Fake news detection problem

Igor Gaidamaka

May 2020

Abstract

Everyday we accessing media outlets such as news blogs, social media feeds, and online newspapers have made it challenging to identify trustworthy news sources, thus increasing the need for computational tools able to provide insights into the reliability of online content.

I'm conducted a set of learning experiments to build accurate fake news detector, and showing accuracy i can achieved.

`https://github.com/Chawalar/Fake-news-detection`.

1 Introduction

In recent years, due to the rapid development of online social media, more and more people tend to seek out and obtain news from it than from traditional media. It gives fake news a lot of chance to spread. Compared with traditional suspicious information such as email spam and web spam, fake news has much worse societal impact. Because fake news spreads faster and broader. Traditional suspicious information often targets specific recipients and only produces a local impact. However, online fake news can disseminate exponentially, affecting more people.

Disseminate news online faster and easier through social media and online news sites, large amounts of disinformation such as fake news, i.e., those news articles with intentionally false information, are produced online for a variety of purposes, ranging from financial to political gains. In our case we take fake news articles as an example of disinformation. The extensive spread of fake news can have severe negative impacts on individuals and society.

The recent proliferation of social media has significantly changed the way in which people acquire information. Fake news, which refer to intentionally and verifiable false news stories, can spread virally on social media platforms as people rarely verify the source of the news when sharing a news article that sounds true. The spread of fake news may bring many negative impacts, including social panic and financial loss. Recent years have witnessed a number of high-impact fake news spread regarding terrorist plots and attacks, presidential election, and various natural disasters. In many of these cases, even when correct information later disseminates, the rapid spread of fake news can have

devastating consequences. Therefore, there is an urgent need for the development of automatic fake news detection algorithms which can detect fake news as early as possible and help to stop the viral spread of such news.

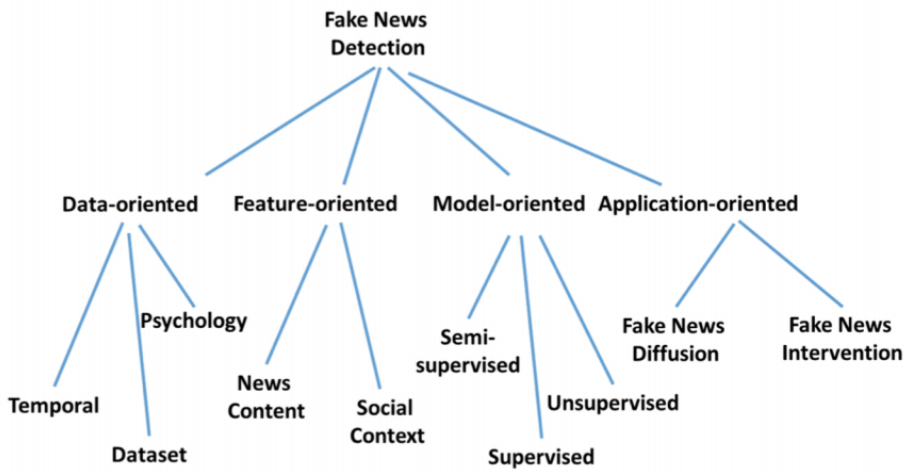

Figure 1: Different approaches to fake news detection.

2 Related Work

Fake news detection has been studied in several investigations. Conroy et al.[1] presented an overview of deception assessment approaches, including the major classes and the final goals of these approaches. They also investigated the problem using two approaches: (1) linguistic methods, in which the related language patterns were extracted and precisely analyzed from the news content for making decision about it, and (2) network approaches, in which the network parameters such as network queries and message metadata were deployed for decision making about new incoming news.

