Fake and Hyper-partisan News Identification

Vlad Cristian Dumitru
University Politehnica of Bucharest
Bucharest, Romania
d.vladk@gmail.com

Traian Rebedea
University Politehnica of Bucharest
Bucharest, Romania
traian.rebedea@cs.pub.ro

ABSTRACT

Fake news represent a phenomenon that has widened over the last period of time. Their spreading is facilitated by social media platforms, which have a major influence on people's lives. In this context, automated solutions are sought to detect the fake news circulating online. In this paper, we developed models designed to detect fake and hyper-partisan news using various machine learning algorithms. The paper contains a description of the attributes used and their importance in identifying fake news, and also different trained classifiers using more dataset distributions in order to see the differences between their results. In addition to the results analyzed on existing datasets in the field, we tested the ability of models to work in a real environment. Thus, we collected URLs from Twitter, tested the models on these data and made statistics on the number of fake news articles found by each classifier. We also gave some examples detected by models on articles from Twitter which we considered fake news or hyper-partisan news by manual analyzing their content.

Author Keywords
Fake news, Hyper-partisan news, Linguistic features, Natural language processing, Machine learning

ACM Classification Keywords
I.2.7 Natural Language Processing

INTRODUCTION

Misinformation, hiding the truth, or presenting information in a manner that is influential in order to serve certain interests are tactics that have their roots well established throughout history. Examples of this are the famous Trojan horse that proved to be decisive in winning a war, or the 1835 publications on the discovery of life on the moon that made from The Sun one of the world's most popular newspapers [1]. Nowadays, with the development of the Internet and technology, news tend to circulate to a large extent in the online environment. Recent studies show that online sources exceed in some cases the power of the television to transmit the information. At the same time, the traditional newspapers have lost much ground in front of online news [2]. These happen due to the fact that people's tendencies have changed. They find it quicker and cheaper to search for online news than to watch a news storyline on TV or to buy a newspaper. Social media platforms also play an important role in this change. Their users can easily share news, comment, and have free talks about the various news circulating in their online circle of friends. However, these advantages of social networks make them a vulnerable point which can be exploited to disseminate fake news.

Although there are more subcategories related to fake news such as satire, parody or clickbait, a general definition of the term could be the following: fake news represent a way to spread false information in order to mislead the public, damage the reputation of an entity or have a political or financial gain. The idea of misleading and influencing the public is also linked to the notion of hyper-partisan news. The latter have the role of presenting extremist or conspiratorial opinions with intentional misconceptions. Given the fact that this phenomenon of online fake news has been growing sharply lately, especially with the US presidential election campaign in 2016, attempts are being made to find automated solutions to identify such news articles circulating online. Approaches aim in most cases to create automated learning models that process language indicators of texts.

Considering the context described above, the paper we proposed consists of creating such machine learning models like Support Vector Machines (SVM), Random Forests (RF) and Logistic Regression (LR) that detect fake and hyper-partisan news. For training the models, we used several datasets from domain. In order to verify the correctness of the obtained models, we used various metrics showing the ability of the models to correctly identify the input data. Although datasets may have a certain theme and may influence the ability of such a model to properly classify an item [3], obtained classifiers had been also tested in a real work environment. For this, we used the Twitter platform to collect news URLs found in various tweets which were further used as data input for models. We conducted a statistical analysis of the number of news detected by each model based on the value of the decision threshold used to see if a news item is false. In the end, some examples of fake and hyper-partisan news detected by models from Twitter data can be observed.

In this paper, our goals are to obtain classifiers with good metrics, to identify linguistic features that can be used to differentiate a real article from a fake or hyper-partisan one and to check if the existing datasets can be used to obtain models capable of predicting fake news in real-time.

The paper continues with a section that describes the current state of the fake news phenomenon and the systems and the approaches studied so far. The following section describes our proposed solution and an analyze of linguistic features differences between real and fake news and between normal and hyper-partisan news. The last two sections of the paper present results of the models and conclusions of the survey.

