UNIVERSITY OF TRIESTE

Data Science and Scientific Computing

Natural Language Processing project

The problem of fake news classification: A comparison between prediction models and the analysis of online found datasets

Federico Boni

Academic Year 2021/2022

Contents

1 Problem statement

The problem considered in this work is the classification of a given article as a fake news or as a trustworthy article. Fake news are a well known problem and they have found widespread diffusion in recent years thanks to social networks. A recent example is given by the randomized trial carried out by [1], that found how Covid-19 fake news have reduced vaccination intention in the United Kingdom and United States.

The starting point of this project is the training of two Lstm neural networks using one of the most used online dataset; these classifiers differ in the embedding: in the first one the embedding layer is part of the neural network and is trained together with it, in the second one embedding is done using the Word2vec model.

After the training phase, these models are compared with an already fine tuned Bert model [2] (trained on the same dataset) using two different datasets than the first one.

The final goal of the project is to compare performances of the 3 models described above using different datasets then the one used for training, therefore trying also to understand if the latter is sufficiently representative of the reality (i.e. if it represents well the characteristics that distinguish a real article from a fake news).

2 Adapted Methodology

Data collection - The selection criteria of the training dataset is its perceived popolarity. In fact, given the above problem statement, this work focuses on neither data collection nor data classification methodologies. Instead, after the evaluation of the trained models, an analysis of the possible characteristics of the data (that make them a good or bad representation of the problem) follows.

Design choices - Regarding models design, the choices made are based on other works or on limits regarding computation times.

Analysis methods - The models evaluation has its focus on both the comparison of model performances and on how models behave on different datasets. Given that and the nature of the problem, the metrics used are accuracy, precision, recall and f1. It is chosen also to use a statistical test to understand if the differences on the accuracy values found would actually suggest different models performances or not. In particular the McNemar test is chosen based on the results of the paper in [3], where its use has been recommended when cross validation is not affordable.

3 Datasets

3.1 Datasets sources

The following 3 datasets are used for this project. Later in the report, these datasets will be referred to with the numbers assigned below.

Dataset 0 - It seems to be a widely used fake and real news datasets. It was created by the authors of the conference paper in [4]. According to the authors, they collected the trustworthy articles from Routers.com and they found fake news articles in an online dataset obtained from Polifact (a fact checking organization in the USA). On kaggle.com [5] this dataset has more than 400 projects. It is the dataset used for the learning of the 2 Lstm models and the already fine tuned Bert model.

Dataset 1 - The second dataset was built by the UTK machine learning club [6]. They state that they have build it by collecting various datasets found on the net.
Dataset 2 - The third dataset is an anonymous Kaggle.com dataset called "Fake News Detection" [7].

Datasets 2 and 3 will be used after learning to compare the performances of the 3 models.

3.2 Datasets structure, cardinality and dimension

Each article of all the 3 datasets always has a title, a body and a label that specify its category. In addition each dataset could also have other informations, such as pubblication date or author. For the purpose of this project, only the body and the title of each article are used.

Table 1 displays a summary of datasets cardinalities after removing duplicates and empty rows (the number of duplicates and empty rows are shown in Table 2).

Dataset	Fake	Real	Total
0	17455	21191	38646
1	12088	12714	24802
2	1193	1670	2863

Table 1: *Total number of unique and not empty articles for each dataset.*

	dataset 0	dataset 1	dataset 2	Across datasets
duplicates	6252	472	1145	0
empty rows	0	631	1	-

Table 2: *Total number of duplicates and empty rows for each dataset.*

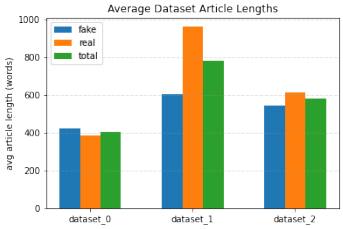

Figure 1: *Average articles length across all datasets.*

Dataset	Fake	Real	Total
0	425	385	405
1	604	963	783
2	547	616	581

Table 3: *Average articles length across all datasets.*

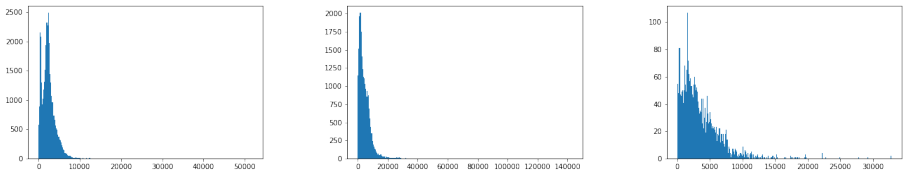

Figure 2: *Distributions of documents length. From left to right: dataset 0,1,2.*

