words and phrases (n-grams, bag-of-words approaches), punctuation and parts-of-speech tagging.
c. **Domain-specific linguistic features**: quoted words, external links, number of graphs, average length per graph deceptive cues in writing styles: lying detection features.

Class	Feature name	Posts	Comments	N. of features	Sample
STRUCTURAL	Number of likes/comments/shares	x	-	3	P_1, P_2
	Number of likes/comments on comments	-	x	2	P_1, P_2
	Average likes/comments on comments	-	x	2	P_1, P_2
SEMANTIC	Number of characters	x	x	2	P_1, P_2
	Number of words	x	x	2	P_1, P_2
	Number of sentences	x	x	2	P_1, P_2
	Number of capital letters	x	x	2	P_1, P_2
	Number of punctuation signs	x	x	2	P_1, P_2
	Average word length[a]	x	x	2	P_1, P_2
	Average sentence length[a]	x	x	2	P_1, P_2
	Punctuation rate[b]	x	x	2	P_1, P_2
	Capital letters rate[b]	x	x	2	P_1, P_2
USER-BASED	Av./Std comments to commenters	-	-	2	P_1, P_2
	Av./Std likes to commenters	-	-	2	P_1, P_2
	Mean std likes/comments to commenters	-	-	2	P_1, P_2
	Av./Std comments per user	-	-	2	P_1, P_2
	Av./Std pages per user	-	-	2	P_1, P_2
	Total engaged users[c]	-	-	1	P_1, P_2
	Rate of engaged users[c]	-	-	1	P_1, P_2
SENTIMENT-BASED	Sentiment score	x	-	1	P_1, P_2
	Av./Std comments' sentiment score	-	x	2	P_2
	Rate positive/negative comments	-	-	2	P_2
	Number of positive over negative comments	x	-	1	P_2
	Mean/Std presentation distance	x	-	2	P_1, P_2
	Number/Rate of captivating entities[d]	x	-	2	P_2
	Av. response distance	-	-	1	P_2
PREDICTED	Numb. of pred. D entities[e] (LOG, NN)	-	-	2	P_1, P_2
	Rate of pred. D entities[e] (LOG, NN)	-	-	2	P_1, P_2

The pros of this approaches reside respectively on the functional simplicity and flexibility of application with respect to general news related text data for classification purposes, but such advantages also lead to its biggest shortcoming.

In addition to relying exclusively on language, the method relies on isolated text features, often divorced from useful context information. In this method, any resolution of ambiguous word sense or text related features remains nonexistent. Consequently, many deception detection researchers have found this method useful in tandem with different, complementary analysis.

It is also important to mention that according a research published by Arizona State University in 2017, the features that give higher accuracy on a particular dataset have a direct correlation with the domain and the content of the dataset as table below shows:

Table 1: Comparison of Fake News Detection Datasets.