RELATED WORK

Automatic detection of fake news supposes technology involvement in identifying articles whose content deliberately misrepresents ideas. Such mechanism has the role of proposing a score that verifies the veracity of a news item [4]. Being a relatively new topic, there is currently no perfect solution that can automatically identify a false story in real-time. Researchers address the issue from two perspectives: a linguistic approach and a network-based approach [5]. The linguistic approach involves analyzing natural language to solve the problem. This proposal aims at identifying text properties as attributes for an automated learning model. Among the identified attributes can be found: the use of punctuation marks, the emotional valences that an article transmits, the analysis of the words used, the similarity of the content, the syntactic analysis of text. A starting point in the linguistic approach is the use of N-grams, which are extracted from the text of the articles and are used as term frequency-inverted document frequency (TF-IDF) in classification models. As for the network approach, it assumes an analysis of information using the structure and behavior of the network. The principle of this method involves the use of graphs built on the relationships between entities. Thus, establishing the veracity of a fact becomes a problem of network analysis.

An analysis of fake news using N-gram models was done by Ahmed et al. [6], where two methods of extracting attributes are studied and the results of several classifiers are compared. First of all, input data was tokenized, the stop words were removed and the stemming process took place. After the processing of the input data, the features were extracted from texts using two approaches, Term Frequency (TF) and Term Frequency – Inverse Document Frequency (TF-IDF). Several machine learning algorithms were used to obtain the various models used for analysis: Stochastic Gradient Descent (SGD), Linear Support Vector Machines (LSVM), Support Vector Machines (SVM), K-Nearest Neighbor (KNN), and Decision Tree (DT). The results of the study show that the accuracy decreases with the increase in N-grams. At the same time, the study shows that the results do not differ greatly considering the length of vectors used for TF and TF-IDF. The best results were obtained for N = 1. Also, the results were better using TF-IDF versus TF. Another aspect to be mentioned is that the best result was obtained using LSVM, with an accuracy of 92%.

Another interesting approach in the process of classifying a news as being fake or not is to identify the position of an article's text in relation to its title. In Hanselowski et al. [7], this metric is seen as an essential step in the process of identifying fake news. As a result, the Fake News Challenge (FNC-1) had as a first step the identification of the relationship between the content of an article and its title. The paper mentioned above describes the systems with best results from FNC-1. The winning team's approach was to use a deep convolutional neural network (CNN) and a gradient-boosted decision trees model. The neural network uses a pre-trained word2vec model, multiple convolution layers and a final softmax layer for classification. The decision trees model uses TF-IDF, word counting, and attributes for feelings that emerge from the text combined with word2vec elements.

In Yang et al. [5] is presented a unified model of convolutional neural networks that uses both extracts from the text of an article and the images appearing in that article called Text and Image Information Convolutional Neural Network (TI-CNN). It is considered that besides the linguistic aspects that differ in a fake article from a real one, the images used are different from the real ones as well. They can encounter images that do not correspond to the text in question or processed images meant to support the story behind the article. Thus, there are two different perspectives that intertwine to form this model: text-based analysis and image analysis. The TI-CNN model has an accuracy of 92%.

Besides producing models with good results on existing datasets in the field of fake news, the main challenge is to create models that are able to classify current news circulating online and thus, to work in a real-time environment. Thereby, Ajao et al. [8] presents three models that identify fake news in tweets posted on Twitter. Such messages posted on social networks can have a major impact in certain situations, with influences on many areas as politics, education or finance. The first proposed model is a Long-Short Term Memory (LSTM) recurrent neural network (RNN). The second variant is LSTM with regularization dropout layers, used to avoid overfitting generated by the train set. The third proposed version is LSTM with CNN. The best results were obtained by the first model, with an accuracy of 82%.

Given the fact that news tends to spread rapidly through social networks, a solution should also be considered for the moment when a fake news is revealed, indicating which is the next step in order to correct the vision already formed on the basis of that news and how to stop distributing it. To combat the above-mentioned problems, there are special users on Twitter who are called guardians [9]. Such a user has the role of correcting the false information that appears on Twitter, giving arguments to prove what is false and what is true. In order to encourage the work of the guardians, the article also proposed a method whereby guardians can get suggestions to check the news that is consistent with their areas of interest. This recommendation has the role of helping guardians to access interesting news for them and to identify fake news as soon as possible. In designing the model, there were taken into account both the articles that a guardian had previously analyzed and the relationship between guardians.