In Figure 1, Figure 2 and Table 3 it can be noted how articles from different dataset have on average different length. An evident observation is that, while for dataset 0 and dataset 2 the average length difference is low between real articles and fake news, for dataset 1 this difference is more marked, with real articles being consistently longer.

4 Preprocessing

In addition to the removal of duplicates and blank articles mentioned in the previous section, several preprocessing operations have been defined before using the datasets. All of the following operations are performed on all datasets before train and/or prediction phases.
The list of performed preprocessing operations is:

- Concatenation of article title and article body
- Removal of stop words (using the standard list of the Gensim package [8]).
- Lemmatization (using lemmatizer from the Nltk package [9])
- Removal of url
- Replacement of numeric characters with "0"
- Removal of non-alphanumeric characters.
- Document padding or trimming.

5 Lstm Models

5.1 Input data

Out of 38,646 total documents of dataset 0, 80% of them (30,913) are choosen randomly as training data, 20% (7,728) as test data during the learning phase. It is chosen to provide inputs of fixed size (500) for each document, by padding or trimming articles when necessary.

5.2 Networks structure

Two Lstm recursive neural network models are trained. They are both stacked Lstm networks (networks with more than one Lstm layer). While the first one uses an embedding layer trained with the network, the second one uses Word2vec model for embedding (weights of the embedding layers are Word2vec vectors of the dictionary words). The structure of both networks provide 2 bidirectional Lstm layers and 2 dense layers that bring outputs dimensions to 32 and finally to 1 (where the latter is our prediction). The loss function chosen for the two models is the binary cross entropy.

5.3 Parameters

Batch size -A batch size of 512 has been choosen, given the results in the work in [10] that as tried 256, 512 and 1024 as batch sizes in Deep Learning tasks.

Epoch number - The number of epochs is decided programmatically during learning using the Keras Early Stopping [11] function. The maximum number of epochs is 5.

Lstm units - A number of 16 units per level is choosen for the two models.

Embedding dimensions - The Google Blog article in [12] tells that a good rule of thumb

for the number of embedding dimensions is to choose the 4^{th} root of the size of the dictionary. Since after stemming and stop word removal the total size of our dictionary is approximately 400,000, 32 was initially chosen as the number of dimensions. After several attempts, the value of 64 is finally chosen for both models.

Drop out rate - A drop out layer with a drop rate of 0.1 is used.

5.4 Learning phase

The two models described in the previous sections are trained. In Figure 3 and Figure 4 the metrics on test and training set are presented. Note how the precision and the recall formulation refer to true positive TPs, which are correct classifications of articles as fake news.

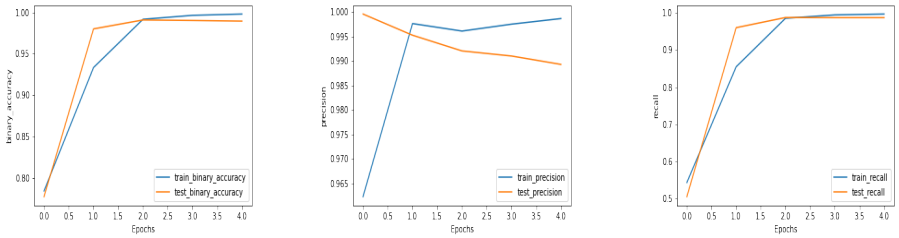

Figure 3: *Binary accuracy, precision, recall for model 1 (Lstm with Keras embedding).*

