Fake News Detection
(NLP 2021 Course)

Assel Yermekova
Assel.Yermekova@skoltech.ru

Mikhail Kuimov
Mikhail.Kuimov@skoltech.ru

Varvara Shushkova
Varvara.Shushkova@skoltech.ru

Mikhail Filitov
Mikhail.Filitov@skoltech.ru

Abstract

Nowadays, the social media becomes more and more powerful tool for shaping public opinion. That is why it's becoming so important to be able to recognize if article is fake or not. Also, due to current pandemic situation it became crucial to avoid spreading disinformation. This article is inspired by another paper (Pelrine et al., 2021) and provides a comprehensive analysis on the various natural language processing models based on transformer models. We compare their performance on datasets of different size and structure and provide the results of different experiments, including two preprocessing approaches testing and cross-dataset tests.

1 Introduction

Social media is a place where anyone can share personal opinion, views or insights. And because of such an open format it is crucial to be able to filter or classify classify news to make it possible to spotlight misinformation. It is important because fake news and misinformation in general may lead to financial losses, people deaths and so on. In current pandemic situation fake news and tutorials may cost people's lives. And that is why we need to develop quick and effective way to classify news.

In this paper we show the results of popular pre-trained NLP model comparison such that bert-base-uncased (Devlin et al., 2018) bert_uncased_L-2_H-128_A-2 (Turc et al., 2019) and others. We use these pre-trained models along with datasets of different sizes and different sources. Original paper proposes that this approach can reach or beat state-of-the-art solutions in some cases.

Also we aim to develop new approaches of pre-processing data and try to improve the quality of prediction of described models.

2 Related Work

Before the discussion of the detection of misinformation, we need to provide an overview of various types of false information that can be found online, and introduce some definition.

The most common terms in mainstream media are fake news and rumours, but however researchers have also analyzed other aspects related to misinformation on the web, such as clickbait, social spam and fake reviews (Bondielli and Marcelloni, 2019).

The definition of fake news proposed by (Allcott and Gentzkow, 2017) as "a news article that is intentionally and verifiably false" have adopted by several recent studies, such as (Conroy et al.). Such definition hinges on two key aspects: intent and verifiability. Fake news are therefore news articles that are intentionally written to mislead or misinform readers, but can be verified as false by means of other sources.

To provide a formal definition of rumour is not straightforward. In fact, researchers have reported different interpretations (Difonzo and Bordia, 2007), (Zubiaga et al., 2015),(Zubiaga et al., 2017), (Cai et al., 2014), (KNAPP, 1944). One of the most widely adopted definitions comes from the authors in (Difonzo and Bordia, 2007). In their research, rumours are identified as "unverified and instrumentally relevant information statements in circulation".

Clickbait refers to article titles or social media posts whose aim is to attract readers to follow a link to the actual article page (Chen et al., 2015). In addition, clickbait headlines have been identified as one of the major contributors to the spread of fake news over the web (Silverman, 2015).

Differently, social spammers on social media refer to users who coordinately launch various types of attacks, such as spreading viruses or ads, and

phishing (Shu et al., 2017).

Related to social spamming, another widely studied aspect is that of fake reviews. Fake reviews are typically found on e-commerce websites and media review aggregators, and typically aim to improve or disrupt the popularity of a given product (Bondielli and Marcelloni, 2019).

While collecting the data, the relevance of the information is defined by context-based and content-based approaches (Shu et al., 2017). Content-based approaches rely on content features, which refer to information that can be directly extracted from text, such as linguistic features. The content has been analyzed previously on multiple levels. At the word level, methods have used features including word counts, derived measures like TF-IDF, small combinations of words like bigrams, and word embeddings (Benamira et al., 2019), (Castillo et al., 2011), (Huang et al., 2020), (Rubin et al., 2016). Context-based approaches are more varied, and generally rely on surrounding information, such as user's characteristics, social network propagation features and reactions of other users to the news or post. At the sentence level, methods rely on features like syntax and complexity (Benamira et al., 2019), (Castillo et al., 2011), (Huang et al., 2020), (Rubin et al., 2016).

Returning to the misinformation detection, most of the approaches proposed in the literature to detect false information face the task as a classification problem: they aim to associate labels such as rumour or non rumour, true or false with a particular piece of text. In most of the cases, researchers have employed machine learning (Breiman, 2001),(Chakraborty et al., 2016), and deep learning approaches, achieving promising results (Cheng et al., 2021),(Han et al., 2020). Alternatively, some researchers have applied other approaches based, for instance, on data mining techniques, such as time series analysis, and have exploited external resources (e.g. knowledge bases), to predict either the class of documents or events, or to asses their credibility (Kumar and Geethakumari, 2014), (Page et al., 1999), (Rubin and Lukoianova, 2015).

In this work, we will consider deep learning approaches, particularly, natural language processing models.

3 Methodology

3.1 Task Definition

Given a sequence of tokens $C = x_1, ..., x_n$, the label $y \in [1, 0]$ needs to be assigned to it, indicating whether it is fake or legit.

