

NLPIR-UNED at CheckThat! 2022: Ensemble of Classifiers for Fake News Detection

Juan R. Martinez-Rico¹, Juan Martinez-Romo^{1,2} and Lourdes Araujo^{1,2}

¹NLP & IR Group, Dpto. Lenguajes y Sistemas Informáticos, Universidad Nacional de Educación a Distancia (UNED), Madrid 28040, Spain

²Instituto Mixto de Investigación - Escuela Nacional de Sanidad (IMIENS)

Abstract

This article describes the different approaches used by the NLPIR-UNED team¹ in the CLEF2022 Check-That! Lab to tackle the Task 3 - English. The goal of this task is to determine the veracity of the main claim made in a news article. It is a multi-class classification problem with four possible values: *true*, *partially false*, *false*, and *other*. For this task, we have evaluated three different approaches. The first has been based on a Longformer transformer model that supports larger input sequence sizes than other transformer models such as BERT. The second approach uses transformer models where an extension of the training set has been carried out. The last approach uses an ensemble classifier composed of a transformer model fed with the sequence of words of the article to be evaluated, and a feed forward neural network fed with features related, among other things, to the number of named entities in the article, and features extracted using the LIWC text analysis tool. With this last approach, we have made our main submission reaching the second position among the twenty-five participating teams.

Keywords

Fake News Detection, Transformer Models, Ensemble of Classifiers

1. Introduction

As events unfold: elections, pandemic, war, etc., the proliferation of fake news that manipulates public opinion becomes more evident. This kind of disinformation does not need to be too sophisticated as it is usually targeted at a biased audience that is receptive to these messages.

Therefore, tools that allow this type of news to be automatically intercepted before they can produce undesirable consequences are increasingly necessary. It can also be of great help in achieving this goal to have forums in which different approaches to this task can be evaluated and shared. One of the events in which this type of activity takes place is the CheckThat! Lab organized at the CLEF conference [1, 2], and specifically in Task 3 [3], which is dedicated to the detection of fake news. In this task, the organizers, who have previously worked on the study of fake-news and the development of datasets related to this problem [4, 5?], provide a set of around 1,300 articles, for which it is necessary to determine if the main claim they contain can

¹Identified as *nlpiruned* in the official results.


CLEF 2022: Conference and Labs of the Evaluation Forum, September 5–8, 2022, Bologna, Italy

✉ jrmartinezrico@invi.uned.es (J. R. Martinez-Rico); juaner@lsi.uned.es (J. Martinez-Romo); lurdes@lsi.uned.es (L. Araujo)

🆔 0000-0003-1867-9739 (J. R. Martinez-Rico); 0000-0002-6905-7051 (J. Martinez-Romo); 0000-0002-7657-4794 (L. Araujo)



© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

be classified as *false*, *partially false*, *true* or *other*. This dataset has been compiled from various fact-checking sites, retrieving the original articles and unifying the labeling.

We have organized the rest of the article as follows: in Section 2 we make a brief review of the different approaches carried out in recent years to the task of detecting fake news and disinformation, in Section 3 we explain our different approaches to this task, Section 4 discuss the results obtained, and Section 5 contains our conclusions and future work.

2. Related Work

We can approach the fake news detection task in many ways but they can generally be grouped into three main strategies: use the information around the article (metadata, author information, publisher information, spatial and temporal dispersion, etc.), use only the information present in the article (textual content, images, writing style, etc.), or try to extract the claims made in the article to verify them in external knowledge bases, understanding that the presence of one or more false claims would also make the article false or at least partially false.

Perhaps the last of them would be the desirable approach since each prediction given by a classifier based on this strategy can provide a direct explanation for said prediction, but the great drawback is the lack of completeness of the available knowledge bases. This means that most of the claims that can be found in an article cannot be verified.

Given the format of Task 3, we will focus on reviewing the approaches made recently based solely on the content of the article. Following this strategy Pérez-Rosas et al. [6], in addition to systematizing the process of creating a corpus that can be used to train fake news detection systems, they propose the use of features such as n-grams encoded as TF-IDF vectors, features extracted using the Linguistic Inquiry and Word Count (LIWC) [7] text analysis tool, syntactic features created from grammar production rules that are also encoded as TF-IDF values, and features associated with readability such as the number of complex or long words, number of paragraphs, etc., using a linear SVM as classifier. Other types of features used in the detection of misleading content compare the coherence and structure of the discourse between misleading and true narratives [8]. Sentiment analysis scoring of news content or social media posts [9] is another method that can give clues about misleading content and suspicious authors such as bots, that are frequently linked to the proliferation of fake news.

