NLP Project - Fake New Detection

Baptiste Engel, Corentin Royer

May 2022

Abstract

In this project, we used Machine Learning methods to classify media articles and social media posts into fake or true news. We applied several algorithms from classic machine learning methods such as SVM to newer deep learning approaches with LSTM. We put an emphasis on data preprocessing and on the understanding of our dataset. Our algorithms were over 95% accurate.

1 Introduction

Fact-checking is an important part of our modern society. However, an vast amount of news are produced each day, and one cannot rely on human verification to classify a news as fake or not. Therefore, fake news can spread really easily and in an uncontrolled manner before being checked. This can have a quantitative impact on public opinions [4].
Natural Language Processing (NLP) alongside with Machine Learning (ML) technologies have shown to be efficient for fake news detection [5]. We chose to start by investigating the performance of standard ML techniques to detect fake news in the ISOT Fake News Dataset [1][2]. Then, we wanted to see if a state-of-the-art recurrent neural network increase the performance of our system.

2 Methodology

2.1 Dataset

For this project, we were interested in classifying fake news by using only textual information. We did not use any metadata (author, date of publication, source or title of the article/post), only the text content. Our study focuses on the ISOT Fake News Dataset [1] [2], containing 44955 news articles of different topics separated in two categories (fake or true).

The true news come from Reuters, a respected news outlet and the fake news come from Facebook pages known for spreading fake news. The news have a wide range of length, the shortest one are a fews words long whereas the longest

article is over 8000 words long. This variation is about the same for true and for fake news.

The news are separated in 8 different subjects, displayed Table 5.

2.2 Standard Machine Learning methods

We first implemented the classification strategy from [1] in which Ahmed et al. applied standard machine learning to features extracted from the texts.

2.2.1 Data preprocessing

We chose to apply the same preprocessing strategy as [1] and add some of the methods cited in [5]. We removed stop-words (i.e small and common words that carry little meaning such as "as", "in" etc.), we removed punctuation and digits, we lower-cased the words, finally we tokenized and applied stemming. After these steps, we had removed most of the less meaningful parts of the texts and all of the remaining words were usable by the feature extraction methods we later used.

We also noticed that the true news all contained the source of the article (i.e they all started by "CITY (Reuters)"), this made it very easy to classify as one just had to check for these specific words to know if the text was a true news. We thus decided to remove these words. Our approach differs from [1] where they decided to reduce the size of the dataset instead. We applied the algorithms both with and without those last modifications.

2.2.2 Feature construction

A classic and simple feature to start with document classification is the Term Frequency (TF), that can be improved by calculating the Inverse Document Frequency (IDF), resulting in the TF-IDF feature. To compute the TF, we count how many times a word appear in the text, and we normalize with the text length. The TF-IDF is a way to highlight words who appears less in the corpus, because they contain more information than a generic word (*the* is a generic term and will have a huge TF, but it will be counter balanced by the IDF.)
For a term t in a document d, those features as computed as follow:

$$TF(t) = \frac{n_{t,d}}{\sum_k n_{k,d}} \tag{1}$$

$$IDF(t) = \log \frac{|D|}{|\{d : t \in d\}|} \tag{2}$$

where $n_{k,d}$ is the number of times term k appears in the document d, and D is the corpus of documents. The TF-IDF feature is then computed by multiplying TF(t) with IDF(t).

As we are dealing with a huge number of documents (44955 articles), we have, after preprocessing, 88337 different words. We need to reduce the dimensionality of those vectors. As in [1], we only took a fixed number of word among the ones who appears more frequently in the corpus, and build the TF-IDF feature with those words.

2.2.3 Classification algorithm

We applied several classification algorithm on the features previously extracted. We used Support Vector Classification with a rbf kernel, K-Nearest Neighbors and Random Forest of 100 trees from the SKLearn library. Training was made using 80% of the dataset, and the evaluation with the remaining 20%.

2.3 Deep Learning method

In a second stage, we implemented a deep learning method to classify the texts. To do that, we needed to adapt the preprocessing and use a suitable neural network.

2.3.1 Data preprocessing

As input of the neural network, we needed a representation of the words. The simplest one is the indices in a word list or a one-hot encodding. These option are not suitable due to the unwanted proximity between adjacent words for the former and the high dimensionality of the latter. We thus needed to embed each word of the sentences in a lower dimensionality. Famous word-level embedding are GloVe [6] and word2vec [3]. We chose to embed with GloVe [6] on vector of length 50, because we already used this embedding and we wanted to know if it can be used for this classification task. We used a pretrained GloVe dictionary because the training requires large amount of data and long training time, in addition excellent models already exist.