Ruchansky et al. [2] proposed an automated fake news detector, called CSI that consists of three modules: Capture, Score, and Integrate, which predicts by taking advantage of three features related to the incoming news: text, response, and source of it. The model includes three modules; the first one extracts the temporal representation of news articles, the second one represents and scores the behavior of the users, and the last module uses the outputs of the first two modules (i.e., the extracted representations of both users and articles) and use them for the classification. Their experiments demonstrated that CSI provides an improvement in terms of accuracy.

Tacchini et al. [3] introduced a new approach which tries to decide if a news

is fake or not based on the users that interacted with and/or liked it. They proposed two classification methods. The first method deploys a logistic regression 3 model and takes the user interaction into account as the features. The second one is a novel adaptation of the Boolean label crowdsourcing techniques. The experiments showed that both approaches achieved high accuracy and proved that considering the users who interact with the news is an important feature for making a decision about that news.

Prez-Rosas et al. [4] introduced two new datasets that are related to seven different domains, and instead of short statements containing fake news information, their datasets contain actual news excerpts. They deployed a linear support vector machine classifier and showed that linguistic features such as lexical, syntactic, and semantic level features are beneficial to distinguish between fake and genuine news. The results showed that the performance of the developed system is comparable to that of humans in this area.

Wang [5] provided a novel dataset, called LIAR, consisting of 12,836 labeled short statements. The instances in this dataset are chosen from more natural contexts such as Facebook posts, tweets, political debates, etc. They proposed neural network architecture for taking advantage of text and metadata together. The model consists of a Convolutional Neural Network (CNN) for feature extraction from the text and a Bi-directional Long Short Term Memory (BiLSTM) network for feature extraction from the meta-data and feeds the concatenation of these two features into a fully connected softmax layer for making the final decision about the related news. They showed that the combination of metadata with text leads to significant improvements in terms of accuracy.

Long et al. [6] proved that incorporating speaker profiles into an attention-based LSTM model can improve the performance of a fake news detector. They claim speaker profiles can contribute to the model in two different ways. First, including them in the attention model. Second, considering them as additional input data. They used party affiliation, speaker location, title, and credit history as speaker profiles, and they show this metadata can increase the accuracy of the classifier on the LIAR dataset.

Ahmed et al. [7] presented a new dataset for fake news detection, called ISOT. This dataset was entirely collected from real-world sources. They used 4 n-gram models and six machine learning techniques for fake news detection on the ISOT dataset. They achieved the best performance by using TF-IDF as the feature extractor and linear support vector machine as the classifier.

Wang et al. [8] proposed an end-to-end framework called event adversarial neural network, which is able to extract event-invariant multi-modal features. This model has three main components: the multi-modal feature extractor, the fake news detector, and the event discriminator. The first component uses CNN as its core module. For the second component, a fully connected layer

with softmax activation is deployed to predict if the news is fake or not. As the last component, two fully connected layers are used, which aims at classifying the news into one of K events based on the first component representations.

Tschiatschek et al. [9] developed a tractable Bayesian algorithm called Detective, which provides a balance between selecting news that directly maximizes the objective value and selecting news that aids toward learning user's flagging accuracy. They claim the primary goal of their works is to minimize the spread of false information and to reduce the number of users who have seen the fake news before it becomes blocked. Their experiments show that Detective is very competitive against the fictitious algorithm OPT, an algorithm that knows the true users parameters, and is robust in applying flags even in a setting where the majority of users are adversarial.

References

[1] Conroy, N. J., Rubin, V. L., & Chen, Y. (2015). Automatic deception detection: Methods for finding fake news. In Proceedings of the Association for Information Science and Technology (pp. 1–4). volume 52.

[2] Ruchansky, N., Seo, S., & Liu, Y. (2017). Csi: A Hybrid Deep Model for Fake News Detection. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, ACM (pp. 797–806).

[3] Tacchini, E., Ballarin, G., Vedova, M. L. D., Moret, S., & de Alfaro, L. (2017). Some Like It Hoax: Automated Fake News Detection in Social Networks. arXiv preprint arXiv:1704.07506.

