Nkovachevich at CheckThat! 2021: BERT fine-tuning approach to fake news detection

Ninko Kovachevich[1]

[1]Sofia University, "St. Kliment Ohridski", bul. "Tsar Osvoboditel" 15, 1504 Sofia Center, Sofia, Bulgaria

Abstract

The success of a text classification approach depends to a large extent on the data that it is trained on. The adaptation of a model with thousands of weights, such as BERT, usually requires large amount of data. CLEF 2021 CheckThat! Lab for fake news detection offers a challenging multi-class task with relatively small data set to train on. The experiments, which include BERT fine-tuning, together with a couple of simpler algorithms, produce average results, and hardly overcome model and data size limitations. Nevertheless, they show a potential to be successful if implemented properly.

Keywords

CLEF CheckThat, fake news, classification, BERT

1. Introduction

This year‘s edition of CLEF CheckThat! Lab [1][2][3] consists of three tasks, which if connected, could represent a comprehensive approach to the verification of a news article.

1. Check-Worthiness Estimation
2. Detecting Previously Fact-Checked Claims
3. Consists of two sub-tasks[4][5]:
    - Sub-task A - multi-class fake news detection of news articles
    - Sub-task B - topical domain classification of news articles

This paper describes developments in the third of the above tasks. The provided data consists of the headline and text of the news, with separate sets for each of the sub-tasks. The classes in which a news item may fall are the following:

- Sub-task A - true, false, partially false, other
- Sub-task B - health, crime, economy, elections, climate, education

2. Related work

Early methods for recognizing fake news are based on statistical and linguistic text characteristics. Gravanis et al. [6] make a comparison of such early methods, with emphasis on the

CLEF 2021 – Conference and Labs of the Evaluation Forum, September 21–24, 2021, Bucharest, Romania

n.kovachevich@mail.com (N. Kovachevich)
0000-0002-4580-3806 (N. Kovachevich)

features that are extracted from text. Some of the feature categories are language complexity, expressiveness, affect, bias, etc. These approaches rely on the idea that the writing style differs in fake news. The authors compare various classification algorithms on different fake news data sets in a bid to provide a benchmark for the task. The reported accuracy results show significant dependence on data set size. The best performing algorithms, SVM and AdaBoost, classify over 90% of the news articles successfully if there are at least few thousand samples, while this percentage is between 60 and 70 if the text documents are just few hundreds.

In modern methods the content of the text is used. Jwa et al. [7] propose a BERT model, trained on CNN and Daily Mail news articles, and subsequently applied to the FNC-1 challenge[8]. Although the FNC-1 challenge is not a straight classification and is actually a different task that involves title to body stance detection, the described approach and the data set itself can be leveraged in other fake news detection problems.

Nasir et al. [9] apply a combination of convolutional and recurrent neural networks whose inputs are popular word embedding such as GloVe[10]. Their best performing hybrid model consists of six layers, which include embedding, 1-D convolution, 1-D max pooling and LSTM. They validate their model on two differently sized data sets, reporting F1-score of nearly 99% for the big one and 60% for the small one. One thing to note is that these data sets represent a binary classification task.

3. Training data and working environment

3.1. Sub-task A data

900 news items are provided, each consisting of a headline and text content, distributed in four classes of misinformation[11]. About 52 percent of the items are labelled as FALSE, i.e. real news, and about 16 percent are labelled as TRUE. The rest are 24 percent and 8 percent, respectively for PARTIALLY FALSE and OTHER.

The articles are of different length, with most of them between 100 and 1000 words. Figure 1 shows the relationship between the articles and their length.

3.2. Sub-task B data

318 articles that are a subset of those in sub-task A, additionally labelled by topic domain. The largest group is health articles, about 40 percent, and the smallest is education - 8 percent.

3.3. Working environment

The developments are done using Google Colaboratory Python IDE that provides about 15GB of GPU, which is important for BERT models application.