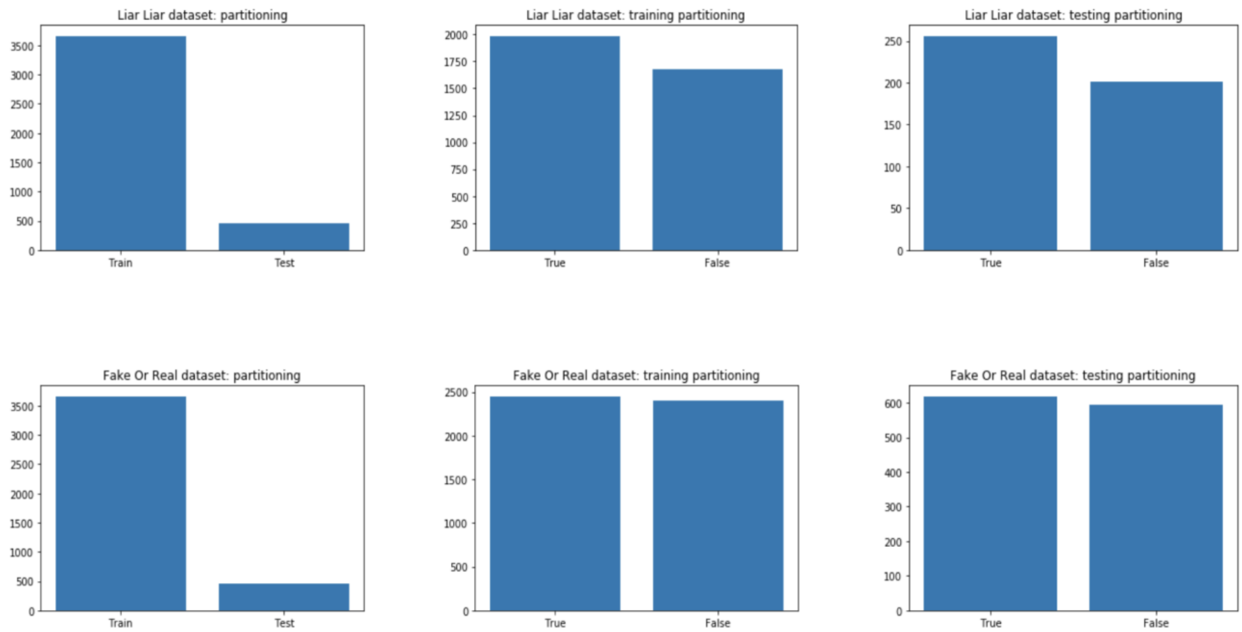
Features / Dataset	News Content		Social Context		
	Linguistic	Visual	User	Post	Network
BuzzFeedNews	✓				
LIAR	✓				
BS Detector	✓				
CREDBANK	✓		✓	✓	✓

5. Datasets

To develop and train our text classification model for real and fake news we worked on two different datasets. We chose these datasets considering the general guidelines defined in previous works of automatic detection of fake news based on text related and linguistic features, that have been summarized below. Dataset corpus should characterize itself with**:**

- availability of both truthful and deceptive instances;
- digital textual format accessibility;
- verifiability of “ground truth”;
- homogeneity in length;
- homogeneity in writing matter;
- predefined timeframe;
- the manner of news delivery (e.g., humor; newsworthiness, believability; absurdity; sensationalism);
- pragmatic concerns (e.g. include copy-right costs, public availability);
- language and culture.

With respect to these guidelines we started from developing a ML model for text classification from the Liar Liar dataset.



5.1 LIAR Dataset

The Liar Liar dataset includes 12.800 human labeled short statements (average words per statement = 17) from POLITIFACT.COM's API5, and each statement is evaluated by a POLITIFACT.COM editor for its truthfulness, structured in six fine-grained labels for the truthfulness ratings: pants-fire, false, barely-true, half-true, mostly-true, and true.

These statements are sampled from various of contexts/venues, and the top categories include news releases, TV/radio interviews, campaign speeches, TV ads, tweets, debates, Facebook posts, etc. To ensure a broad coverage of the topics, there is also a diverse set of subjects discussed by the speakers. The top-10 most discussed subjects in the dataset are economy, healthcare, taxes, federal-budget, education, jobs, state budget, candidates-biography, elections, and immigration.

For the purpose of our work, which aims in developing a ML model to detect fake news based on linguistic and NLP related features of a text source, we will consider only two possible classifications of the dataset rows whose label is 'true' or 'false'.

By selecting these classifications, we obtain a balanced distributed database of 3 671 instances of opposed news, which are 'explicitly' labeled false for 1995 or 1676 for true instances.

Noting that fact-checking is not a classic labeling task in NLP, the challenge of developing a ML model of text classification to detect automatically fake from real news based on linguistic features of the text corpus of the source news, with respect to the limitations of our dataset is:

Can we develop a model for fake news detection from a real-world collection of short statements of heterogeneous subjects that can be consistent with larger datasets taken from a homogeneous topic?

To do so, we extracted and analyzed different NLP features. Then, we tested and compared the performance of different classification models based on accuracy of prediction.