With the advent of transformer models [10], which have become the default choice in many of the tasks related to natural language processing, numerous proposals have emerged to apply these models to the detection of fake news. These models are pre-trained with a large amount of textual information with learning objectives, such as predicting the next sentence or predicting a masked word in the sentence, that do not require manual annotation of these large corpora. Schütz et al. [11] evaluate different transformer models: BERT [12], ALBERT [13], RoBERTa [14], DistilBERT [15] and XLNet [16] on the FakeNewsNet dataset [17], using the text of the article, the title and the concatenation of both as input. In their conclusions, they confirm that performing a fine adjustment of these models with a dataset oriented to the detection of fake news allows obtaining promising results, with an accuracy greater than 80% in almost all configurations. Gundapu and Mamidi [18] after experimenting with different classifier types: Linear Regression (LR), Support Vector Machines (SVM), Multi-Layer Perceptron (MLP), Long

Short-Term Memory (LSTM), Convolution Neural Network (CNN), etc., finally found that an ensemble of three transformer models BERT, ALBERT and XLNet, fed with the sequence of words of the news item to be analyzed outperforms all the alternatives studied. Instead, Mehta et al. [19] uses three BERT models with shared weights whose outputs are concatenated in order to be able to feed the first one with the sequence of text tokens, the second with the metadata associated with this text, and the third model is fed with the justification given for its truth value. This architecture also shows superior behavior than other types of classifiers like LR, SVM, CNN, or Bi-LSTM.

3. Approaches to Fake News Detection Task

To tackle Task 3 in this 2022 edition of the CheckThat! Lab, our team has evaluated three strategies. The first of them is, given that this task is posed with articles with full text (the maximum number of tokens detected is 5816), to use a type of transformer model that allows sequence sizes greater than those admitted for example by BERT, whose maximum sequence size in its standard implementation is 512.

The second strategy can be considered the opposite of the first: using a small sequence size such as 128 tokens, and assuming that a true or false instance of the *training* dataset still holds the same truth value if we chop it into 128-token chunks, we have expanded the *training* dataset and evaluated different transformer models on this extended dataset.

Our last strategy has tried to take advantage of the transformer models for their ability to extract latent features present in the text, with the use of explicit features such as those provided by the LIWC text analysis tool, which have also given good results in detecting misleading content. For this, we have developed an ensemble classifier that can be fed with both types of inputs simultaneously.

We detail each of these approaches below. All used pre-trained transformer models have been downloaded from <https://huggingface.co/>.

3.1. Using Transformers With Larger Sequence Sizes

To evaluate this option we have made use of a Longformer [20] pre-trained model. This architecture is a variation of the standard transformer models with the particularity that the self-attention operation scales linearly with the sequence length, unlike others where this operation scales quadratically and limits its performance with long sequences.

The implementation of a sequence classifier with this model does not differ from that carried out with any other transformer model. We have used the pre-trained model *allenai/longformer-base-4096* that supports sequences of length up to 4096.

To carry out the training process, we have unified the provided files *task3_english_training* and *task_3a_sample_data* and then we have left 33% of the instances as a *test* dataset. Of the remaining instances, we have reserved 20% as a *dev* dataset. No preprocessing has been done to the input text.

This process has been carried out with maximum sequence sizes of 512 and 1024, and a batch size of 8. To determine the configuration with the best performance, the system was

configured to use deterministic algorithms and the tests were repeated for 10 different random seeds, obtaining the average of the precision, recall, and F1 measurements.

An early stopping mechanism has also been implemented. This mechanism stores the updated state of parameters after each epoch and stops training when there have been no improvements in the measure F1 over the *dev* dataset in the last n epochs (default value 2), then selecting the saved configuration with the best F1 measure during that interval.

3.2. Training Dataset Expansion

Our second option to address this task has been the use of various transformer models but with an extended *training* dataset. To do this expansion, we have partitioned the data in a similar way as described in the previous section, except to extract the *dev* dataset from *training* dataset since expanding the latter has allowed us to experiment with smaller size partitions on the *dev* dataset: 20% and 10%.

We have configured a maximum sequence size of 128 tokens and during the training process the instances of the *training* dataset are dynamically partitioned into 128-word chunks. The same early stopping mechanism and random management configuration described above has also been used.