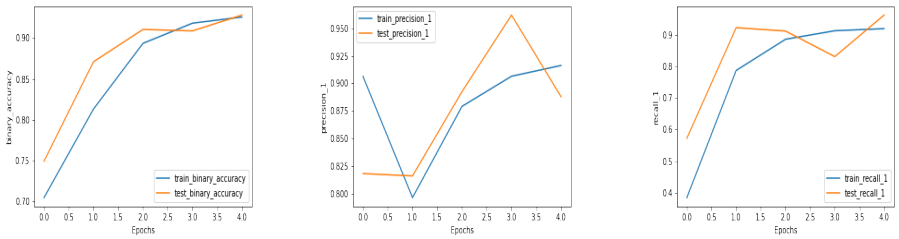

Figure 4: *Binary accuracy, precision, recall for model 2 (Lstm with Word2vec embedding).*

First of all, we note how the metrics, even if slightly, often have better results in the test set than in the training set. This seems a strange result as performance should be better in the training set than in the test set, since the model is "optimized" for the latter. This behavior is hypothesized to be due to regularization through drop out: during the training phase we do not use all the units of the network, hence the performances are slightly worse.

Looking at the test metrics in Figure 3 and Figure 4, it seems that the use of Word2vec for embedding did not lead to improvements in the model. In particular:

- Accuracy, precision and recall appear to have decreased.
- In the first model precision and recall have almost equal values; using Word2vec the recall is now greater than the precision. This indicates that, while without Word2vec the trustworthy articles classified as fake news (FP) are as many as the fake news articles classified as trustworthy (FN), with word2vec there are more FPs than FNs.

While it could be said that all metrics of the first model are better than the ones of the second model, models look suspiciously good. In particular, even if the first model accuracy is very close to 100% in training data, it does not seem to overfit as the accuracy continues to increase across epochs even in the test set.

These results raised some suspicions about the goodness of the database, hence the 2 models (together with the previously mentioned fine tuned Bert model) are tested on other datasets: datasets 1 and 2.

6 Bert model

The fine tuned Bert model used is a model called "roberta-fake-news-classification" [2] found on hugginface.co and published by user Hamza Benyamina. The model was released on March 29, 2022 and recorded a total number of 75 downloads in the month between June 11, 2022 and July 11, 2022.

The description of the model on hugginface.co indicates the dataset used for learning, which is the so called dataset 0 of this project. The description claims also that this model got an accuracy of 100% on it.

7 Models predictions

In this section the two Lstm models and the Bert model are compared. This comparison is done by evaluating the 3 models separately on datasets 2 and 3, providing accuracy, precision, recall and f1 as metrics in Table 4. Once again, precision and recall formulation refer to true positive TPs as correct classifications of fake news articles.

	Lstm 1 (Keras emb)		Lstm 2 (W2v emb)		Bert model	
Metric/Dataset	dataset 1	dataset 2	dataset 1	dataset 2	dataset 1	dataset 2
Accuracy	0.55	0.69	0.50	0.53	0.49	0.43
Precision	0.53	0.61	0.49	0.46	0.49	0.43
Recall	0.64	0.75	0.81	0.80	1.00	1.00
F1	0.57	0.67	0.61	0.58	0.66	0.60

Table 4: *Accuracy, precision, recall and f1 of dataset 1 and dataset 2 evaluations with the 3 models taken into account*

The results obtained seem to suggest that all the models compared were not able to generalize during learning. These results are much worse than those obtained on the test data of dataset 0. In particular, from Table 4, it can be noted that:

- the accuracy (the number of correct classifications with respect to the total) is about 0.5, with 2 exceptions:
  - The higher accuracy value is 0.69 from the first Lstm model on dataset 2;
  - The Bert model added to the comparison obtains accuracy values even lower than 0.5 (therefore with worse results than the majority baseline would have).
- For the two Lstm models, recall is consistently greater than accuracy: these models are "more inclined" to classify articles as fake news than as trustworthy one when they are wrong.

- Recall values for the Bert model are even equal to 1: this means that there were no fake news classified as trustworthy articles (number of FNs = 0).

However, as expected, results differ from model to model. McNemar's statistical test is now applied to the results obtained. The null hypothesis is "Marginal frequencies (sums of rows and columns of the contingency table) are the same". This hypothesis can also be formulated as "The 2 models disagree to the same amount (i.e. b=c in the contingency table)".