[4] Prez-Rosas, V., Kleinberg, B., Lefevre, A., & Mihalcea, R. (2018). Automatic Detection of Fake News. In Proceedings of the International Conference on Computational Linguistics, (p. 33913401).

[5] Wang, W. Y. (2017). " Liar, Liar Pants on Fire": A New Benchmark Dataset for Fake News Detection. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (p. 422426).

[6] Long, Y., Lu, Q., Xiang, R., Li, M., Huang, C.-R. (2017). Fake news detection through multi-perspective speaker profiles. In Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers) (pp. 252–256).

[7] Ahmed, H., Traore, I., & Saad, S. (2017). 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.

[8] Wang, Y., Ma, F., Jin, Z., Yuan, Y., Xun, G., Jha, K., Su, L., & Gao, J. (2018). Eann: Event Adversarial Neural Networks for Multi-Modal Fake News Detection. In Proceedings of the 24th acm sigkdd international conference on knowledge discovery data mining, ACM (pp. 849–857).

[9] Tschiatschek, S., Singla, A., Rodriguez, M. G., Merchant, A., Krause, A. (2018). Fake News Detection in Social Networks via Crowd Signals. In Companion Proceedings of the The Web Conference (pp. 517–524).

3 Model Description

In my approach i used machine learning algorithms to get baselines, then i used main model named BERT to get best accuracy score.

For the baseline models will be used: Passive Aggressive Classifier, Naive Bayes, Logistic Regression, Random Forest, XGBoost, with different methods of text prepossessing and TF-IDF transform. This classic machine learning algorithms will be show basic accuracy score i can achieve on fake news classification task. Then i use main model (BERT) to get more accuracy.

BERT (Devlin et al., 2018), which stands for Bidirectional Encoder Representations from Transformers, is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. BERT uses the "masked language model" (MLM) pre-training objective, inspired by the Cloze task (Taylor, 1953). The masked language model randomly masks some of the tokens from the input, and the objective is to predict the masked word based on its context. In addition to the masked language model, BERT also uses a "next sentence prediction" (NSP) task that jointly pre-trains text-pair representations. For the pre-training corpus (Devlin et al., 2018) use the BooksCorpus (800M words) and English Wikipedia (2,500M words). The self-attention mechanism in the Transformer allows BERT to model many downstream tasks — whether they involve single text or text pairs. For each task, the steps are: (1) simply plug in the task-specific inputs and outputs into BERT and (2) fine-tune all the parameters end-to-end.

With news articles classification task for finding fakes i want to understood problem with all sides. First of all i'm using machine learning methods for building fake news detector on my data, second is going deeper to this problem in word wide and looking for some things to getting this on new level.

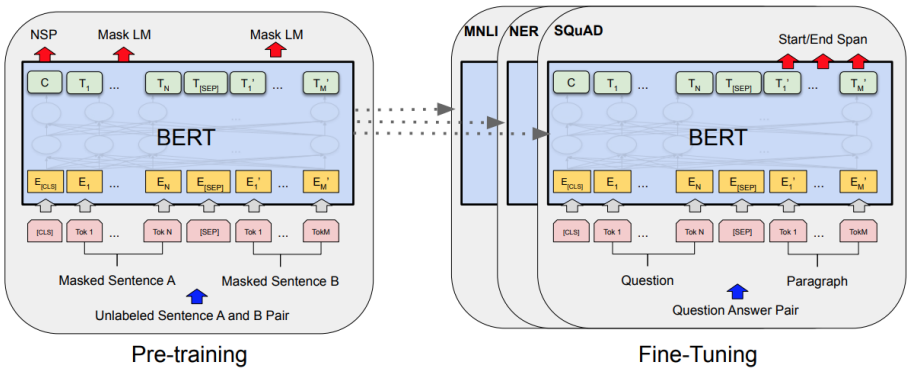

Figure 2: Overall pre-training and fine-tuning procedures for BERT. Apart from output layers, the same architectures are used in both pre-training and fine-tuning. The same pre-trained model parameters are used to initialize models for different down-stream tasks. During fine-tuning, all parameters are fine-tuned. [CLS] is a special symbol added in front of every input example, and [SEP] is a special separator token (e.g. separating questions/answers).