PROPOSED SOLUTION

In this paper, we implemented several classification models, both for detecting fake news and for identifying hyper-partisan news circulating in an online environment. In order to train these models, we used several datasets in the field. This section has the role to describe the used datasets, the selected features for training the models, as well as an analysis of these attributes. At the same time, this section describes how models were tested on news collected from a real environment and how data was obtained. So, we considered that the diversity of data helps to make models work better in a practical situation and facilitates the generalization of their ability to detect false news.

Fake News Identification

As for datasets used for training models in order to identify fake news, we tried to collect training articles from as many sources as possible. The more diverse the datasets are, the less likely it is for trained models to detect only articles specific to a particular writing style or category of news. So, we considered that the diversity of data helps to make models work better in a practical situation and facilitates to generalize of their ability to detect false news. An entry in the dataset is characterized by the title of the article, its author, the publication where the article originates, its textual contents and a label that describes whether the article is fake or true. For articles where the title, the author or the publication is missing, we have noted entries with generic names such as notitle, noauthor, and nopublication.

The dataset is not balanced, meaning that real and fake articles are not in equal distribution. In general, in a real environment true news are more frequent than fake ones, so the same distribution was preserved to our dataset. The idea is that a model should learn as well as possible how a real news article looks like in order to detect when something is not real. However, we have also made models with fewer, but balanced data in order to see differences between their results in how they detect items collected from Twitter. For the moment, the next statistics about features are made for the real distribution dataset.

Datasets consist in articles written in English and can be found on Kaggle. In this paper, the accent is on linguistic features.

First of all, we used TD-IDF to have a statistical representation of the importance of words from the corpus of articles. A TF-IDF value for a word increases proportionally to its number of occurrences in the document and decreases proportionally to the number of documents in the corpus that contain the word. Thus, a TF-IDF value measures how relevant a word is in a particular document. The first attributes calculated for models were TF-IDF values of the article text content. Moreover, there have been computed TF-IDF values for news headlines as well. N-gram used were of one and two words and we used stop words in order not to taking usual terms into account. To get the TF-IDF attributes from the article's content we kept the most important 2500 words sorted by the TF value, while for the TF-IDF attributes in the news headlines we kept the first 500 words.

In addition to these TF-IDF values calculated from title and content of the articles, we have also computed other attributes related to the text of the news, seeking to find various features that help distinguish between real and fake news. Firstly, there were added features related to the notion of word such as the number of words in an article, the average length of a word in a news item and the number of words in an article written in capital letters. Words written in capital letters are important because these kind of words are frequently used in a story in order to capture the attention of the users and to make them read that story. In order to compute the occurrences of these words, there have been considered only words with more than three characters to avoid counting the abbreviations. Figure 1 presents graphically represented differences between real and fake news for mean values of word features.

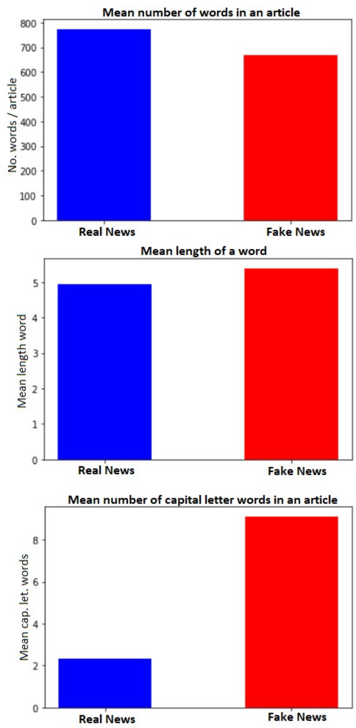

Figure 1. Mean differences between real and fake news considering word related features.

Secondly, there were added features related to sentence like number of sentences in each article, mean length of a sentence in an article and mean number of words in a sentence. There has been used a tokenizer from Natural Language Toolkit (NLTK) in order to split texts in sentences and to extract these features. We could observe that a real news contains more sentences than a fake one, instead the average length of a sentence in an authentic article is smaller than the average length of a sentence in a fake article. However, the mean number of words in a sentence of a real news is approximately equal to that in a fake article. This can be explained by the fact that the average length of a word is larger in a fake news.

Furthermore, we computed the number of exclamation marks and question marks. Both values are low, but in both cases these values are a little bit higher for fake news.