3.2 Datasets

In this project five datasets from different sources were used to train and test the models, which will be described in the section 3.3.
The first one is a part of **FakeNewsDataset** (Pérez-Rosas et al., 2018) - dataset collected for supervised learning. The authors did a lot of manual work to collect and verify the data. As a result, they managed to collect to 240 fake and 240 legit news on six different domains – sports, business, entertainment, politics, technology, and education. All the news are for the year 2018.
The second dataset is also a part of **FakeNewsDataset - CelebrityDataset** . It is dedicated to rumors, hoaxes, and fake reports about famous actors, singers, socialites, and politicians and has 250 fake and 250 legit articles almost like the previous one.
The next one is **ReCOVery** (Zhou et al., 2020). The main topic of 2020 was COVID-19 pandemic. Plenty of fakes and rumors were produced around this virus, which caused the creation of several datasets for automatic fake news detection. **ReCOVery** ia a multimodal dataset. It includes 140820 labeled tweets as well as 2029 news articles on coronavirus collected from reliable and unreliable resources. We used only articles in our project.
As you can see, all three datasets are not big. The first two have only about 500 samples. (Although the articles are quite long themselves, about $200 - 300$ words.) That is why it was decided to find bigger datasets to do experiments on them. The first one was the **FakeNewsNet** (Shu et al., 2018). It contains two comprehensive datasets that includes news content, social context, and dynamic information. All the news were collected with PolitiFact and GossipCop[1] crawlers. In general, 187014 fake and 415645 real news were crawled. In addition, this dataset was used in the reference article (Pelrine et al., 2021) and could be used for validation of the results.
The second big dataset we've chosen is **NELA-**

[1] https://www.gossipcop.com

GT-2018 (Nørregaard et al., 2019). In this dataset authors gathered a wide variety of news sources from varying levels of veracity and scraped article data from the gathered sources' RSS feeds twice a day for 10 months in 2018. As a result, a new dataset was created consisted of 713534 articles from 194 news and media producers.

3.3 Language models

Like in the reference article (Pelrine et al., 2021) we evaluate a variety of language models. Where available, we use the implementations and pre-trained weights provided by the Huggingface Transformers library (Wolf et al., 2019) and train using PyTorch library. Actually, all models were taken from there except $ElMo$. This model was implemented with the help of AllenNLP library (Gardner et al., 2018). In addition to the models evaluated in the original article we've tested another models, which was designed to have longer sequences of tokens as an input[2]: **Longformer** (Beltagy et al., 2020). All the models, which were implemented and tested are listed bellow[3].

1. **BERT** (Devlin et al., 2018) `(bert-base-uncased)`
   Bidirectional Encoder Representations from Transformers, or BERT for short, is a large pre-trained bidirectional transformer. BERT was pre-trained using two objectives. The first, masked language modelling, required BERT to predict a masked token from the input. The second, next sentence prediction, required predicting whether two sentences appeared consecutively in the training corpus. To complete the latter task, BERT prepends the input text with a special [CLS] token, and inserts a special [SEP] token between the first and second sentence as well as at the end of the second sentence. The [CLS] token is commonly used as a document-level representation for classification tasks.

2. **BERT-Tiny** (Turc et al., 2019) `(google/bert_uncased_L-2_H-128_A-2)`.
   BERT-Tiny is the smallest of several pre-trained language models that follow the same pre-training procedure as BERT, as well as a knowledge distillation fine-tuning procedure. These smaller models achieve strong performance on downstream tasks with significantly fewer parameters.

3. **RoBERTa** (Liu et al., 2019) `(roberta-base)`.
   RoBERTa follows a similar architecture to BERT but removes the next-sentence prediction pre-training objective while making the masking procedure dynamic by regenerating the mask for example every time and leverage larger batch sizes and increased training iterations to improve performance. They find that BERT underfits its training data.

4. **ALBERT** (Lan et al., 2019) `(albert-large-v2)`
   ALBERT adds an additional pre-training objective while incorporating two methods that reduce the number of parameters in the model, factorizing the embedding layer and tying weights across hidden layers.

5. **BERTweet** (Nguyen et al., 2020) `(vinai/bertweet-base)`
   BERTweet follows an identical training procedure to RoBERTa, and is pre-trained on 850 million tweets.

6. **COVID-Twitter-BERT** (Müller et al., 2020) `(covid-twitter-bert-v2)`
   COVID-Twitter-BERT follows an identical training procedure to BERT, and the latest version is pre-trained on 1.2 billion training examples generated from 97 million tweets.

7. **DeCLUTR** (Giorgi et al., 2020) `(declutr-base)` DeCLUTR is a transformer-based language model that proposes a contrastive, self-supervised method for learning general purpose sentence embeddings. The model is trained with a masked language modelling objective as well as with contrastive loss using both easy and hard negatives.

8. **Funnel Transformer** (Dai et al., 2020) `(funnel-transformer/small-base)`

[2]Longer than standard 512 tokens.