For our experiments on fake news classification task i used pre-trained BERT by Hugging Face Transformers library and fine-tuned it on scraped news articles dataset.

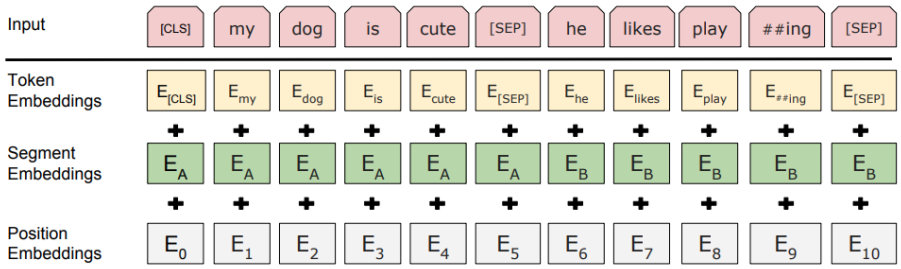

Figure 3: BERT input representation. The input embeddings are the sum of the token embeddings, the segmentation embeddings and the position embeddings.

Fine-tuning is straightforward since the selfattention mechanism in the Transformer allows BERT to model many downstream tasks— whether they involve single text or text pairs—by swapping out the appropriate inputs and outputs.

For classification tasks, we must prepend the special [CLS] token to the beginning of every sentence. This token has special significance. BERT consists of 12 Transformer layers. Each transformer takes in a list of token embeddings, and produces the same number of embeddings on the output.

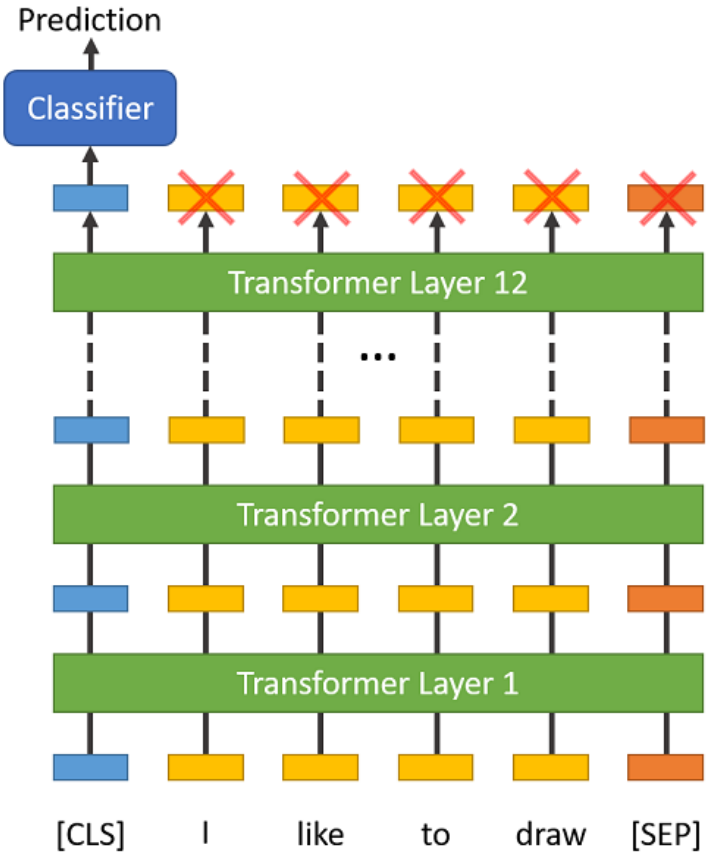

Figure 4: On the output of the final (12th) transformer, only the first embedding (corresponding to the [CLS] token) is used by the classifier.The first token of every sequence is always a special classification token ([CLS]). The final hidden state corresponding to this token is used as the aggregate sequence representation for classification tasks.