After having explored the best combination of features parameters and the best performance of the different classifiers we generalized our findings testing the model on a larger dataset: Fake or Real dataset.

5.2. Fake or Real dataset

The dataset is a subset of 6059 news comprising of a real and fake part. The fake part has been taken from a larger dataset that contains text and metadata from 244 websites and represents 12,999 posts collected and tagged as real or fake by the BS Detector Chrome Extension by Daniel Sieradski. The majority of the news in this dataset are related to US politics, and the average length of the news are longer (761 words per news on average).

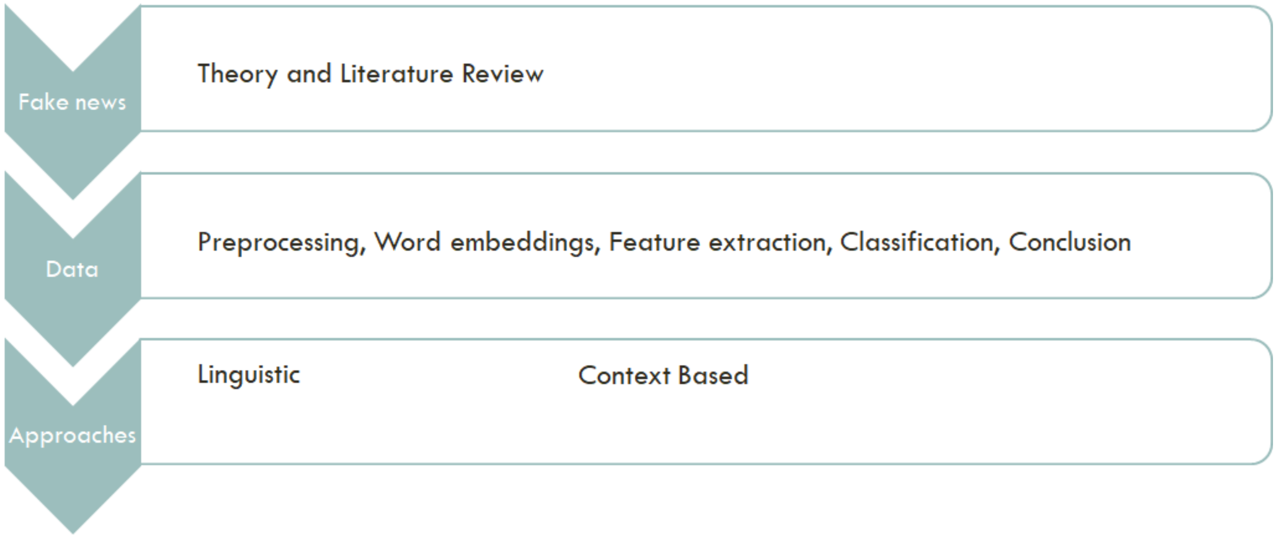
B.S. Detector is a browser extension that searches all links on a given webpage for references to unreliable sources, checking against a manually compiled list of domains. It then provides visual warnings about the presence of questionable links or the browsing of questionable websites. It uses this definition of fake news: "Sources that entirely fabricate information, disseminate deceptive content, or grossly distort actual news reports".

The real part was collected from “All Slides”, a website dedicated to hosting news and opinion articles from across the political spectrum. The articles in the real news dataset came from media organizations such as the New York Times, WSJ, Bloomberg, NPR, and the Guardian that were published in 2015 or 2016.

6. Project Timeline

The figure represents the main steps followed in our project timeline. First couple of weeks were spent in Literature Review and interpreting the previous research done during the past few years related to detection of fake news.

After this, the main steps followed in the code have been mentioned sequentially. As a group, we focused on analyzing linguistic and context based features extracted from the datasets for fake news detection.