The test statistic is (1) and the contingency table is defined in Table 5:

$$\chi^2 = \frac{(|b-c|-1)^2}{b+c} \qquad (1)$$

	M2 Correct	M2 Wrong
M1 Correct	a	b
M1 Wrong	c	d

Table 5: *McNemar's Contingency Table*

The corresponding p-values are those in Tables 6.

Dataset 1			
Models	Lstm 1	Lstm 2	Bert
Lstm 1	-	<.001	<.001
Lstm 2	<.001	-	<.001
Bert	<.001	<.001	-

Dataset 2			
Models	Lstm 1	Lstm 2	Bert
Lstm 1	-	<.001	<.001
Lstm 2	<.001	-	<.001
Bert	<.001	<.001	-

Table 6: *All possible p-values for McNemar test. All values are extremely low and, as suggested by APA style guidelines [13], they are reported as less than 0.001. Tables are symmetric.*

The contingency table (Table 7) of the test between the two models with the most similar accuracy is also reported. This is also the test that has reported the lowest p-value (7×10^{-5}, all the others are lower). The two models are the Lstm with Word2Vec embedding and the already fine tuned Bert model; the target dataset is dataset 1.

	M2 Correct	M2 Wrong
M1 Correct	9839	2249
M1 Wrong	2523	10191

Table 7: *Contingency table of the McNemar test between the classifications of Lstm with Word2Vec and the Bert model on dataset 1.*

Using a threshold of 0.05 for the p-values of the McNemar tests, in all cases the null hypotesis is rejected. Thus, the test seems to suggest that the differences in accuracy values between the various models (compared on the same dataset) always reflect the presence of a model with better performance than the other.

After obtaining unpleasant results in the performance of the various models, it has been tried to analyze the textual content of the datasets in more detail. The purpose of the following analysis, which has a focus on dataset 0, is to understand the reason that led to the results just obtained.

8 In depth Data analysis

The results obtained so far have raised doubts about the dataset used for the training phase: it may not be representative of the nature of the problem. In this section some of the characteristics of all the 3 datasets are presented, trying to extrapolate information and criticalities.

8.1 Number of mentions

A first characteristic that could be captured from the data is the number of social media mentions that are present in all datasets. Under the hypotesis that each time an "@" is found then we have a social media mention, the number of these occurrences is obtained.

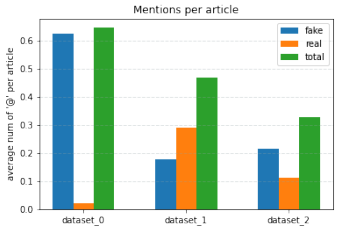

Figure 5: *Average mentions per article for each dataset.*

Dataset	Fake	Real	Total
0	24195	807	25002
1	4422	7232	11654
2	617	323	940

Table 8: *Total number of mentions for each dataset.*

What stands out from Figure 5 and Table 8 are the values obtained for dataset 0: while having the highest number of mentions per article (on average more than 1 article out of 2 has a mention), it has many more mentions in fake articles than in the real ones (despite having an almost equal number of fake news and real articles).

8.2 Number of links

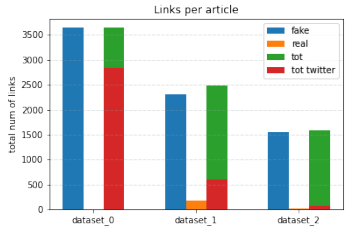

Figure 6: *Total number of links for each dataset.*

Dataset	Fake	Real	Total
0	3645 (2832)	0 (0)	3645 (2832)
1	2306 (594)	176 (27)	2482 (621)
2	1558 (75)	22 (14)	1580 (89)

Table 9: *Total numbers of links (the numbers of Twitter links are highlighted in brackets)*

The total number of links in the body of each article has also been obtained for each dataset. In Figure 6 and in Table 9 results can be seen. It is evident that in all datasets articles classified as trustworthy have in total a very small number of links (even 0 for the first one).

In addition, these numbers have been filtered searching only for Twitter links. It is interesting to note that in dataset 0, 76% of links are Twitter links.