Another category of features that we chose was composed of the number of adjectives and superlatives found in the title and text of an article. In order to get these features we used another facility of the NLTK platform, called part-of-speech tagging (POS-tagging). By computing the mean value of adjectives and superlatives in an article, we found that in average both numbers are higher for real articles. An explanation for these results could be that real text relies heavily on the description of presented events because they describe authentic facts, so more adjectives will be used. Instead, a fake text will not insist on the description in order not to reveal itself.

Moreover, we computed features related to sentiment analysis using Vader Lexicon from NLTK. Thus, for each article, four scores were obtained: positive, negative, neutral and compound. The last feature that was computed is the cosine similarity between text and title. This notion is a way to check how similar two words, sentences or documents are. Cosine similarity involves calculating the cosine given by the angle formed by the scalar product between two vectors that identify a sentence or text.

Finally, an attribute vector for an item will contain 3015 features. 2500 are TF-IDF values for text of the article, 500 are TF-IDF values for title, and the other 15 attributes are linguistic.

To detect fake news, we trained three types of machine learning models. The first one is based on the Support Vector Machines (SVM) algorithm. This algorithm is used in classification issues and uses a hyper-plan to delimit class values that separate them. The method follows the discovery of the hyper-plan that best separates the points in the two classes. If the points in the two classes cannot be linearly separated, a kernel is used to transform the input data into a suitable shape for finding a hyper-plan, so that the transformed data can be separable. Thus, in practice, SVMs are implemented using the kernel concept. For the SVM model that we proposed, we used the linear kernel because this kernel is recommended for text classification due to the fact that texts are, in general, linearly separable and additionally, the linear kernel performs well when many attributes are used for classification. The second model that was implemented is based on Random Forests (RF) algorithm. This method involves building multiple decision trees that will independently calculate the class of a particular input. Finally, the exit will be given by the majority class. Thus, the phenomenon of overfitting can be controlled. The last model implemented is based on Logistic Regression (RL) algorithm. All three models were obtained using elements from scikit-learn library.

In order to test the functionality and correctness of the above models, the datasets were split in 75% train set and 25% test set. Another important aspect of model trainings is data scaling. This scaling is needed to bring the values of the attributes in approximately the same range so that all attributes can equally influence the calculation of hypothesis for a model.

Hyper-partisan News Identification

Regarding the dataset for hyper-partisan news, it contained only URLs for news. This corpus is also compound from article written in English. Therefore, URLs needed to be processed in order to get the content of the articles. For this, we used newspaper3k library [10]. The analysis of attributes mean values is also made on a dataset which imitates a real world environment distribution where real news are more common than hype-partisan news. The attributes used to build models that detect hyper-partisan news are the same as those used to identify fake news.

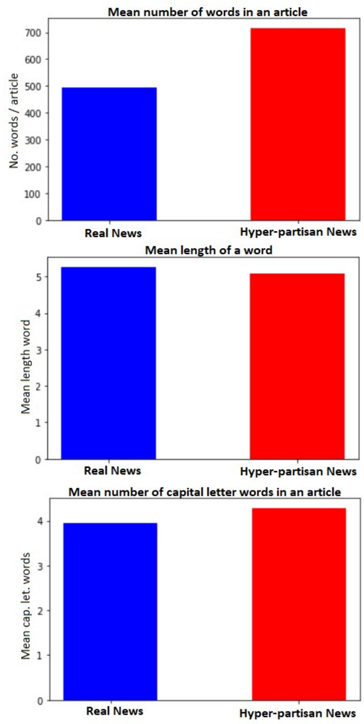

Figure 2. Mean differences between normal and hyper-partisan news considering word related features.

Figure 2 presents differences between normal and hyper-partisan news related to the notion of the word. It can be observed that the average number of words in a hyper-partisan news is higher than in a normal news.

The next comparison is given by the sentence-related features. We noticed that a hyper-partisan article contains on average more sentences than a normal story. This statistic is somewhat intuitive given that the number of words in a hyper-partisan article is on average higher than that of an authentic article, and the average length of a sentence and the average number of words in a sentence have values close to each other in both cases.

Another important aspect that worth mentioning is that the mean number of adjectives is greater in a hyper-partisan news than in a normal one. This can be explained by the fact that a hyper-partisan news can use more adjectives to emphasize the ideas it wants to convey and the reasoning it exposes.