[3]Also, unlike the reference article, smaller version of several models were taken: roberta-base, funnel-transformer/small-base. That was done due two the limitations in computational power, provided by Google Colab.

The Funnel Transformer improves the efficiency of bidirectional transformer models by applying a pooling operation after each layer, akin to convolutional neural networks, to reduce the length of the input.

9. **ELMo** (Peters et al., 2018)
   ELMo is one of the first large-scale pre-trained language models. It is a character-based model and is the only model in this list that does not optimize for masked language modelling. Instead, it uses bidirectional LSTMs to perform autoregressive language modelling in both the forward and backward directions.

10. **Longformer** (Beltagy et al., 2020) `(allenai/longformer-base-4096)`.
    Transformer-based models are unable to process long sequences due to their self-attention operation, which scales quadratically with the sequence length. To address this limitation, the Longformer was introduced with an attention mechanism that scales linearly with sequence length, making it easy to process documents of thousands of tokens or longer. Longformer's attention mechanism is a drop-in replacement for the standard self-attention and combines a local windowed attention with a task motivated global attention.

3.4 Implementation details

In our implementation we use the [CLS] token embedding from the final layer of output for all models except ELMo. For this transformer we use mean pooling over output embeddings. As in the reference article we allow all the parameters of the language model to be fine-tuned and use a single fully-connected layer on the pooled embeddings for later classification. We train the models using cross-entropy loss and AdamW optimizer (Loshchilov and Hutter, 2017). As a learning rate scheduler we use a slanted triangular learning rate scheduler[4] (Howard and Ruder, 2018). All the models were trained on GPUs provided by Google Colab with batch size[5] of 32 and learning rate equal to $1e\text{-}5$. We train all the models with "500 symbols" method (4.1) for 10 epochs, except BERT-tiny. We train it for 50 epochs. And all the models with "many splits" method (4.4) for 5 epochs, because the datasets become significantly larger in this case. We ran each models 5 times and report mean and standard deviation.
All the implementation details can be found in our Colab notebook[6].

3.5 Metrics

For evaluation the results we used standard metrics: *f1 score*, *precision score* and *recall score*.

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

$$\text{precision} = \frac{tp}{tp + fp}$$

$$\text{recall} = \frac{tp}{tp + fn}$$

Where $tp,\ fp,\ fn -$ the number of true positive, false positive and false negative detection respectively.

4 Experiments

4.1 General models

Since all the computations were done on Google Colab, it's limitations in GPU memory forced us to accept some compromises to get the results. For each model and each dataset we performed preprocessing, taking only a part of the article (500 symbols). Let us call this approach "500 symbols" form now and so on. Then, we forwarded this cutted texts to tokenizer and to the model input after that. As you can see in the table 3 such a simple approach gave us great results on our main datasets.

4.2 Large datasets

In this subsection we will describe our approaches to use large dataset and test quality of classification.

First of all we tried to use FakeNewsNet dataset (Shu et al., 2018), but faced problems with code from official repository of dataset. We downloaded only small part of described data (about 400 samples). Anyway we decided to show the results on

[4] We use the implementation from https://gist.github.com/ceshine/ff32968bafc6fead87d7b6233ad8ab69 for all models except ELMo. There AllenNLP implementation was used.

[5] For COVID-Twitter-BERT and BERTweet batch size was reduced to 16 and for ALBERT and Longformer to 8 to fit in CUDA memory.

[6] https://colab.research.google.com/drive/1VsBolErB0QAdhBiAc176Qyt3Xp5oB5B5?usp=sharing

this dataset. Surprisingly, we got quite high results (table 1)[7].

Model name	F1	Recall	Prec
Bert	0.823	0.749	0.848
Bert-tiny	0.759	0.648	0.829
RoBERTa	0.717	0.827	0.663
Funnel Transformer	0.828	0.858	0.777
COVID-Twit-BERT	0.725	0.883	0.668
BERTweet	0.785	0.608	0.838
DeCLUTR	0.773	0.733	0.933
Albert	0.849	0.892	0.693

Table 1: Results on FakeNewsNet dataset

Another dataset we tried was NELA_GT_2018 (Nørregaard et al., 2019). There was tricky moment. The data was labeled in the following way: each journal had a score from multiple groups of journalists. And to get a real/fake labeling for article we had to figure out which journal published the article, decide if we could trust the journal based on scores of assessors. We decided to use NewsGuard scores. They had the following system: each journal got some point for avoiding deceptive headlines, some point for clearly labeling advertising and so on, 9 categories giving the sum of 100. To make Fake and Real sets approximately equal we decided to use 70 point to determine fake publisher. 70 and more points - article is real, below 70 - fake. Using this scheme we got $0, 1$ labels for each article. Due to Google Colab limitations we were able to test only small models. The results are provided in table 2.

