BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Excel@FIT 2023
http://excel.fit.vutbr.cz

Detection of Fake News Using Machine Learning

Matej Koreň*

Abstract

This project focuses on the use of machine learning in fake news detection. For this purpose, four models have been selected - Bayesian, Decision Tree, Support Vector Machine and a Neural Network. In five experiments on various datasets, these models were trained, tested, evaluated and compared with state-of-the-art methods. Final implementation is in the form of a console application, which allows it's users to replicate this procedure with their own data. Beyond the assignment, Slovak dataset **Dezinfo SK** was created. The purpoue of its creation was the lack of available datasets in slovak language. It also serves as a demonstration of robustness for models used in this case as in the meaning of using various other languages. Dataset was created manually with the intention to sustain objectivity and it's overall meaningfulness. It is available online to be used for model training for free.

*xkoren10@stud.fit.vutbr.cz, *Faculty of Information Technology, Brno University of Technology*

1. Introduction

[Motivation] Recently, the issue of fake news has become increasingly worrying for society. In a time when social media has a significant impact on how people make decisions (especially in politics), it is important to carefully consider the credibility of information that is being spread on them. Of course, not every piece of information is easily verifiable and often cannot be verified at all. One proposal to solve the problem of the large amount of data that needs to be verified is to classify them as fake or true based on common parameters of individual groups. This is where the performance of computers combined with the ingenuity of humans - machine learning - comes into play.

[Existing solutions] Among the most commonly used and successful models for text classification are pre-trained deep machine learning models based on transformers, such as **BERT**, **RoBERTa**, or **XLNET**. **BERT**, or Bidirectional Encoder Representations from Transformers in its full name, was created in 2018 by Google. It is based on a deep learning model where the weight of each output is dynamically re-calculated as input and allows for bidirectional passage through the analyzed text. Transformers allow this model to be trained on a larger volume of data than other models.

[Our solution] Despite the obvious superiority of large pretrained models, some mathematical models are still used for this purpose due to their simplicity – for example, Bayesian models or Support Vector Machines. Based on this observation, the question arises as to whether they are still relevant to use and whether it is even possible to achieve comparable efficiency compared to neural networks and transformers.

2. Impelementation

The Python 3.9 language was chosen as the development environment, which is very suitable thanks to the large number of libraries for machine learning, text processing, or working with a graphical user interface. There is also enough online documentation and sample code available, so their use is clear and makes work easier. Specifically, the following libraries are used in the project :

- pandas - data manipulation and analysis [1]
- sklearn - models, vectorisers, scoring [2]
- matplotlib - graphs [3]
- nltk - Natural Language Processing library [4]
- other (numpy, sys, string, seaborn)

[1] https://pandas.pydata.org/
[2] https://scikit-learn.org/stable/index.html
[3] https://matplotlib.org/
[4] https://www.nltk.org/

The program is divided into modules that are called from the main program - Text Processing, individual models, and the end interface. In addition to the code part, pre-trained models, datasets, and auxiliary files used with them are stored.

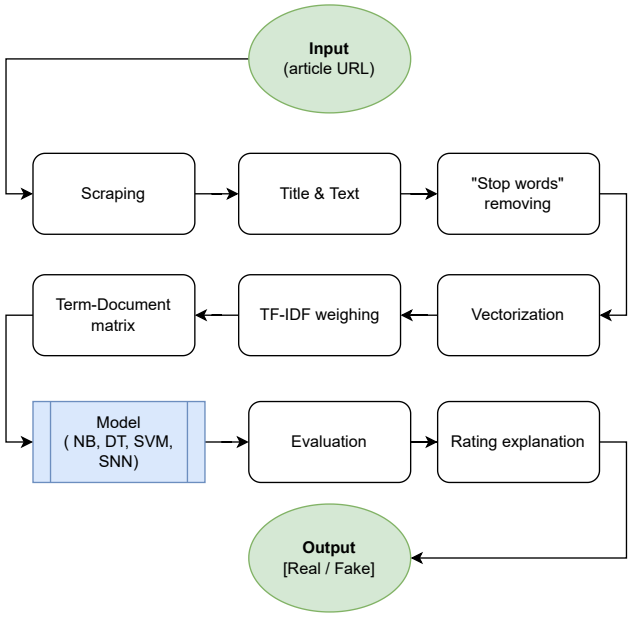

Figure 1. Diagram of program data pipeline from input to output

3. Training and testing

Various datasets from `kaggle.com` and previously mentioned Dezinfo SK dataset were used as a training and testing set for Naive Bayes, Decision Tree, SVM, and Sequential Neural Network models. The results were evaluated using the F1-score and are shown in the table below:

Used model	Precision (Fake/True)	Recall (Fake/True)	F1-score (Fake/True)	Accuracy
NB	0.76/0.92	0.93/0.75	0.84/0.83	0.83
SVM	0.94/0.92	0.94/0.92	0.94/0.92	0.93
DT	0.65/0.85	0.85/0.65	0.73/0.73	0.73
SNN	0.94/0.85	0.89/0.92	0.91/0.88	0.90

Table 1. Model scoring in the experiment using Dezinfo SK dataset

the results suggest that the structure and content of the data is appropriate and the models are trained adequately. The resulting models were further tested on articles from the internet outside of those included in the original dataset:

Used model	True	False
NB	55.53%	44.47%
DT	68.75%	31.25%
SVM	86.09%	13.91%
SNN	76.30%	23.70%

Table 2. Evaluation of an article from TA3

Used model	True	False
NB	36.61%	63.39%
DT	36.36%	63.64%
SVM	1.29%	98.71%
SNN	2.56%	97.44%

Table 3. Evaluation of an article from Badatel.net

These results are matching the expected classes of the articles. For reference, ChatGPT 3.5. with its BERT model classified the first article as "True" (90%) and the second article as "False" (%80).

4. Conclusions

These experiments are showing that despite the growing popularity of pre-trained models, transformers and other vast machine learning principles used for text classification, simple mathematical models are still relevant. However, it all comes down to the quality of the training datasets. Another factor, which is very important in this topic, is memory efficiency and time efficiency compared to overall results. If the circumstances allow it, bigger models are far superior than those used and they are also being optimalised day by day.

Acknowledgements

I would like to thank my supervisor Ing. David Hříbek for his help and cooperatiton during the whole semester.