These results, in addition to the previous results on number of mentions, suggest that a lot of fake news articles could contains one or more tweets. Therefore being a fake news article in this dataset seems to be strictly linked to report tweets, hence having (at least in part) a "tweet-like" content.

8.3 Word clouds and words occurrences

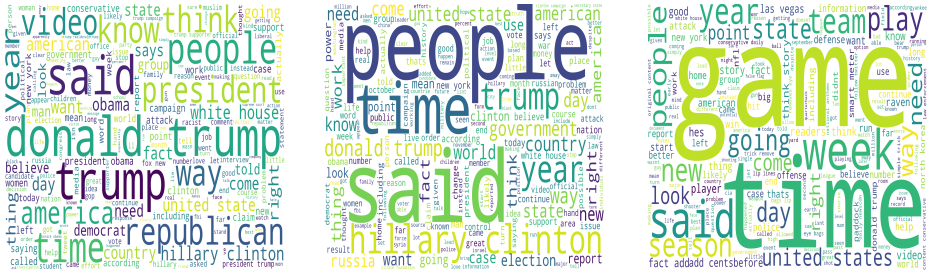

Figure 7: *Fake news articles word clouds. From left: dataset 0, dataset 1, dataset 2*

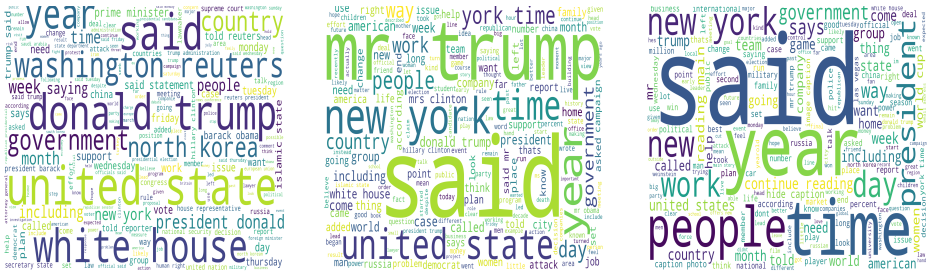

Figure 8: *Trustworthy articles word clouds. From left: dataset 0, dataset 1, dataset 2*

In Figure 7 and Figure 8 we can see word clouds for each dataset. It seems that there are no distinctive features between trustworthy articles and fake news articles based on the context of the words. Anyway, it seems that a lot of the most frequent occurrences regard specific contexts or periods, such as "donald trump", "white house" or "republican".

Finally, among the most recurring words of dataset 0, the most unbalanced ones have been found. With the term "unbalanced word" we refer to all those words whose occurrences are many more in fake news articles than in trustworthy articles or vice versa. Among these words, 4 of them have been chosen and their occurrences have been reported in Table 10.

	"said"		"reuters"	
Dataset	Fake	Real	Fake	Real
0	24,841 (20%)	97,866 (80%)	319 (1%)	28792 (99%)
1	15,356 (15%)	85,457 (85%)	527 (60%)	350 (40%)
2	994 (11%)	7,797 (89%)	36 (2%)	1532 (98%)

	"eu"		"image"	
Dataset	Fake	Real	Fake	Real
0	4808 (9%)	46583 (91%)	14274 (95%)	683 (5%)
1	10464 (43%)	14062 (57%)	1777 (48%)	1962 (52%)
2	636 (16%)	3372 (84%)	201 (13%)	1339 (87%)

Table 10: *Occurrences in datasets of 4 of the most unbalanced words found.*

The results in Table 10 show that the 4 words are extremely unbalanced in dataset 0. In particular, 99% of the occurrences of the word "reuters" come from trustworthy articles. Not surprisingly, Reuters.com is the news website from which the authors of [4] have taken the trustworthy articles.

Another interesting fact is that even for dataset 2 "reuters" is extremely more frequent in trustworthy articles. This suggests that this dataset (which has unknown origins) was also built with Reuters.com articles. Looking at the provenance of some of his articles, this hypothesis is confirmed.