For models built to detect hyper-partisan news, we used the same algorithms used to classify fake news. If for fake news we left all the default parameters for the SVM model, this time we built two models based on SVM by changing parameter C. For the first model we left the value C = 1, which is the default value, and for the second model built with SVM we chose the value C = 0.025. When using SVM, parameter C has the role of determining how the separation hyper-plan between the two classes is chosen. Also, the dataset was divided into 75% training set and 25% test set, and features were normalized.

Fake News vs. Hyper-partisan News Characteristics

First of all, in the case of fake articles, we could see that the average length of a word is higher than the one calculated for real news. Also, the number of words written only in capital letters is higher in fake news. However, in the case of hyper-partisan news, these values are similar to those of authentic articles. Another interesting observation is that a hyper-partisan article tends to have more sentences than a normal article, instead, a fake news tends to have fewer. As for the average of question and exclamation marks, both have greater values in the case of fake and hyper-partisan articles. At the same time, in the case of fake news, they contain on average fewer adjectives and superlatives than real news, but hyper-partisan articles contain more of these parts of speech versus authentic news.

These results can be influenced by the datasets used to produce these statistics, but it can still be said that the features of fake news are somewhat different from the characteristics of hyper-partisan news.

Twitter Data Collection and Processing

We tested the models in a practical environment in order to see if existing datasets in the field are useful for implementing classifiers that can be used in practice. For this purpose, we collected posts from Twitter, extracted the links found in those posts, and with the help of a few applied filters we obtained the articles from those links. In this process, we used Tweepy library, which provides working methods to access and work with the Twitter API. We used a streamer object to collect real-time data using some keywords. Among keywords used to collect tweets there were usual English words as *in*, *the*, *on*, *for*, *you*, *at*, etc. as well as some words that could indicate a fake article from politics field as *election*, *politics*, *news*, *fake*. The latter words were chosen because in general fake or hyper-partisan news is related to the political domain. The next step was to extract the URLs that were contained in the collected posts and to keep the unique URLs. These URLs extracted from the posts are like *https://t.co/tUolsLpTEE*, so they needed an extension to reach the final link that leads to that page as the newspaper3k library cannot directly open Twitter links. Only those links were kept, which, according to the newspaper3k, have great chances to represent an article. Then, observing that the remaining data still has lots of articles that are not written in English or that are not news, we added a filter to keep only items that have English content, that have over 450 characters and that do not contain certain words that may represent a page which is not a news item. Figure 3 presents the steps that were followed in order to obtain articles from data collected among Twitter posts.

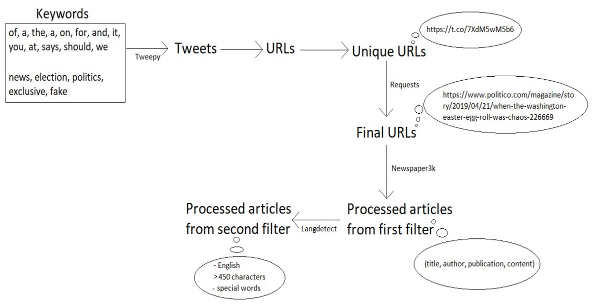

Figure 3. Process used in order to collect and filter data from Twitter.

RESULTS

This section is intended to present the results of the models on the datasets we trained using various metrics, as well as a statistic of identifications on data collected from Twitter. Finally, there are some online news that we considered fake or hyper-partisan and which we identified with the help of the implemented models. To evaluate and compare the obtained results, we used four metrics: accuracy, precision, recall, and F1 score. As mentioned in the previous section, datasets were divided into 75% training data and 25% testing data to verify the functionality of the models.

Fake News

The initial configuration used to train models that identifies fake news contained 191383 articles, of which 159884 real news and 31499 fake news. The results obtained can be seen in Table 1. According to the values in the table, the best metrics were reached by Random Forests, while for SVM and Logistic Regression metric values are quite close.

	Accuracy	Precision	Recall	F1 score
SVM	0.951	0.887	0.807	0.845
RF	0.982	0.990	0.899	**0.942**
LG	0.949	0.896	0.782	0.835

Table 1. Results for fake news models trained with a real distribution of news in dataset.