Model name	F1	Recall	Prec
Bert-tiny	0.894	0.904	0.796
Funnel Transformer	0.917	0.823	0.948
BERTweet	0.940	0.817	0.901

Table 2: Results on NELA-GT-2018 dataset.

4.3 ELMo

This model has completely different from the other models implementation. The approach was written with the help of AllenNLP library. While conducting experiments with this model, it was discovered that creating the vocabulary straight from the training data does not give the good quality of classification, *f1 score* was about 50%. That is why it was decided to use a vocabulary from BERT (`bert-large-uncased`). It gave the excellent quality of 1.0 with zero standard deviation for all metrics on our 3 main datasets (first 3 in the list). Also, to verify this score, the cross-dataset test was done. The model was trained on Celebrity dataset and tested on ReCOVery dataset. The ideal result has remained.

4.4 Many splits approach

In the subsection 4.1 we used "500 symbols" method, where we took only 500 symbols of each sample text. However, our transformers overfitted on the 3rd or 4th epoch and. Also, friendly speaking it was not very natural to cut words. That is why, it was decided to test another approach. Let us call it "many splits". Instead of taking only 500 symbols, we can split each article on pieces and get several samples corresponding to one piece of news. In this case, we get the larger dataset, more stable to overfitting. To get the metric values during the test phase we used the majority vote strategy. If majority of fragments of the sample text were classified as legit, we treated this article as legit, and fake otherwise. As it was expected, this method gave the improvements in scores and the models started overfitting less. In the table 3 you can find the obtained results and comparison with "500 symbols" approach. It's important to mention that we haven't manage to make the experiments with "many splits" method for ReCOVery dataset for all models, only for BERT-Tiny, the smallest one. The reason was the Google Colab limitations. The experiments simply were not able to finish. But for BERT-Tiny we get the ideal score. And it was proved by the cross-dataset test[8]. Just look at the loss graph for this experiment.

[7] Std values in table 1 and 2 are not provided to make them more compact.

[8] The resullts for all cross-dataset tests for "many splits" methods are not provided in this report.

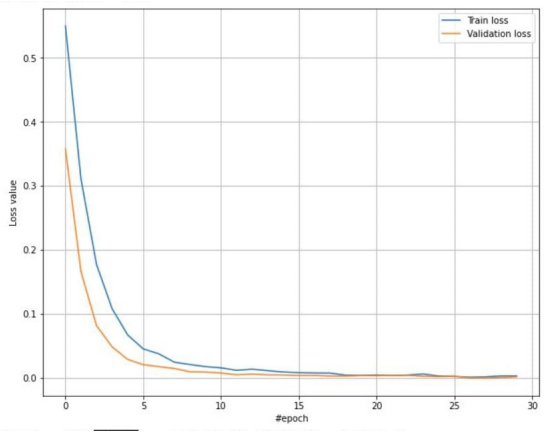


Figure 1: Loss curve for BERT-Tiny on ReCOVery with many splits method.

Also, we provide the comparison of loss curves corresponding to two approaches considered. In the figure 2 loss curves for the small BERT-Tiny model can be seen. On the figure 3 - for huge BERT model. These models were trained on the Celebrity dataset for 10 and 50 epochs respectively. The figures provide a good illustration, how the second approach improved the model quality over the first.

4.5 Cross-dataset tests

Cross-validation experiments were performed to test the quality of the trained model. While training on celebrity dataset data, testing was conducted on ReCOVery dataset data. The results are shown in the table 4. The results on the reCOVery test are comparable to those on Celebrity. Basically, the quality of F1-score ranges from -2% to $+2\%$. Thus, we see that for Bert the values on the test are worse than those on the train, and the situation is reversed for the Bert-tiny.

4.6 Longformers

Transformer-based models are unable to process long sequences due to their self-attention operation, which scales quadratically with the sequence length. To address this limitation, we use the Longformer with an attention mechanism that scales linearly with sequence length, making it easy to process documents of thousands of tokens or longer. We tested trained LongTransformer on all available Datasets and similarly tested cross testing on a ReCOVery dataset. For celebrity dataset we got 0.875 $f_1 score$, 0.844 precision and 0.942 recall, on cross testing the results are slightly worse 0.824 $f_1 score$, 0.842 precision and 0.831 recall.

4.7 Attention visualization

Almost all of the models in our list use attention mechanism. That is why, it was decided to get the visualization of attention layers in order to get some understanding, what words are important for model to make a decision. We've chosen BERT model for our tests as the most famous model. The figures we obtained showed that the most information is contained in [CLS] and [SEP] tokens. That is why taking the [CLS] token for classification is a good choice. This experiment illustrates the choice of the authors of the reference article (Pelrine et al., 2021). Unfortunately, no more interesting dependencies were found. You can see the visualizations in figures 4 and 5.