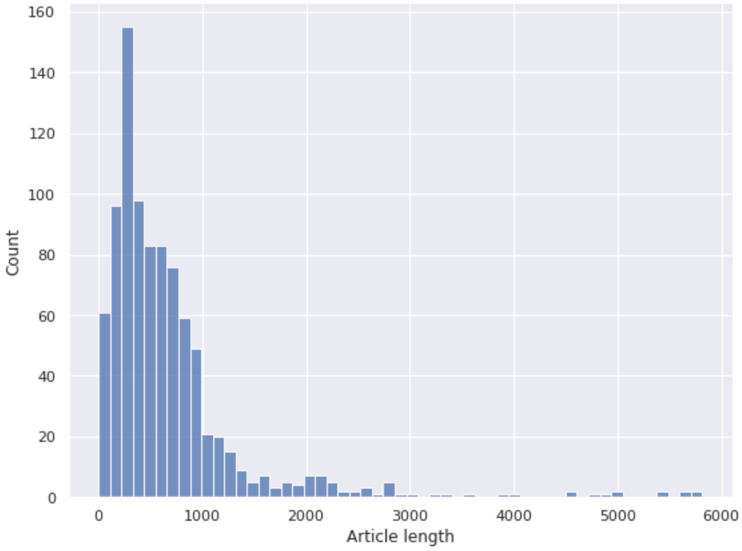


Figure 1: Number of articles according to their length, measured in words.

4. Implementation details

4.1. Articles title use

The best option is to determine the stance between the text of the news article and its title and then use this as a starting point or a parameter for the classification. This is not done because a separate classifier must be trained on a specifically labelled set of data, such as FNC-1[8]. Therefore, this issue is addressed separately for each of the approaches, usually experimenting with both options - text contents only, or concatenation of article and text.

4.2. Including additional training data

Considering that the available training data set is small, additional sources of similar fake news challenges can be included. Unfortunately, with these sources the prediction task is different, usually a binary classification. Given that the evaluation metric is F1-macro, which does not take label imbalance into account, changing the original class distribution during training and thus suppressing the minor class, seriously degrades the evaluation score.

4.3. Training data split

Although the data set is small, the separation is static without cross-validation. 10 percent is set aside for testing.

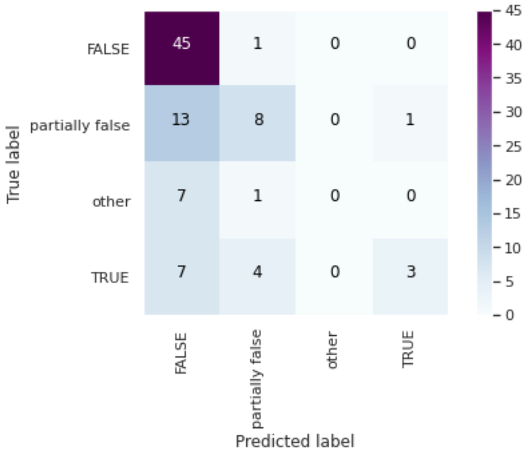


Figure 2: Confusion matrix for the results of applying Multinomial Naïve Bayes classification on TF-IDF presentation of articles, with a limit of 500 most common stems that occur in the text. The results are for sub-task A over a test set of 10% of all articles.

5. Developments

5.1. Baseline - Naive Bayes

The text is cleared of everything possible (stop words, numbers, punctuation, etc.) and stemming is applied to the words. In sub-task A, the title is used, while in sub-task B it is not.

Multinomial Naïve Bayes classification is applied. Before that, the documents are converted to TF-IDF vectors and only the most frequent stems are used for vocabulary. For sub-task A, limiting to the most common 500 stems surprisingly, given the simplicity of the method, leads to some meaningful results. For sub-task B, the optimal value is 200. Only single stems, i.e. uni-grams, are considered.

The results for sub-task A are shown in Figure 2. The predictions are not really diverse and the score is quite poor - F1-macro is 0.39. Nevertheless, the news articles declared by the model to be fake are in fact such. In addition, some meaning can be sought in the wrong cases. For example, the four articles classified as partially false, that are truly fake. The opposite case is an article that is classified as fake, but in fact is partially untrue. All this comes as a prove that there is some underlying logic in the text, which if extracted, can be used to judge the authenticity of a news article.