In addition to this news distribution that is designed to simulate real-world proportions, a configuration was also selected where the two types of classes used can be found in equal proportions. So, the second configuration used 10465 real news and 9514 fake news. The results obtained for the models are shown in Table 2. Figure 4 shows the most important attributes taken into account for detecting a real or a fake news for the SVM trained with this configuration of data. In red are the top 20 attributes used to detect a real news, and in blue are the 20 main attributes used to identify a fake story.

	Accuracy	Precision	Recall	F1 score
SVM	0.922	0.926	0.910	0.918
RF	0.924	0.938	0.902	**0.920**
LG	0.913	0.916	0.903	0.909

Table 2. Results for fake news models trained with an equal distribution of news in dataset.

Hyper-partisan News

The first configuration used to train models to detect hyper-partisan news was composed of 44267 articles, of which 35414 normal and 8853 hyper-partisans. Table 3 lists the results obtained by the four implemented models. SVM1 is the model that has parameter C = 1 and SVM2 is the model that has parameter C = 0.025. For the latter model, the recall value is very low, suggesting that the model fails to identify the hyper-partisan news.

	Accuracy	Precision	Recall	F1 score
SVM1	0.928	0.871	0.758	**0.810**
SVM2	0.842	0.976	0.221	0.361
RF	0.919	0.969	0.619	0.755
LG	0.919	9.878	0.697	0.777

Table 3. Results for hyper-partisan news models trained with a real distribution of news in dataset.

The second configuration was with equally distributed data. In this case, the dataset was composed of 75,000 articles, of which 35414 authentic and 39,586 hyper-partisans. Table 4 shows the results obtained with this configuration. The weakest results for the metrics were obtained by the SVM model with the parameter C = 0.025. Results for the SVM model with C = 1 and for the LR model are close to each other, while RF obtained the best values for the analyzed metrics.

	Accuracy	Precision	Recall	F1 score
SVM1	0.927	0.911	0.954	0.932
SVM2	0.885	0.849	0.948	0.896
RF	0.965	0.954	0.979	**0.967**
LG	0.923	0.908	0.950	0.928

Table 4. Results for hyper-partisan news models trained with an equal distribution of news in dataset.

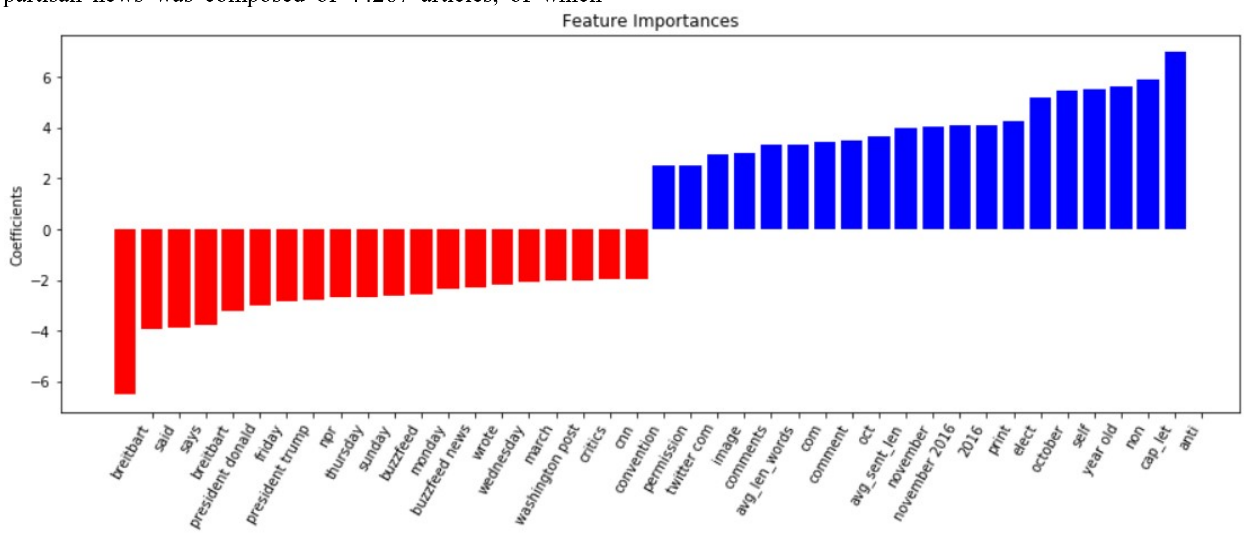

Figure 4. Most important features used by SVM model trained with an equal distributed dataset.