4.8 Comparison to baselines

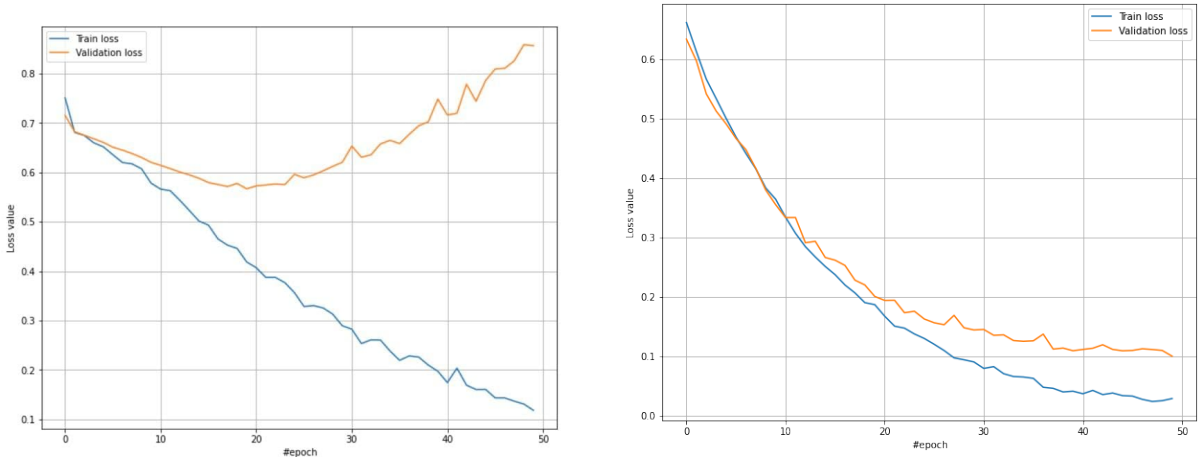
As far as we were inspired by related work, we have intersection in models we used in the project. Nevertheless we used different datasets and baseline comparison can't be truly fair. For models the results are pretty similar. For example, for Bert we have average F1 score around 0.8 and reference result is approximately 0.82. Similar situation with Bert-tiny, Albert, Roberta e.t.c. The only model that differs a lot is ELMo: we have F1 equal to 1.0, whereas in article the results are about 0.75.

5 Results and discussion

In this section we are to summarize all the experiments' results described above. First of all, all the methods from the original article were reproduced and tested on our 3 main datasets (Celebrity, FakeNews, Recovery). The models have shown great performance on them (table 3). It can be noted that with "500 symbols" approach RoBERTa has shown the best results, giving top $f_1 scores$ on all 3 datasets[9]. Also, it can be seen (table 3) that BERT-Tiny has the worst performance and also the smallest size. It is important to repeat here that all the models overfitted on early epochs.

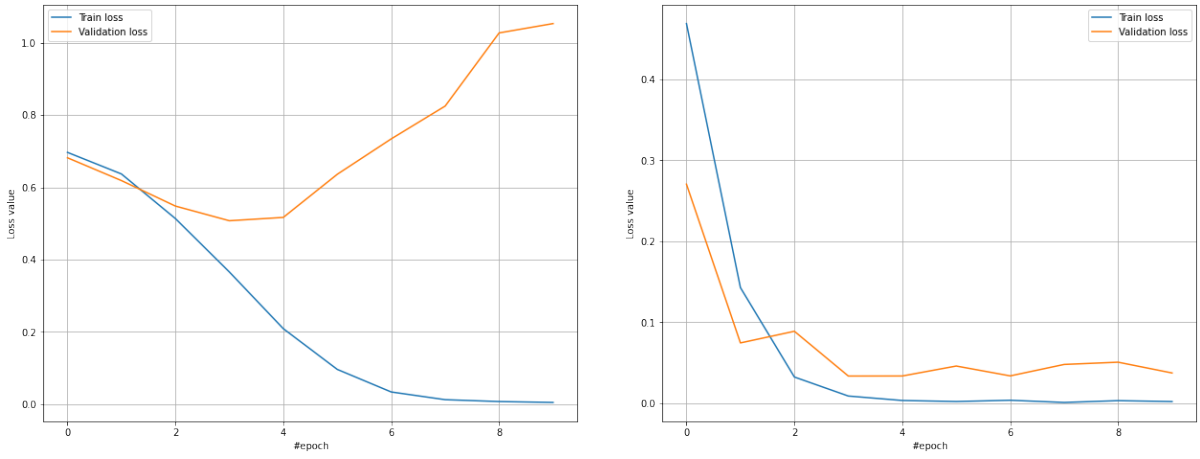
To estimate the reliability of the implemented methods we've done the cross-dataset tests and conducted experiments on bigger datasets. The cross-dataset tests gave us results comparable in quality to ones obtained on the test part of the dataset the model had been trained at. As for large datasets, although we didn't manage to get on of the big datasets from the reference article, we've taken

[9]The result on DeCLUTR is a bit better but the improvement can be included in std.