Before feeding the sentences to the neural network, we tokenized them and lower-cased the words. We also have the option to remove the stop-words and the beginning of the true news (the reference to Reuters). We cannot use stemming in this case because the resulting words wouldn't match the GloVe dictionary. The neural network also needs sentences of the same length so that we can batch them, we decided to only take the first 100 words (i.e to truncate the longer stories and to pad the shorter ones). It was supposed to be a first simple approach but proved effective enough so we kept it.

2.3.2 Implementation of the neural network.

The architecture we used is as shown in 1, we feed the words from the sentences we preprocessed one by one, first into the word embedder and then into the LSTM cell with one layer of 128 neurons. When we reach the end of the sentence, we use the output of the LSTM as input of a linear layer which reduces the

dimension from the size of the LSTM's hidden layer to 1. Finally we apply the sigmoid function and we have our prediction.

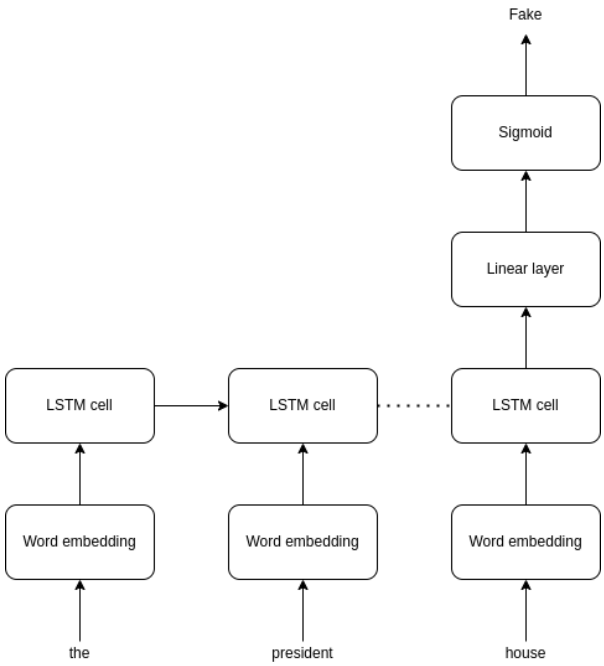

Figure 1: Architecture of the neural network.

We train the model using batches of 250 sentences on 80% of the dataset (leaving 20% for the testing). We trained for 3 epochs which lasted about 20min on a Nvidia GTX1060 GPU.

3 Results

We used the accuracy as our main metric to evaluate the performance of our algorithms. We also used confusion matrices to have a bit more depth in the results of the classifications. We split the dataset in 2 with 20% dedicated to testing and the rest used for training.

3.1 Comparison of performances of standard Machine Learning methods

Result for classification with the standard ML methods over all corpus are shown Table 1. We can see that even by taking only the 10 most frequent words in the corpus to produce the feature, we obtain a high accuracy for the classification. The least performing classifier is the k-nearest neighbour. Also, we can see that

the TF-IDF feature is mostly worst than the TF alone. This might be because most of the meaningful words are removed, therefore IDF impact negatively the classification.

	TF		TF-IDF	
Feature Size	10	100	10	100
SVM	99.48	99.48	99.20	99.20
5-NN	82.19	82.19	71.29	71.29
Random Forest	99.17	99.15	99.33	99.29

Table 1: Results of standard classification algorithm on the complete corpus, for different feature size.

When we remove the mention to the city and to Reuters in the true dataset, the classification becomes a bit more difficult which translate in lower accuracy. Still, the absolute result is good with always more than 93% accuracy.

	TF		TF-IDF	
Feature Size	10	100	10	100
SVM	93.93	93.93	98.44	98.44
Random Forest	97.46	97.57	97.55	97.33

Table 2: Results of standard classification algorithm on the complete corpus, for different feature size. The mention to the city and to reuters have been removed.

Table 6 shows that the distribution of articles through year is unbalanced. Therefore, we chose to subsample our dataset to only use articles from a specific year, and to use only the TF feature of size 10. The result of this experience are shown Table 3.

	2016	2017
SVM	99.24	99.38
5-NN	85.79	75.78
RF	98.72	99.13

Table 3: Testing accuracy when training with a subsample of the dataset.

We can see with this experience that the SVM classifier with a rbf kernel seems to be the best of all 3 tested classifiers. However, to test generalization performance, we trained a SVM classifier on articles from a specific year and test the performance on another year or on the whole corpus. The results of this experience are shown Table 4.