BERT has a max length limit of tokens - 512, so in our case we will be used this limit and set max length to first 512 tokens. For else approaches we can use different way of choosing tokens from large text, but in our case its works well.

4 Dataset

Collecting data of news articles will be doing by web scraping popular news sites. In kind of this task i tested many approaches to getting information about fake

news by myself. Reading and labeling all articles, checking facts and etc by yourself its really hard and spending more time, so because i working on this alone i getting off this idea. I'm checked hundreds sites labeled like fake news or spam (more of this are currently didn't work) i found some nowadays worked sites and scrapped articles from it RSS. In the end i worked with two dataset, political fake news and finance/business.

	Train	Valid	Test
Articles	5454	303	303
Vocabulary size		6060	

Table 1: Statistics political dataset, with balance classes, without duplicates articles.

Second dataset scraped from 25 news sites, where 5 are sites with fake news articles. This articles will be classify like fakes. Scraped articles in dataset have links, published date, title, text, author, and after preprocessing also label.

	Train	Valid	Test
Articles	4344	543	543
Vocabulary size		5430	

Table 2: Business/finance dataset with class imbalance, with preprocessing and without duplicates.

I iterate through each news site, getting the page and downloading article and some other information to json file, then preprocessing text.

5 Experiments

Overall i perform four sets of experiments. In the following sub-sections i describe and analyze them.

5.1 Metrics

The main chosen metric its accuracy, but for better understanding prediction process i also suggest to look at precision, recall and F1-score, its important in our second dataset because of class imbalance.

Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations. One may think that, if we have high accuracy then our model is best. Yes, accuracy is a great measure but only when you have symmetric datasets where values of false positive and false negatives are almost same.

$$Accuracy = TP + TN/TP + FP + FN + TN,$$

Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. High precision relates to the low false positive rate.

$$Precision = TP/TP + FP,$$

Recall is the ratio of correctly predicted positive observations to the all observations in actual class - real.

$$Recall = TP/TP + FN,$$

F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy, but F1 is usually more useful than accuracy, especially if you have an uneven class distribution. Accuracy works best if false positives and false negatives have similar cost. If the cost of false positives and false negatives are very different, it's better to look at both Precision and Recall.

$$F1Score = 2 * (Recall * Precision)/(Recall + Precision),$$

5.2 Experiment Setup

Starting point its this machine learning algorithms: Passive Aggressive Classifier, Naive Bayes, Logistic Regression, Random Forest, XGBoost, with text preprocessing and TF-IDF transform.

Text preprocessing its always important part. I want to list the methods that i used for our experiments: lower casing, removal of punctuation, removal of stopwords, removal frequent/rare words, steamming, removal url and HTML tags, chat words conversation, spelling correction.

After text preprocessing i split the data with train test split method with test size = 0.2, then transform it with TF-IDF and train my classifiers.

For second dataset with class imbalance we can use oversampling to get better results.

BERT model doing preprocessing by bert tokenizer so usually we don't get benefit from standard preprocessing tips. It's can be useful to pass through a contraction corrector to replace misspelled words with correction. Training parameters: max length 512, batch size 16, learning rate 2e-5, number of epochs 20. Without hyperparameters tuning.

5.3 Baselines

One of the best baselines for fake news classification task what i found its Logistic Regression over TF-IDF embedding and multinomial Naive Bayes also

like Passive Aggressive Classifier. It's works fast and get really fine results on classification task with text preprocessing.