(a) (b)

Figure 2: Loss curves comparison for BERT-Tiny model. a) ”500 symbols” approach b) ”many splits” approach



(a) (b)

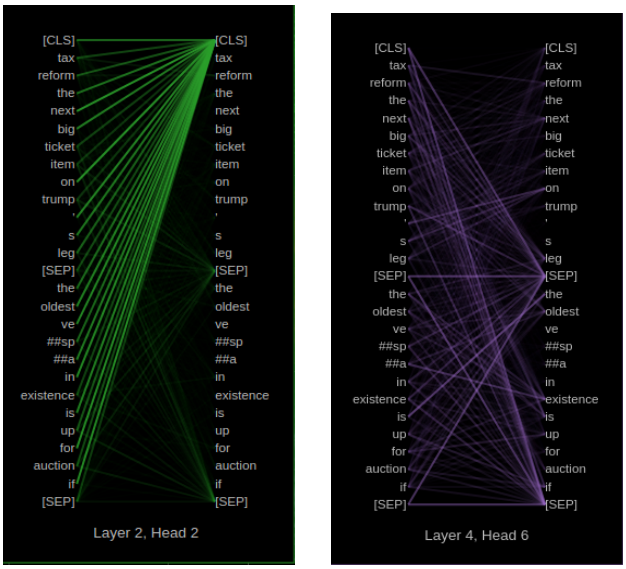
Figure 3: Loss curves comparison for BERT model. a) ”500 symbols” approach b) ”many splits” approach

Model name	Dataset	F1 score	Precision	Recall	F1 score	Precision	Recall
Approcach		500 symbols			Many splits		
BERT	FakeNews	0.674±0.011	0.670±0.089	0.730±0.139	0.810±0.029	0.728±0.040	0.915±0.029
	Celebrity	0.800±0.005	0.728±0.026	0.917±0.041	0.993±0.004	0.987±0.010	0.998±0.002
	Recovery	0.927±0.002	0.909±0.011	0.951±0.011	-	-	-
BERT-Tiny	FakeNews	0.643±0.026	0.587±0.049	0.737±0.062	**0.982±0.001**	**0.982±0.002**	**0.982±0.002**
	Celebrity	0.667±0.024	0.705±0.040	0.644±0.059	0.971±0.008	0.966±0.013	0.977±0.005
	Recovery	0.870±0.011	0.863±0.023	0.884±0.036	1 ±0.000	1 ±0.000	1 ±0.000
RoBERTa	FakeNews	0.953±0.011	0.938±0.024	0.973±0.006	0.886±0.009	0.871±0.042	0.907±0.050
	Celebrity	**0.856±0.006**	**0.784±0.014**	**0.952±0.022**	0.988±0.005	0.981±0.013	0.995±0.004
	Recovery	**0.975±0.003**	**0.960±0.008**	**0.992±0.003**	-	-	-
Funnel Transformer	FakeNews	0.842±0.016	0.762±0.025	0.958±0.019	0.821±0.003	0.748±0.008	0.910±0.010
	Celebrity	0.816±0.009	0.770±0.023	0.911±0.026	0.992±0.002	0.987±0.005	0.997±0.005
	Recovery	0.942±0.005	0.942±0.013	0.946±0.008	-	-	-
COVID-Twitter-BERT	FakeNews	0.800±0.015	0.738±0.040	0.893±0.052	0.872±0.006	0.820±0.007	0.930±0.006
	Celebrity	0.822±0.006	0.799±0.026	0.890±0.024	0.989±0.004	0.984±0.010	0.993±0.002
	Recovery	0.966±0.004	0.949±0.007	0.986±0.004	-	-	-
BERTweet	FakeNews	0.759±0.008	0.741±0.055	0.804±0.075	0.849±0.019	0.819±0.022	0.882±0.015
	Celebrity	0.796±0.026	0.759±0.022	0.861±0.039	**0.998±0.998**	**0.997±0.005**	**0.998±0.002**
	Recovery	0.949±0.003	0.943±0.016	0.964±0.007	-	-	-
DeCLUTR	FakeNews	**0.956±0.024**	**0.954±0.041**	**0.963±0.018**	0.891±0.004	0.873±0.017	0.910±0.015
	Celebrity	0.823±0.008	0.782±0.020	0.915±0.025	0.989±0.004	0.987±0.008	0.990±0.004
	Recovery	0.966±0.002	0.961±0.008	0.974±0.010	-	-	-
Albert	Celebrity	0.821±0.017	0.825±0.050	0.859±0.054	0.893±0.119	0.840±0.179	0.974±0.021
	FakeNews	0.870±0.016	0.887±0.022	0.883±0.046	0.772±0.056	0.673±0.074	0.910±0.020
	Recovery	0.905±0.006	0.898±0.008	0.928±0.007	-	-	-

Table 3: Comparison between two text preprocessing approaches.