We can see on those results that the SVM classifier capacity to generalize is great, and still maintain a high accuracy even on data from another year, and

	Training year	
	2016	2017
Testing year		
2015	96.58	93.96
2016	-	99.03
2017	95.52	-
All corpus	97.22	98.99

Table 4: Accuracy of a SVM classifier trained on a specific year, and evaluated on another.

even if the dataset is unbalanced. For instance, the network train on the 2017 dataset (64% true data) perform well on the 2016 dataset (71% of fake data), with a testing accuracy of 93,96%.

3.2 Performance of a Recurrent Neural Network (RNN)

The first experience we did regarding the RNN approach was without removing the stop words, the punctuation and the reference to Reuters in the true news. In this setup, we got 99.9% accuracy with only 10 true sentences classified as fake. This is slightly better than the performance of standard ML techniques.

We then removed the mention to Reuters as we judged it gave an artificial help for the classification. With the same hyperparameters, we got an accuracy of 98.3%. The lower performance was expected but the accuracy is still very high. The neural network becomes more effective with this setup as compared with the standard ML approaches (whose accuracy ranged from 93% to 98% on this task).

4 Conclusion

The performance of both methods (Standard Machine Learning and RNN) are high on the ISOT Dataset. It would be interesting to compare the performance of the networks we construct on another datasets, such as the LIAR [7], to assess the generalization capability of such methods. Indeed, the data from the ISOT dataset seems to be to easy to classify, probably because of the huge quality gap between the news providers. Other methods, such as Convolutional Neural Networks, are also used for fake news classification, and can be interesting to evaluate.

References

[1] Hadeer Ahmed, Issa Traore, and Sherif Saad. Detection of online fake news using n-gram analysis and machine learning techniques. pages 127–138, 10 2017.

[2] Saad S. Ahmed H, Traore I. Detecting opinion spams and fake news using text classification" journal of security and privacy, volume 1, issue 1, wiley, january/february 2018.

[3] Tomas Mikolov, Kai Chen, G.s Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *Proceedings of Workshop at ICLR*, 2013, 01 2013.

[4] Jayawickrama U. Arakpogun E.O. et al. Olan, F. Fake news on social media: the impact on society. *Inf Syst Front*, jan 2022.

[5] Ray Oshikawa, Jing Qian, and William Yang Wang. A survey on natural language processing for fake news detection. *CoRR*, abs/1811.00770, 2018.

[6] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[7] William Yang Wang. "liar, liar pants on fire": A new benchmark dataset for fake news detection. *CoRR*, abs/1705.00648, 2017.

A ISOT Fake News Dataset description

Subject	True	Fake
politicsNews	11272	0
worldnews	10145	0
News	0	9050
politics	0	6841
left-news	0	4459
Government News	0	1570
US_News	0	783
Middle-east	0	778

Table 5: Subject repartition in the dataset. We can see that a topic are unique for categories. It is therefore not interesting to use topics as a feature.

Year	2015	2016	2017
True	0	4716	16701
Fake	2485	11754	9203

Table 6: Year distribution for each category. We can see that this distribution is quite unbalanced.

	True article	Fake Article
title	U.S. embassy to Russia to resume some visa services after diplomatic row	HOMELAND SECURITY: ISIS Has Already Tried To Exploit Refugee Program To Enter U.S. [Video]
text	"MOSCOW (Reuters) - The U.S. Embassy to Russia said on Monday it would restart some visa services in U.S. consulates which it had previously canceled due to diplomatic expulsions that had left it short-staffed. The United States began to scale back its visa services in Russia in August, drawing an angry reaction from Moscow three weeks after President Vladimir Putin ordered Washington to more than halve its embassy and consular staff. The U.S. step meant Russian citizens wanting to visit the United States for business, tourism or educational reasons were & no longer able to apply via U.S. consulates outside Moscow and had to travel to the Russian capital instead. The embassy said in a statement on Monday some visa services would resume on Dec. 11. "On December 11, the U.S. consulates in St. Petersburg, Yekaterinburg, and Vladivostok will begin to offer limited interviews for non-immigrant visas," it said. "	Our Homeland Security Director gave us all the lowdown on the refugee program and it s what we all thought: ISIS has been trying to use the refugee program to come to America. Why in the heck are we still trying to bring refugees here? Can t we help them in the Middle East? The only thing I can guess is that this is Obama s agenda and he s not letting up on fulfilling it. The State Department and UN are pushing refugees through the system as fast as possible. This is big, big business for the Refugee Resettlement contractors ho get paid per refugee and then get millions from the feds on top of that. The kicker is that YOU pay for the goodies the refugees get once they re here. That s why America is a global magnet for refugees!
subject	politicsNews	politics
date	December 4, 2017	Dec 7, 2015

Table 7: Dataset samples of ISOT Fake News Dataset.