6 Results

Machine learning algorithms parameters for baseline approach

- Passive Aggressive Classifier (max_iter = 50)
- Naive Bayes
- Logistic Regression (solver = saga, penalty = l1)
- Random Forest (n_estimators = 100)
- XGBoost

6.1 Political dataset

Accuracy of algorithms used with already preprocessed text and transformed to TF-IDF on political dataset:

Algorithms	Accuracy	Training time
Passive Aggressive Classifier	93.45 %	103ms
Naive Bayes	95.21 %	312ms
Logistic Regression	90.13 %	3.7s
Random Forest	89.98 %	2.05s
XGBoost	88.87 %	30.4s

It's really good results. Then let's look at BERT model. BERT pre-trained model by Hugging Face fine-tuned on our political dataset with training parameters: max length 512, batch size 16, learning rate 2e-5, number of epochs 20.

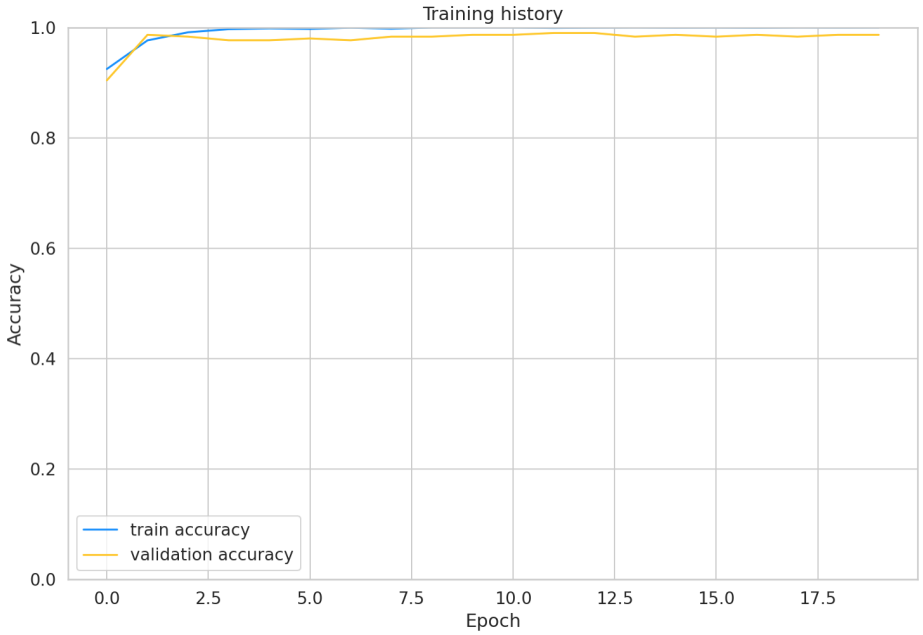

Figure 5: BERT training process.

Then look at our metrics:

	precision	recall	f1-score	support
true	0.98	1.00	0.99	157
false	1.00	0.97	0.99	146
accuracy			0.99	303
macro avg	0.99	0.99	0.99	303
weighted avg	0.99	0.99	0.99	303

Figure 6: BERT classification report.

This dataset with balanced classes, precision, recall and f1-score looks good. BERT achieved best score on this task in our models.

For better visibility let's look at confusion matrix:

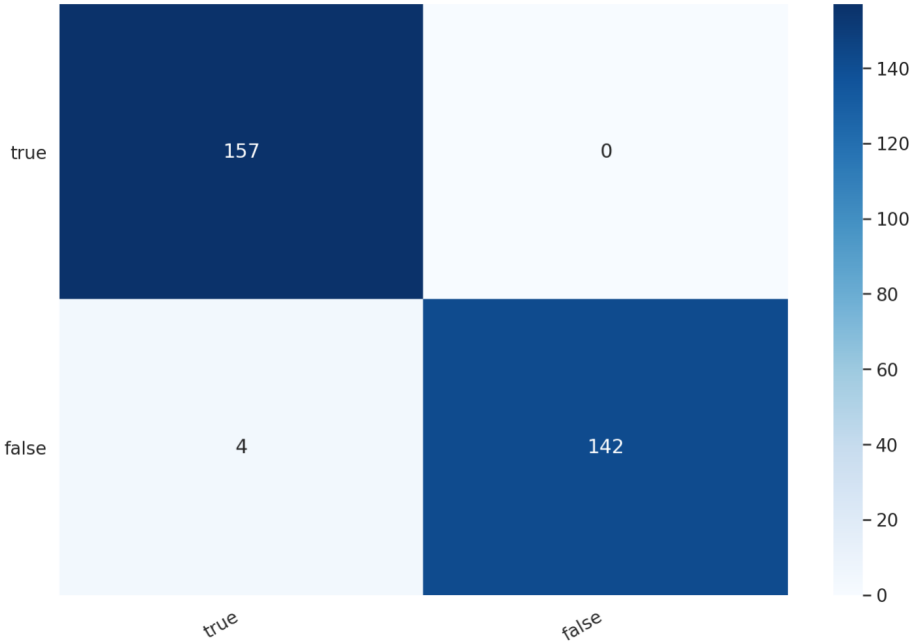

Figure 7: BERT confusion matrix.

Accuracy on test dataset: 0.9867, training time on Google Colaboratory: 3h 20min 37s with Tesla T4.

6.2 Business/finance dataset

Let's take the second dataset with business/finance news articles and imbalanced classes.

Accuracy of algorithms used with already preprocessed text and transformed to TF-IDF on business/finance dataset:

Algorithms	Accuracy	Training time
Passive Aggressive Classifier	93.53 %	106ms
Naive Bayes	99.13 %	38.3ms
Logistic Regression	80.22 %	4.95s
Random Forest	91.0 %	1.99s
XGBoost	89.34 %	35.8s

As we can see from scores with this algorithms, logistic regression accuracy drops, but naive bayes show great result.

Then fine-tune our BERT pre-trained model on second dataset with training parameters: max length 512, batch size 16, learning rate 2e-5, number of epochs 20.

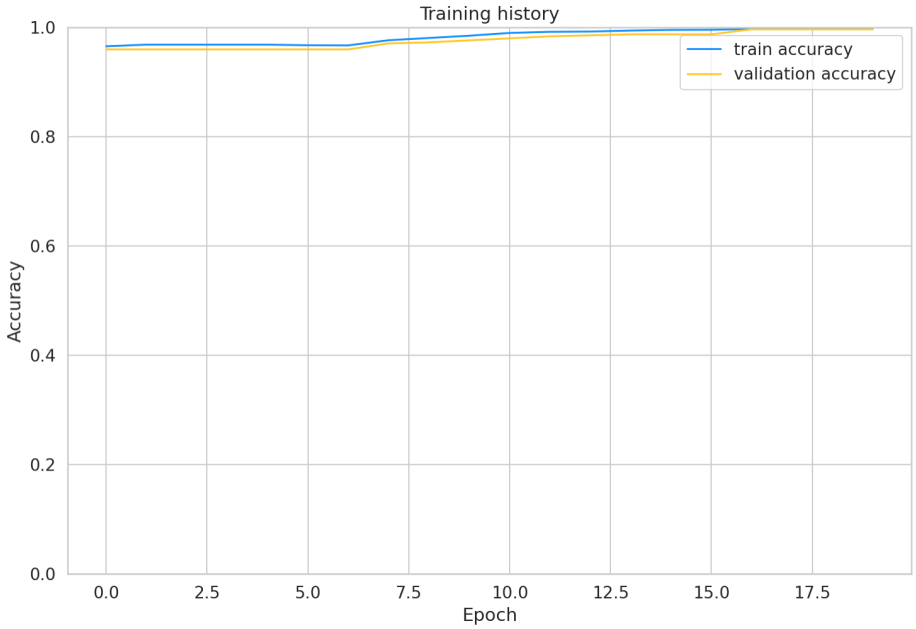

Figure 8: BERT training process.