For sub-task B, the results are much closer to the goal and the value for the F1-macro metric is 0.81, which is good enough, considering how simple the model is.

5.2. Word embedding with GloVe

The idea is to get a vector with some dimensionality for each word in the text. The word vectors can then be:

- Averaged and passed as parameters to a classification algorithm.

- Used in raw form by specifying a fixed number of words for all text articles. Thus, the long articles are cut and the short ones are complemented with padding. In this form, the vectors are fed to a neural network in which the first layer is an Embedding layer.
  An obvious disadvantage is that information is lost in texts longer than the selected number of words. On the other hand, a higher number means more weights need to be learned by the neural network.

GloVe[10] provides pre-trained models that differ in output vector size and training source. The one used in the experiments, glove-wiki-gigaword-100, provides word embedding with size 100 and is trained on Wikipedia and Gigaword text corpus. The experiments produce the following results:

- When averaging the vectors, XGBoost[12] with tree-based booster and 200 estimators gives 0.47 and 0.80 for the two sub-tasks, respectively.
- By training a neural network for 8 epochs with the vectors of the first 1000 words of each article, F1-macro for sub-task A is 0.51. The network architecture has the Embedding layer and two LSTM layers, consisting of 64 and 32 neurons. For optimizer is used Adam with learning rate of 0.01 and categorical cross-entropy as a loss function.
  For sub-task B none of the tested network architectures returns a good result.

5.3. BERT fine-tuning

The pre-trained BERT models combine word embedding and a multilayer neural network with weights trained on a huge corpus of texts. The fine-tuning of such a model basically means several epochs of additional training on texts for the specific classification task. The authors of BERT recommend this to be done in two, three, or four epochs[13].

The Hugging Face implementation[14] is used and in particular experiments were performed with:

- BertForSequenceClassification - implementation of the original BERT[13] model with a linear classifier on top of it.
- RobertaForSequenceClassification - implementation of RoBERTa[15] with a linear classifier on top of it.

A major limitation of this approach is that these models can ingest up to 512 tokens as input. In addition, they are so large that the resources in Google Colaboratory are not enough to process more than about 400 for batch size of 10 text documents. However, in case of small number of training documents, even this number of input tokens could be too high to be able to achieve decent classification results.

One decision that can be made is where to cut those 400 tokens from - beginning of text, the middle part, or the end. Experiments show that cutting from the beginning produces better results than the other two options.

Applying BertForSequenceClassification to sub-task A produces F1-macro value of 0.46 with 300 tokens as input and training for 4 epochs. It is somewhat surprising that there are almost no mistakes in news longer than 1000 tokens. Prediction results by text length are shown in Figure 3.