7. Feature Extraction

Text Analysis is a major application field for machine learning algorithms. However, the raw data, a sequence of symbols cannot be fed directly to the algorithms themselves as most of them expect numerical feature vectors with a fixed size rather than the raw text documents with variable length. In our work, we analyzed typical common linguistic features extraction approaches, such as:

a) text-frequency based features:

- Bag Of Words (BOW);
- TFIDF;
- n-grams;

b) syntactic features, including sentence-level features, such as frequency of function words and phrases:

- parts-of-speech (POS);
- Word2Vec.

7.1 Text-frequency based features

Bag of Words:

This strategy consists of:

- tokenizing strings and giving an integer id for each possible token, for instance by using white-spaces and punctuation as token separators.
- counting the occurrences of tokens in each document.
- normalizing and weighting with diminishing importance tokens that occur in the majority of samples/documents.

A corpus of documents can thus be represented by a matrix with one row per document and one column per token (e.g. word) occurring in the corpus. Documents are described by word occurrences while completely ignoring the relative position information of the words in the document.

TF-IDF:

In a large text corpus, some words will be very present (e.g. “the”, “a”, “is” in English) hence carrying very little meaningful information about the actual contents of the document. If we were to feed the direct count data directly to a classifier those very frequent terms would shadow the frequencies of rarer yet more interesting terms. In order to re-weight the count features into floating point values suitable for usage by a classifier it is very common to use the tf–idf transform.

Tf means ‘term-frequency’ while tf–idf means ‘term-frequency times inverse document-frequency’: tf-idf(t,d) = tf(t,d) × idf(t).

N-grams:

A collection of unigrams (what bag of words is) cannot capture phrases and multi-word expressions, effectively disregarding any word order dependence. Additionally, the bag of words model doesn’t account for potential misspellings or word derivations. Instead of building a simple collection of unigrams (n=1), one might prefer a collection of bigrams (n=2), where occurrences of pairs of consecutive words are counted. One might alternatively consider a collection of character n-grams, a representation resilient against misspellings and derivations.

While some local positioning information can be preserved by extracting n-grams instead of individual words, bag of words and bag of n-grams destroy most of the inner structure of the document and hence most of the meaning carried by that internal structure. That’s why we try also to deploy syntactic features.

7.2 Syntactic features

Part of Speech:

This strategy consists of:

- tagging - starting from a plain text sentence, process associates to every word and every punctuation sign the corresponding Part of Speech. In this way, it is possible to find out, for example, how many verbs are present in a sentence.
- chunking - given a sentence previously POS-tagged and a grammar, namely a set of rules which connect different POS tags with each other, process retrieves a tree in which previously found tags are related by some other nodes. Those nodes are the relations nodes computed by following the grammar rules.

The rules defined inside the grammar that we used and investigated are the above ones:

a. NP (Noun Phrase) is a sequence of determiner, adjective and a noun.
b. PP (Preposition Phrase) is a noun phrase preceded by a preposition.
c. VP (Verb Phrase) is a sequence made by a verb and its arguments.
d. CLAUSE is a sequence of a noun phrase and a predicate.

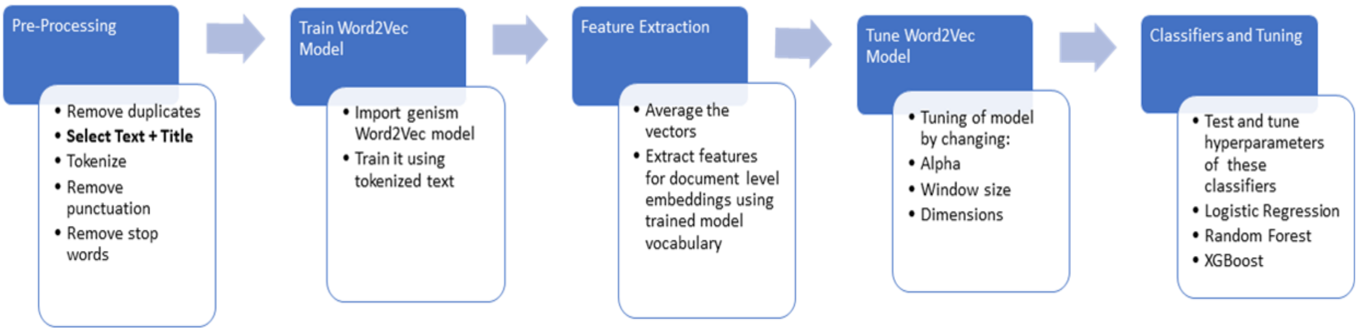
Word2Vec:

This model was created by Google in 2013 and is a predictive deep learning based model to compute and generate high quality, distributed and continuous dense vector representations of words, which capture contextual and semantic similarity. Essentially these are unsupervised models which can take in massive textual corpora, create a vocabulary of possible words and generate dense word embeddings for each word in the vector space representing that vocabulary. Usually you can specify the size of the word embedding vectors and the total number of vectors are essentially the size of the vocabulary. This makes the dimensionality of this dense vector space much lower than the high-dimensional sparse vector space built using traditional Bag of Words models.

There are two different model architectures, which can be leveraged by Word2Vec to create these word-embedding representations:

a) The Continuous Bag of Words (CBOW) Model;
b) The Skip-gram Model.

The steps followed in our project are:



8. ML Text Classification

As a general aim of our project we wanted to develop a model of text classification for fake and real news from a short statement fact-checking dataset and then generalize our results on a bigger and more consistent dataset.

To start our Machine Learning model for automatic detection of fake news we decide to explore different combination of features with a wide range of classification models, such as:

- Decision Trees;
- Support Vector Machines (SVM);
- K-Nearest Neighbors (KNN);
- Multinomial Naive Bayes;
- Random Forest;
- Logistic Regression;

- Convolutional Neural Networks;
- XGBoost.

The use of these classifiers has already been deployed for automatic detection of fake news in early works of automatic detection of fake news with promising results. Apart of them we will use also:

8.1. LSTM

LSTM was developed as a neural network architecture for processing long temporal sequences of data. Other recurrent neural networks trained with variants of back-propagation proved to be ineffective when the input sequences were too long. LSTMs are explicitly designed to avoid the long-term dependency problem. Remembering information for long periods of time is practically their default behavior, not something they struggle to learn.

8.2. GloVe

GloVe is an unsupervised learning algorithm for obtaining vector representations for words. Training is performed on aggregated global word-word co-occurrence statistics from a corpus, and the resulting representations showcase interesting linear substructures of the word vector space.

In this work was used Pre-trained word vector: glove.6B. It was trained on a dataset of one billion tokens (words) with a vocabulary of 400 thousand words. The glove has embedding vector sizes, including 50, 100, 200 and 300 dimensions. For this work was chosen the 100-dimensional version.

Namely we trained our models on the Liar Liar dataset and tried to improve the accuracy score of our classifiers prediction, and then tested again our models on the second Fake or Real dataset. To improve the accuracy of our models we used to main approaches:

1. Selection of best feature parameters to fit our models: in the feature engineering section, we generated a number of different feature vectors, selecting them to improve the accuracy of the classifier.
2. Selection of best hyperparameter in modelling: we implemented different strategies to fine tuning the parameters of our models to get a best fit model.

To do so, we explored different approaches and relied on the free software machine learning library for the Python programming language that features various classification, regression and clustering algorithms: Scikit-Learn.

To expand our explorative analysis, we tried to develop and combine different kind of approaches to improve the accuracy of our models on the Liar Liar dataset and then test them on the second one.

Generally, we used methods such as pipelines for both feature extraction and model selection, in which we applied Grid Search hyperparameter optimization to choose a set of optimal hyperparameters to maximize accuracy for a learning algorithm. This method selected the hyper parameters on a 5-fold cross-validation to estimate performance, based on accuracy.

Moreover, we also applied statistical methods for feature selection in addition to the hyperparameters selection such as the statistical chi-square of independence to determine the best dependency between the features and the predictions of our models. In the next section, we present the best outcomes of our extensive analysis.