Then look at our metrics:

	precision	recall	f1-score	support
true	1.00	1.00	1.00	530
false	0.92	0.85	0.88	13
accuracy			0.99	543
macro avg	0.96	0.92	0.94	543
weighted avg	0.99	0.99	0.99	543

Figure 9: BERT classification report.

This dataset with imbalanced classes, precision, recall and f1-score slightly drops. BERT also achieved best score on this task in our models.

For better visibility let's look at confusion matrix:

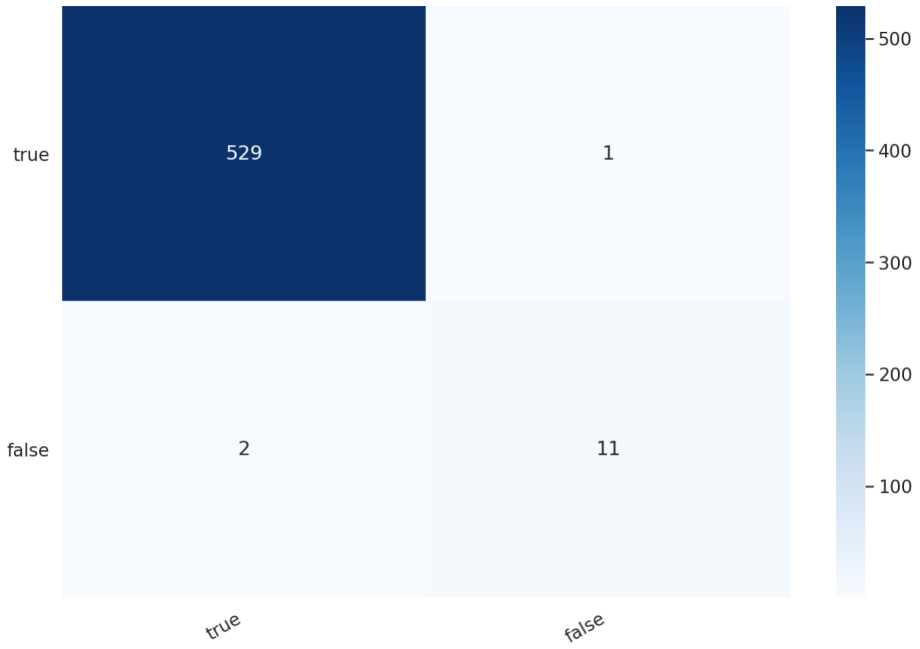

Figure 10: BERT confusion matrix.

Accuracy on test dataset: 0.9944, training time on Google Colaboratory: 1h 18min 37s with Tesla P100.

7 Conclusion

In this paper, we have studied in the fake news article classification problem. We have analysed how traditional machine learning algorithms works on two different datasets, the second being imbalanced. Results be improved by using BERT model fine-tuned on our datasets.

This approaches for fake news classification task looks good. Of course we can do better, for example by scraping more articles, hyperparameters tuning, using oversampling, data augmentation technique, like SMOTE, but i prefer to look at this task little wide and get some tips for future work.

Main idea of this project it's understanding fake news classification problem, not the getting best accuracy on one dataset, as we can see, classification task on our datasets doing well. Detection articles style on one topic or fake news can appear in well written articles and vice versa. Human language is nuanced. Determining a single statement as true or false. No databases of what's true or false. Many facts in a single article existing on all sides of the truth spectrum

— is that article true or false? I want to share thoughts for future work, for creating a system for fake news classification at all topics.

Here are some of them:

- Title classification for clickbaits.
- Author classification with rating database.
- Newspaper resource classification on media bias or something like it.
- Images classification on fake/true.
- Fact cheking model on database with facts.

For all of this we need to collect big databases of actual information and labeled it, its a lot of work. But only a merged methods can solve a problem like fake news classification in wide spectre.