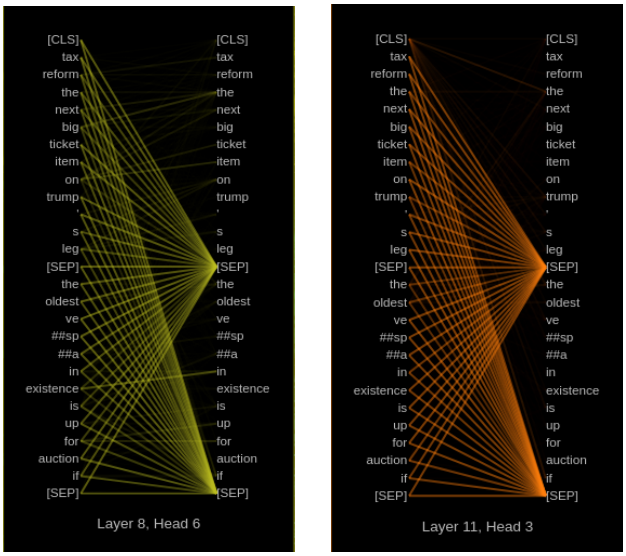
Approach	500 symbols			Many splits		
Model Name	F1	recall	precision	F1	recall	precision
Bert	0.757±0.007	0.891±0.003	0.667±0.013	0.756±0.003	0.784±0.002	0.737±0.007
Bert-tiny	0.766±0.006	0.921±0.010	0.620±0.004	0.768±0.008	0.842±0.005	0.707±0.012
RoBERTa	**0.83±0.009**	**0.967±0.016**	**0.782±0.011**	**0.831±0.005**	**0.967±0.002**	**0.728±0.016**
Funnel Transformer	0.763±0.006	0.867±0.09	0.697±0.004	0.779±0.004	0.808±0.009	0.751±0.020
COVID-Twitter-BERT	0.818±0.004	0.962±0.014	0.718±0.005	0.829±0.005	0.989±0.010	0.713±0.005
BERTweet	0.811±0.007	0.89±0.013	0.745±0.017	0.811±0.005	0.891±0.016	0.745±0.032
DeCLUTR	0.827±0.004	0.976±0.003	0.722±0.011	0.820±0.006	0.929±0.014	0.735±0.035
Albert	0.759±0.002	0.894±0.009	0.675±0.016	0.712±0.007	0.703±0.014	0.737±0.001
LongTransformer	0.823±0.004	0.831±0.007	0.842±0.005	-	-	-

Table 4: Results of the cross-dataset test and comparison between two text preprocessing approaches.



(a) (b)

Figure 4: Attention visualization of early layers.



(a) (b)

Figure 5: Attention visualization of late layers.

similar one and the results appeared to be very similar to ones in the article.

The problem of overfitting limited the scores of considered models. That is why we've tested the different approach for preprocessing which gave us the extension of the datasets and better learning curves. It can be concluded from table (table 3) that the results on Celebrity and Recovery datasets became significantly better. However, results on fakeNews remained the same or even a little worse. And in addition, the results on cross-dataset tests remained the same in general. Only small improvement can be seen (table 4). Hence, it can be concluded that "500 symbols" approach has only a little worse performance, but it is significantly faster in computations.

Last but not least, we've tested two more language models, which were not included in the comparison tables. Longformer has shown the similar results on our datasets. No improvement was obtained. On contrary, Elmo gave us excellent results of 1.0 on all datasets and metrics, and on cross-dataset test. And taking into account it's speed, Elmo became the best model of all considered.

6 Conclusion

During this project our team compared various pretrained models for fake news detection, compared quality of classification on different datasets of different sizes and conducted various experiments. Our research showed that pre-trained models can be used for such type of tasks even with relatively small datasets.

Many improvements can be made to this solution. For starters, one might try different preprepreprocessing solutions. Furthermore, test the considered models with larger batch sizes using more powerful hardware.

References

Hunt Allcott and Matthew Gentzkow. 2017. Social media and fake news in the 2016 election. *Journal of Economic Perspectives*, 31(2):211–36.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *CoRR*, abs/2004.05150.

Adrien Benamira, Benjamin Devillers, Etienne Lesot, Ayush K. Ray, Manal Saadi, and Fragkiskos D. Malliaros. 2019. Semi-supervised learning and graph neural networks for fake news detection. In *2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 568–569.

Alessandro Bondielli and Francesco Marcelloni. 2019. A survey on fake news and rumour detection techniques. *Information Sciences*, 497:38–55.

Leo Breiman. 2001. Random forests. *Mach. Learn.*, 45(1):5–32.

Guoyong Cai, H. Wu, and Rui Lv. 2014. Rumors detection in chinese via crowd responses. *2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014)*, pages 912–917.

Carlos Castillo, Marcelo Mendoza, and Barbara Poblete. 2011. Information credibility on twitter. pages 675–684.

Abhijnan Chakraborty, Bhargavi Paranjape, Sourya Kakarla, and Niloy Ganguly. 2016. Stop clickbait: Detecting and preventing clickbaits in online news media. *CoRR*, abs/1610.09786.

Y. Chen, Niall Conroy, and Victoria L. Rubin. 2015. Misleading online content: Recognizing clickbait as "false news". *Proceedings of the 2015 ACM on Workshop on Multimodal Deception Detection*.