Testing models in a real-time environment

Obtained models were tested on data collected and processed from Twitter in order to analyze how many of the articles are detected as fake or hyper-partisan news. We also chose more values for the threshold used to identify a suspicious news. By default, this threshold value is 0.5. Each model returns a score, which is the probability that an item is fake or hyper-partisan. If this probability is greater than the threshold value then the article will be labeled with 1. The threshold values chosen for the model results were: 0.5, 0.75, 0.85 and 0.9. In total, there were 48769 supposed articles on Twitter. All detections were made on the same 48769 articles, both for fake and hyper-partisan news.

	Accuracy	Precision	Recall	F1 score
SVM	23.90%	12.75%	8.74%	6.55%
RF	2.38%	0.08%	0.01%	0.00%
LG	18.58%	8.13%	4.88%	3.35%

Table 5. Detections percentages for fake news made by models trained with a real distribution of news in dataset.

Table 5 lists the detection percentages for each fake news model and for each selected threshold value for classifiers trained with a real distribution of news in dataset. As one can see, the highest decrease in percentages with the increase in the threshold value is recorded by the SVM model. Table 6 has the role to present percentages of detections made by models trained with balanced data. Compared to the percentages proposed by the models trained with a set of data where true news prevails, in the case of models trained with a balanced dataset the percentages have increased. The highest drop in percentage with the rise in threshold value is given by the Random Forest model.

	Accuracy	Precision	Recall	F1 score
SVM	45.44%	29.79%	22.15%	17.69%
RF	43.87%	5.43%	1.07%	0.29%
LG	41.14%	16.22%	7.85%	4.63%

Table 6. Detections percentages for fake news made by models trained with an equal distribution of news in dataset.

Further, there are detections made by hyper-partisan models with both configurations on datasets. Table 7 contains the detections made by the four models in percent for a real-time distribution of news in dataset. The difference between the values of parameter C for the SVM model can be observed by the way the two models detect new data. It can be noted that for each threshold value the SVM model with C = 0.025 identifies fewer items as hyper-partisan. Table 8 contains percentage of models trained with balanced data. As one can see, the highest fall in percentage is given by RF model, followed by LR model.

	Accuracy	Precision	Recall	F1 score
SVM1	34.81%	18.86%	12.53%	9.98%
SVM2	30.35%	14.22%	8.76%	6.14%
RF	5.92%	0.12%	0.04%	0.01%
LG	24.78%	7.41%	3.32%	1.87%

Table 7. Detections percentages for hyper-partisan news made by models trained with an equal distribution of news in dataset.

	Accuracy	Precision	Recall	F1 score
SVM1	67.34%	46.73%	35.32%	28.11%
SVM2	64.47%	38.43%	24.72%	16.95%
RF	68.12%	10.28%	1.49%	0.32%
LG	69.17%	36.67%	21.53%	13.88%

Table 8. Detections percentages for hyper-partisan news made by models trained with an equal distribution of news in dataset.

In order to check the functionality of models in a real environment and to establish the accuracy of classifiers in practice a manually labeling of data collected from Twitter would be needed. For the time being, the real-world statistics obtained for the created models are only theoretical. These detections have to be checked manually to determine if they are valid or not.

Fake News and Hyper-partisan News detected examples

We have manually analyzed some of the news that have been flagged as fake or hyper-partisan, and we chose some of them that we really considered correctly identified, highlighting them in the next tables. These news were classified with SVM models with the C = 1 parameter and have a higher probability than 0.90 to be fake or hyper-partisan according to the scores returned by these models. Further, there are listed five of the possible fake news that have been identified in Twitter collected articles:

- Crooked Hillary Lashes Out at Trump: “The President is Not Above the Law” (thegatewaypundit.com, June 2019)
- Muslims have killed nearly 700 Million NON -Muslims, but no one is saying anything. (trump-train.com, March 2019)
- Trump Won the Popular Vote (magamedia.org, May 2019)
- Remember: UK Intel and Obama’s Deep State Tried to Stop Trump in 2016 – UK Did More to Interfere on US Election than Russia (thegatewaypundit.com, June 2019)
- LEAKED: OBAMA TEAM KEPT LIST OF MUSLIMS FOR TOP JOBS, EXCLUDED NON-MUSLIMS (dailycaller.com, October 2016)