9. Conclusions

Features	Classifiers	LIAR Dataset	Fake or Real Dataset
TF-IDF	Decision Tree	58%	78%
	SVM -SVC	61%	93%
	KNN	62%	81%
	MultinomialNB	63%	82%
	Random Forest	63%	85%
	Logistic Regression	63%	91%
BOW → POS	CNN	54% → 57%	75% → 69%
	Logistic Regression	63% → 59%	91% → 80%
	Tree	55% → 53%	78% → 73%
BOW → BOW + POS	CNN	54% → 54%	75% → 76%
	Logistic Regression	63% → 64%	91% → 89%
	Tree	55% → 58%	78% → 81%
Word2Vec	Logistic Regression	55%	92%
	Random Forest	56%	90%
	XGBoost	48%	91%

In the upper table, we presented the best outcomes of our analysis and that we think are more consistent for the aim of our project. Considering the text frequency related features (BOW, TFIDF and n-grams) the general limitations of the corpus of text data of Liar Liar dataset are reflected also on the general outcomes.

Relevant accuracy performance has been reached with the TFIDF feature on average (62%) with slightly, but not general results for all models in accuracy improvements considering also some n-grams patterns. On the other side the BOW performance on average among the classifiers was poor attesting just slightly above the human-guess performance (50%). So, with respect to this approaches we could suggest that best estimations for automatic detection of fake news could be improved by deploying a Logistic Regression Model 63% and Random Forest Model 63% considering the TFIDF features of a text. The consistency of these results is also relevant with respect to the accuracy of these models tested on the Fake or Real dataset that respectively gave 85% and 91%.

Moreover, with respect to the findings of syntactical features (POS, Word2Vec) we had some evidence that it is correlated to a news being fake or real due to the increase of accuracy that we reached when POS was run in addiction to BOW and either alone. This behaviour has been highlighted either in the former and latter datasets with a better performance on the second dataset.

In addition, POS tagging and chunking seems to perform better on sentences which have a lower number of words. However, since POS only approaches have a worst accuracy than BOW, while the combination of the two generally increases accuracy, at least on Liar Liar dataset, it seems that POS tags and chunk capture some information which is not sufficient alone, but meaningful if computed along with other features.

Word2Vec Features gives higher accuracy on fake and real news dataset. The results are much more accurate on this dataset as compared to Liar Liar dataset because this dataset is solely based on political news. The domain is specific and as we know that the Word2Vec model solely depends on the context or the vocabulary of a given dataset i.e. every word vector is computed using the contextual words around it depending on the window size, we can therefore conclude that Word2Vec gives better results on a more specific domain rather than a mixed domain. It reconstructs linguistic contexts of word and is based on the logic that the words that occur in the same contexts tend to have similar meanings. In our project, the Word2Vec model was tuned using different parameters and the best results were given by the Word2Vec model with dimension size = 300, window size = 6 and alpha = 0.04. After training this Word2Vec model, the highest accuracy of 92.05% was achieved with Log Regression on Fake or Real dataset.

In general, our findings relate specifically not just on the method of feature selection and model training and testing for the detection of fake news, but also on the fact that to the performance of such model relates specifically to the nature and quantity of the data. In fact, better accuracy performance of prediction has been reached in the Fake or Real Dataset, which contains more text data and which contents focus on a specific homogeneous topic.

APPENDIX A:
(Pavlov V Approach)
LSTM and LSTM + Glove Results
In this part the possibility of using *Long short-term memory* **(LSTM)** neural networks for fake news detection was investigated. And the second achieved result is research of different datasets and the impact of their features on the accuracy of predictions. This solution was based on the model for detecting fake or real reviews Text Classification using LSTM. [6] As full solution is too huge, this report contains only general explanation and conclusions. The full version can be found in the appendix.

Training datasets:

a. Fake or Real dataset is a table of news and marks if it's real or fake - size: 6335 articles (REAL - 3171, FAKE - 3164); news about different topics;
b. train data from Liar Liar dataset - size: 3671 articles (TRUE - 1676, FALSE - 1995); news about USA politics in general.