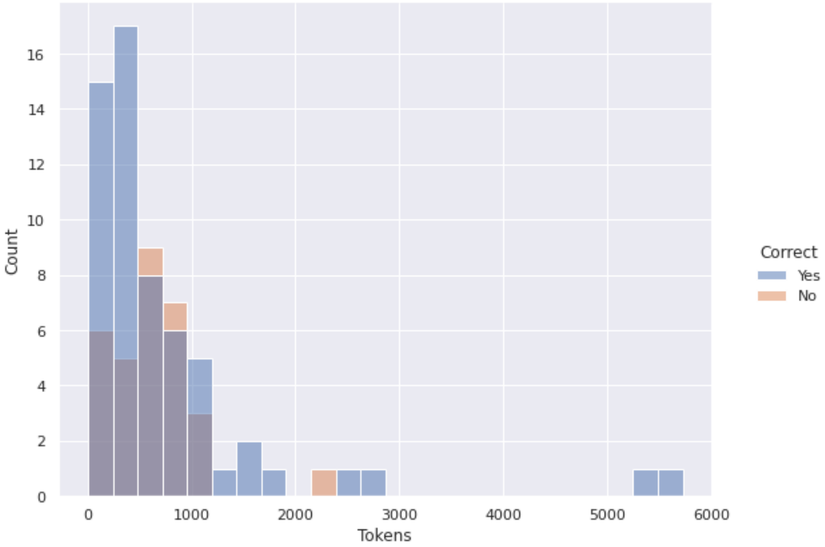


Figure 3: Text length of the predicted news articles by BERT model for sub-task A. The largest number of erroneous predictions were made in articles with length of 500 to 1000 tokens. Although only the first 300 tokens of each news item are submitted to the model, there are just few errors in texts longer than 1000 tokens.

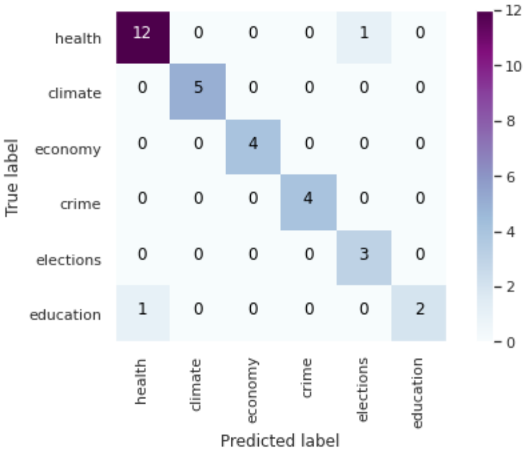
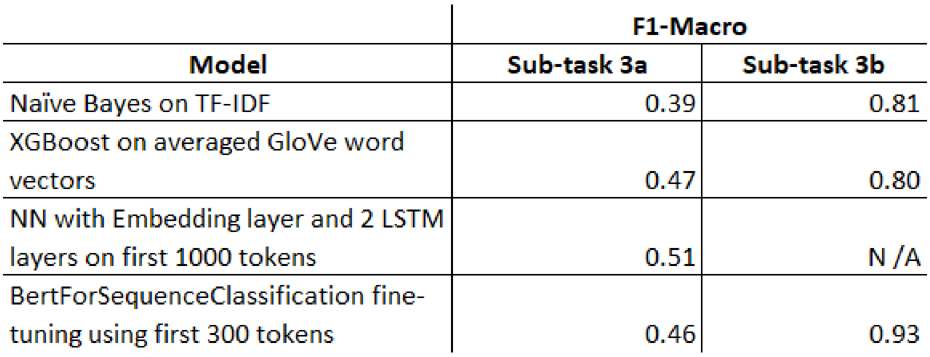


Figure 4: Confusion matrix for the prediction results by BERT for sub-task B over testing set of 10% of all articles. The classification domain is quite common and the model does well.

With the difference that it requires more epochs for training, RobertaForSequenceClassification does not show significant improvement.

For sub-task B, despite the limited amount of sample news articles, the pre-training of the BERT models provides good enough background for classification, and the results are satisfactory. Again, only the first 300 tokens are used, with 3 learning epochs. The F1-macro score is 0.93. Results' visualization is shown in Figure 4.

Table 1
F1-macro score summary of the results obtained during training.



	F1-Macro	
Model	**Sub-task 3a**	**Sub-task 3b**
Naïve Bayes on TF-IDF	0.39	0.81
XGBoost on averaged GloVe word vectors	0.47	0.80
NN with Embedding layer and 2 LSTM layers on first 1000 tokens	0.51	N /A
BertForSequenceClassification fine-tuning using first 300 tokens	0.46	0.93

6. Results and conclusion

Table 1 shows a summary of all the described approaches to the task. Although for sub-task A the expectations for decent results using BERT were not met, this model generalized better. Applying it to the labelled test set after the competition ended, the results were close to those obtained during training, while the other methods‘s score was down by a margin.

The main direction of further development is the inclusion of article titles as an input. For the purpose, a separate classifier can be trained on FNC-1 data.