The next ones are five possible hyper-partisan news that have been detected in data collected from Twitter:

- The Mueller Report: Trump Too Inept to Obstruct Justice (counterpunch.org, April 2019)
- New York Times Apologizes To Trump in Stunning Reversal, Demands Other Media Do Same (kagdaily.com, May 2019)
- Americans Are Brutally Paying Back CNN & MSNBC for 2 Years of Lies About Trump (westernjournal.com, May 2019)
- Hannity: Fox News 'Talk Show Hosts' Are Better Journalists Than '99%' of the Media (thedailybeast.com, May 2019)
- Elizabeth Warren Calls on America to Boycott Fox News: They Spread Hate And Racism (hillreporter.com, May 2019)

Analyzing the detections made by the models we have implemented on the data collected from Twitter, we noticed that there are quite a lot of real news that were considered fake. This happen because most of the news that make up the current datasets is based on articles related to the US presidential election in 2016, and some features that could had indicated fake news in that time may be perfectly valid for real news nowadays. For example, two of the high coefficient attributes indicating a fake news in the models obtained were *anti* and *non*. Instead, these words are used quite often in the actual news of anti-Semitism.

CONCLUSIONS

The purpose of this paper is to present the concepts of fake and hyper-partisan news and to propose solutions in order to identify such articles. We first made a description of the problem and of the context in order to draw a clearer picture of this phenomenon. Then, we described our proposed solution. We implemented several automated learning models using existing datasets in the field, and we conducted an analysis of the differences in attributes between a real and a fake news and between a normal and a hyper-partisan news story to form an idea about their linguistic differences. We presented the results obtained and compared them using various metrics. Generally, the models obtained have an accuracy of more than 90% on datasets. In addition, we wanted to see how the models behave in a real working environment. So we collected Twitter data from which we got news URLs and used them with trained models. Finally, we added a few examples of what classifiers detected on Twitter data that we considered to be fake or hyper-partisan.

Related to the future approaches of the issue of identifying online fake news from the current study, a first step would be to add more datasets. Analyzing data detected by models on Twitter articles, we noticed that the news circulating in the online environment is from diverse domains. So, we consider it useful for models to be trained with news from multiple domains, not just with politically related news, as is currently the case because datasets are largely composed of such news. Other proposals for future approaches are adding new categories of attributes, as well as implementing other learning models.

REFERENCES

1. Greener, D. (2018, September 27). 4 Shocking Examples of Fake News Throughout History. Retrieved from https://medium.com/bigger-picture/4-shocking-examples-of-fake-news-through-history-1dce02385ba6.
2. Oxford University. (2017, May 30). Where do people get their news? Retrieved from https://medium.com/oxford-university/where-do-people-get-their-news-8e850a0dea03.
3. Office, R. (2019). Peering under the hood of fake-news detectors. Retrieved from http://news.mit.edu/2019/opening-machine-learning-black-box-fake-news-0206.
4. Conroy, N. J., Rubin, V. L., & Chen, Y. (2015). Automatic deception detection: Methods for finding fake news. Proceedings of the Association for Information Science and Technology, 52(1), 1-4.
5. Yang, Y., Zheng, L., Zhang, J., Cui, Q., Li, Z., & Yu, P. S. (2018). TI-CNN: Convolutional neural networks for fake news detection. arXiv preprint arXiv:1806.00749.
6. Ahmed, H., Traore, I., & Saad, S. (2017, October). Detection of online fake news using n-gram analysis and machine learning techniques. In International Conference on Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments (pp. 127-138). Springer, Cham.
7. Hanselowski, A., PVS, A., Schiller, B., Caspelherr, F., Chaudhuri, D., Meyer, C. M., & Gurevych, I. (2018). A retrospective analysis of the fake news challenge stance detection task. arXiv preprint arXiv:1806.05180.
8. Ajao, O., Bhowmik, D., & Zargari, S. (2018, July). Fake news identification on twitter with hybrid cnn and rnn models. In Proceedings of the 9th International Conference on Social Media and Society (pp. 226-230). ACM.
9. Vo, N., & Lee, K. (2018, June). The rise of guardians: Fact-checking url recommendation to combat fake news. In The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval (pp. 275-284). ACM.
10. Newspaper3k. https://pypi.org/project/newspaper3k/