Mingxi Cheng, Shahin Nazarian, and Paul Bogdan. 2021. Vroc: Variational autoencoder-aided multi-task rumor classifier based on text. *CoRR*, abs/2102.00816.

Niall J. Conroy, Victoria L. Rubin, and Yimin Chen.

Zihang Dai, Guokun Lai, Yiming Yang, and Quoc V. Le. 2020. Funnel-transformer: Filtering out sequential redundancy for efficient language processing. *CoRR*, abs/2006.03236.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Nicholas Difonzo and Prashant Bordia. 2007. Rumor, gossip and urban legends. *Diogenes*, 54:19–35.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew E. Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. Allennlp: A deep semantic natural language processing platform. *CoRR*, abs/1803.07640.

John M. Giorgi, Osvald Nitski, Gary D. Bader, and Bo Wang. 2020. Declutr: Deep contrastive learning for unsupervised textual representations. *CoRR*, abs/2006.03659.

Yi Han, Shanika Karunasekera, and Christopher Leckie. 2020. Graph neural networks with continual learning for fake news detection from social media. *CoRR*, abs/2007.03316.

Jeremy Howard and Sebastian Ruder. 2018. Fine-tuned language models for text classification. *CoRR*, abs/1801.06146.

Yen-Hao Huang, Ting-Wei Liu, Ssu-Rui Lee, Fernando Henrique Calderon Alvarado, and Yi-Shin Chen. 2020. Conquering cross-source failure for news credibility: Learning generalizable representations beyond content embedding. In *Proceedings of The Web Conference 2020*, WWW '20, page 774–784, New York, NY, USA. Association for Computing Machinery.

ROBERT H. KNAPP. 1944. A PSYCHOLOGY OF RUMOR. *Public Opinion Quarterly*, 8(1):22–37.

K. P. K. Kumar and G. Geethakumari. 2014. Detecting misinformation in online social networks using cognitive psychology. *Human-centric Computing and Information Sciences*, 4:1–22.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. ALBERT: A lite BERT for self-supervised learning of language representations. *CoRR*, abs/1909.11942.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Ilya Loshchilov and Frank Hutter. 2017. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101.

Martin Müller, Marcel Salathé, and Per Egil Kummervold. 2020. Covid-twitter-bert: A natural language processing model to analyse COVID-19 content on twitter. *CoRR*, abs/2005.07503.

Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. 2020. Bertweet: A pre-trained language model for english tweets. *CoRR*, abs/2005.10200.

Jeppe Nørregaard, Benjamin D Horne, and Sibel Adalı. 2019. Nela-gt-2018: A large multi-labelled news dataset for the study of misinformation in news articles. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 13, pages 630–638.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab. Previous number = SIDL-WP-1999-0120.

Kellin Pelrine, Jacob Danovitch, and Reihaneh Rabbany. 2021. The surprising performance of simple baselines for misinformation detection. *CoRR*, abs/2104.06952.

Verónica Pérez-Rosas, Bennett Kleinberg, Alexandra Lefevre, and Rada Mihalcea. 2018. Automatic detection of fake news. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3391–3401, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *CoRR*, abs/1802.05365.

Victoria Rubin, Niall Conroy, Yimin Chen, and Sarah Cornwell. 2016. Fake news or truth? using satirical cues to detect potentially misleading news. In *Proceedings of the Second Workshop on Computational Approaches to Deception Detection*, pages 7–17, San Diego, California. Association for Computational Linguistics.

Victoria L. Rubin and Tatiana Lukoianova. 2015. Truth and deception at the rhetorical structure level. *Journal of the Association for Information Science and Technology*, 66.

Kai Shu, Deepak Mahudeswaran, Suhang Wang, Dongwon Lee, and Huan Liu. 2018. Fakenewsnet: A data repository with news content, social context and spatialtemporal information for studying fake news on social media. *arXiv preprint ArXiv:1809.01286*.

Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. 2017. Fake news detection on social media: A data mining perspective. *CoRR*, abs/1708.01967.

C. Silverman. 2015. Lies, damn lies, and viral content. how news websites spread (and debunk) online rumors, unverified claims, and misinformation. *Tow Center for Digital Journalism 168*.

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Well-read students learn better: The impact of student initialization on knowledge distillation. *CoRR*, abs/1908.08962.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.

Xinyi Zhou, Apurva Mulay, Emilio Ferrara, and Reza Zafarani. 2020. Recovery: A multimodal repository for covid-19 news credibility research. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 3205–3212.

Arkaitz Zubiaga, Ahmet Aker, Kalina Bontcheva, Maria Liakata, and Rob Procter. 2017. Detection and resolution of rumours in social media: A survey. *CoRR*, abs/1704.00656.

Arkaitz Zubiaga, Maria Liakata, Rob Procter, Kalina Bontcheva, and Peter Tolmie. 2015. Towards detecting rumours in social media. *CoRR*, abs/1504.04712.