9 Extraordinary models

To make clear the lack of goodness of the dataset used for training, 3 extremely simple models with exceptional performance are trained. These models are SVM models with linear kernel. Bag of Words is used for feature extraction. In particular:
Model 1 - The first model has 3 features corresponding to the number of occurrences of the words "said", "eu", "image".
Model 2 - The second model is the same as model 1, without the feature related to the word "image".
Model 3 - The third model has only 1 feature, the number of occurrences of the word "reuters".

The models were trained with the same 80% of dataset 0 used for previous models.

Model	Accuracy	Precision	Recall	F1
Model 1	0.94	0.98	0.87	0.92
Model 2	0.87	0.83	0.88	0.86
Model 3	0.99	1	0.99	0.99

Table 11: *Metrics obtained from the evaluation of model 1,2 and 3 on the test set.*

Table 11 shows the metrics obtained from the predictions of the 3 models on the remaining 20% of the dataset 0, the test set. As always, TPs are correct classifications of fake news articles. Results are surprising: Model 1, with only 3 words used, boasts an accuracy of 0.94. It is interesting to note that even removing the "image" feature, the measured accuracy is 0.87 (Model 2). However, the most disconcerting results are the ones of Model 3: with a single feature we have an accuracy of 0.99!

10 Conclusions

The goal of obtaining a prediction model for this categorization problem has not been very successful. After an initial euphoria given by the excellent results obtained from the tests during learning, the comparison between the models highlighted the problems of them.

Although the models have been compared, it was not possible to reach conclusions on which is the best on the problem under consideration. This is due to the results obtained from the analysis of the dataset used for training. These results lead to the conclusion that the training dataset has guided the models to distinguish differences between fake news and trustworthy articles typical of that dataset only.

To better highlight the criticalities of the training dataset, three simple SVM models were trained. Using BoW with small sets of features (even only one) they have obtained outstanding results.

It is therefore concluded that the dataset used to train the models in this analysis is not representative of reality. Lastly, it is reported a passage from the work in [14], which outline the difficulties of the problem taken into consideration:

> A general challenge of content-based methods is that fake news's style, platform, and topics keep changing. Models that are trained on one dataset may perform poorly on a new dataset with different content, style, or language.

11 Future works

The results of the analysis of the training dataset may lead to reflect on the fact that collecting and categorizing articles is not a trivial job: the collection and the classification of each article comes from a manual job or from an act of faith towards the source, and can potentially lead to a biased dataset. A problem to consider in future works would therefore be to find a sufficiently rigorous definition of a fake news article that can be applied with consistency to any article, in order to be able to build a balanced dataset that is representative of the reality.

References

[1] Sahil Loomba, Alexandre de Figueiredo, Simon J. Piatek, Kristen de Graaf, and Heidi J. Larson. Measuring the impact of covid-19 vaccine misinformation on vaccination intent in the uk and usa. *Nature Human Behaviour*, 5(3):337–348, Mar 2021.

[2] Hamza Benyamina. roberta-fake-news-classification. https://huggingface.co/hamzab/roberta-fake-news-classification, 2022.

[3] Thomas G. Dietterich. Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms. *Neural Computation*, 10(7):1895–1923, 10 1998.

[4] 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.

[5] clmentbisaillon. Fake and real news dataset - https://www.kaggle.com/datasets/clmentbisaillon/fake-and-real-news-dataset, 2020.

[6] UTK Machine Learning Club. https://www.kaggle.com/c/fake-news/overview, 2018.

[7] jruvika. fake-news-detection. https://www.kaggle.com/datasets/jruvika/fake-news-detection, 2021.

[8] Radim Rehurek and Petr Sojka. Gensim–python framework for vector space modelling. *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic*, 3(2), 2011.

[9] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.", 2009.

[10] Leslie N. Smith. A disciplined approach to neural network hyper-parameters: Part 1 – learning rate, batch size, momentum, and weight decay, 2018.

[11] Francois Chollet et al. Keras, 2015.

[12] Google Blogs. Introducing tensorflow feature columns. https://developers.googleblog.com/2017/11/introducing-tensorflow-feature-columns.html, 2017.

[13] APA. Apa style, numbers and statistics guide. Technical report, 02 2022.

[14] Shaina Raza and Chen Ding. Fake news detection based on news content and social contexts: a transformer-based approach. *International Journal of Data Science and Analytics*, 13(4):335–362, May 2022.