Another development direction is to divide longer texts into paragraphs or sentences thus supplying the BERT model with text in pieces. On the one hand all the information from the text will be used, but on the other hand valuable context can be lost.

References

[1] CLEF, Clef2021–checkthat! lab, 2021. Https://sites.google.com/view/clef2021-checkthat.

[2] P. Nakov, G. Da San Martino, T. Elsayed, A. Barrón-Cedeño, R. Míguez, S. Shaar, F. Alam, F. Haouari, M. Hasanain, N. Babulkov, A. Nikolov, G. K. Shahi, J. M. Struß, T. Mandl, The CLEF-2021 CheckThat! lab on detecting check-worthy claims, previously fact-checked claims, and fake news, in: Proceedings of the 43rd European Conference on Information Retrieval, ECIR '21, Lucca, Italy, 2021, pp. 639–649. URL: https://link.springer.com/chapter/10.1007/978-3-030-72240-1_75.

[3] P. Nakov, G. Da San Martino, T. Elsayed, A. Barrón-Cedeño, R. Míguez, S. Shaar, F. Alam, F. Haouari, M. Hasanain, N. Babulkov, A. Nikolov, G. K. Shahi, J. M. Struß, T. Mandl, S. Modha, M. Kutlu, Y. S. Kartal, "overview of the CLEF-2021 CheckThat! lab on detecting check-worthy claims, previously fact-checked claims, and fake news", in: "Proceedings of the 12th International Conference of the CLEF Association: Information Access Evaluation Meets Multiliguality, Multimodality, and Visualization", CLEF '2021, Bucharest, Romania (online), 2021.

[4] G. K. Shahi, J. M. Struß, T. Mandl, Overview of the CLEF-2021 CheckThat! lab task 3

on fake news detection, in: Working Notes of CLEF 2021—Conference and Labs of the Evaluation Forum, CLEF '2021, Bucharest, Romania (online), 2021.

[5] G. K. Shahi, J. M. Struß, T. Mandl, Task 3: Fake news detection at CLEF-2021 CheckThat!, 2021. URL: https://doi.org/10.5281/zenodo.4714517. doi:10.5281/zenodo.4714517.

[6] G. Gravanis, A. Vakali, K. Diamantaras, P. Karadais, Behind the cues: A benchmarking study for fake news detection, Expert Systems with Applications 128 (2019). doi:10.1016/j.eswa.2019.03.036.

[7] H. Jwa, D. Oh, K. Park, J. Kang, H. Lim, exbake: Automatic fake news detection model based on bidirectional encoder representations from transformers (bert), Applied Sciences 9 (2019) 4062. doi:10.3390/app9194062.

[8] FNC-1, Fake news challenge, 2017. Http://www.fakenewschallenge.org/.

[9] J. Nasir, O. Khan, I. Varlamis, Fake news detection: A hybrid cnn-rnn based deep learning approach, International Journal of Information Management Data Insights 1 (2021) 100007. doi:10.1016/j.jjimei.2020.100007.

[10] J. Pennington, R. Socher, C. D. Manning, Glove: Global vectors for word representation, in: Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1532–1543. URL: http://www.aclweb.org/anthology/D14-1162.

[11] G. K. Shahi, A. Dirkson, T. A. Majchrzak, An exploratory study of covid-19 misinformation on twitter, Online Social Networks and Media 22 (2021) 100104.

[12] T. Chen, C. Guestrin, XGBoost: A scalable tree boosting system, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16, ACM, New York, NY, USA, 2016, pp. 785–794. URL: http://doi.acm.org/10.1145/2939672.2939785. doi:10.1145/2939672.2939785.

[13] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186. URL: https://www.aclweb.org/anthology/N19-1423. doi:10.18653/v1/N19-1423.

[14] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Brew, HuggingFace's Transformers: State-of-the-art Natural Language Processing (2019). URL: http://arxiv.org/abs/1910.03771. arXiv:1910.03771.

[15] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized BERT pretraining approach, CoRR abs/1907.11692 (2019). URL: http://arxiv.org/abs/1907.11692.