Testing sets:

a. Fake or Real testing set is subset of full Fake or Real - size is 1267 articles (20%); news about USA politics in general;
b. test set from Liar Liar dataset - size: 457 articles (REAL - 208, FAKE - 249); news about different topics.

	"self accuracy"	*test* dataset	*train* dataset	*Fake or Real* dataset
LSTM (fake or real news)	0.96	0.53	0.5	x
LSTM + GLOVE (fake or real news)	0.78	0.52	0.53	x
LSTM (train from liar liar)	0.81	0.51	x	0.47
LSTM + GLOVE (train from liar liar)	0.63	0.51	x	0.52

At the step of testing there were used different datasets. Test dataset is specially prepared one for testing train data set and both are from Liar Liar dataset. This table shows the accuracy of both models trained on both data sets. As we can see in general the value of accuracy approximately the same in every case and equal 52-53%. And it means almost complete unsuitability of these approximations.
But attention should be paid to the "self-accuracy" values. These values are accuracy of predictions, which were received as testing at the training dataset. It is obvious that the accuracy of these predictions should be quite high. However, only accuracy of LSTM model trained on Fake or Real dataset high enough. This may mean that only the Fake or Real dataset is balanced enough. Let's try to explore this dataset deeper.

	"self accuracy"	Fake or Real test dataset validation_split = 0.2
LSTM (fake or real news)	0.96	0.87
LSTM + GLOVE (fake or real news)	0.85	0.76

Now the table represents the accuracy of models, which were trained on 80% of Fake or Real dataset. The other 20% was used as a test set. And this time accuracy of prediction is quite high, close to 85%. The results can show that for good quality of predictions is important to have enough homogeneous dataset. This means at least that news topics should be about the same.

10. References

[1] "A FIELD GUIDE TO FAKE NEWS" compiled by Liliana Bounegru, Jonathan Gray, Tommaso Venturini, 2017 Public Data Lab.

[2] https://scikit-learn.org/stable/modules/feature_extraction.html

[3] https://www.buzzfeednews.com/article/craigsilverman/fake-news-survey

[4] https://www.nytimes.com/2016/12/06/us/fake-news-partisan-republican-democrat.html

[5] https://www.oxforddictionaries.com/press/news/2016/12/11/WOTY-16

[6] https://medium.com/@sabber/classifying-yelp-review-comments-using-lstm-and-word-embeddings-part-1-eb2275e4066b

[7] https://www.statista.com/statistics/678000/fake-news-media-frequency/

[8] Conroy N, et al., *Automatic Deception Detection: Methods for Finding Fake*. 2015. Language and Information Technology Research Lab.

[9] Del Vicario M, et al., *News consumption during the Italian Referendum: A cross-platform analysis on Facebook and Twitter*. 2017. International Conference on Data Science and Advanced Analytics

[10] Del Vicario M, et al., *Polarization and Fake News: Early Warning of Potential Misinformation Targets*. 2018

[11] Kleinberg B, et al., *Using Named Entities for Computer-Automated Verbal Deception Detection. 2018*. Journal of Forensic Sciences, Vol. 63, No. 3

[12] Perez-Rosas V, Kleingberg B, Lefevre A, Mihalcea R, *Automatic Detection of Fake News*. 2017.Proceedings of the 27th International Conference on Computational Linguistics

[13] Potthast M, Kiesel J, Reinartz K, Bevendorff J, Stein B, *A Stylometric Inquiry into Hyperpartisan and Fake News*. 2017. Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Vol 2: Long Papers

[14] Rubin V, et al., *Deception Detection for News: Three Types of Fakes*. 2015. Language and Information Technology Research Lab

[15] Shu K, Sliva A, Wang S, Tang J, Liu H, *Fake News Detection on Social Media: A Data Mining Perspective*. 2017. ACM SIGKDD Explorations Newsletter, Volume 19 Issue 1

[16] Wang W, *"Liar, Liar Pants on Fire": A New Benchmark Dataset for Fake News Detection*. 2017. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vol 2: Short Papers