Transformer models *funnel-transformer/intermediate*, *bert-base-cased*, *bert-base-uncased*, and *albert-base-v2* have been evaluated by repeating the training process for 10 pre-selected random seeds and obtaining the average of the precision, recall and F1 measurements. In this case the batch size has been 16.

3.3. Ensemble Classifier

In our last approach we wanted to check if using a transformer classifier with its typical input, i.e. a sequence of words, together with a feed forward neural network (FFNN) classifier with discrete features, gave better performance than using only a transformer model.

For this, we have built an ensemble classifier that is internally composed of a transformer classifier and an FFNN classifier. The input of the ensemble classifier is therefore composed of a section in which the sequence of words of the article is introduced, and a section we feed with a vector of discrete features extracted from that article.

The hidden layer of the FFNN and the first token of the last hidden layer of the transformer (classification token) are concatenated and form the first layer of the ensemble classifier. Behind this concatenation layer are two hidden layers and one output layer (Figure 1). It is also possible to disable one of the hidden layers by configuration.

Before training the ensemble classifier, the transformer and FFNN models are trained separately on the same dataset and stored in binary files. These models are then loaded in evaluation mode in the ensemble classifier to prevent their parameters from being modified during ensemble training.

On the other hand, we have selected two different sets as discrete features: the 93 features generated by the LIWC text analysis tool, and a series of features elaborated from the syntactic analysis, dependencies, named entities, and sentiment score information which generates the

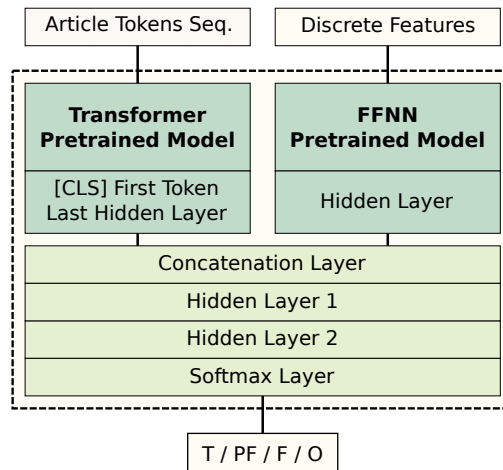


Figure 1: Transformer-FFNN ensemble.

Table 1

Features generated from Stanza tool data

Name	Description
no_sentences	Number of sentences in the article
no_subj_sentence	Ratio between number of subjects detected and number of sentences
no_pred_sentence	Ratio between number of predicates detected and number of sentences
no_obj_sentence	Ratio between number of objects detected and number of sentences
no_ne_sentence	Named entities per sentence
no_ne_person_sentence	Number of person-type named entities per sentence
no_ne_product_sentence	Number of product-type named entities per sentence
no_ne_org_sentence	Number of org-type named entities per sentence
no_ne_norp_sentence	Number of norp-type named entities per sentence
no_ne_fac_sentence	Number of fac-type named entities per sentence
no_ne_gpe_sentence	Number of gpe-type named entities per sentence
no_ne_loc_sentence	Number of loc-type named entities per sentence
no_ne_work_of_art_sentence	Number of work-of-art-type named entities per sentence
sentiment	Sentiment score

Stanza NLP tool [21] from the article text. The complete list of features that make up this second group can be seen in Table 1.

To select the appropriate configuration of the FFNN classifier, a grid search has been carried out by varying the following hyper-parameters:

- FFNN activation function: *relu*, *sigmoid*, *tanh*.
- Hidden layer size: 100, 500, 1000.

After evaluating the results, the *sigmoid* activation function has been selected and a hidden layer size of 500 has been configured. These parameters have been set in the ensemble classifier.

Finally, a second grid search has been carried out by altering the following hyper-parameters of the ensemble classifier:

- Pretrained model: *funnel-transformer/intermediate*, *bert-base-cased*, *bert-base-uncased*, *albert-base-v2*.
- Ensemble activation function: *relu*, *sigmoid*, *tanh*.
- Number of hidden layers: 1 or 2.
- Dropout: 0, 0.2, 0.5.
- Discrete features: LIWC, Stanza, All (LIWC + Stanza).

In all cases, the early stopping mechanism described above has continued to be used. The result obtained is detailed in the next section.

4. Results

This section describes the results obtained with the three strategies described. In the evaluation of results we have used F1-macro measure, which is the measure proposed by the organizers of the task.

Table 2 shows the averaged results of the transformer models after 10 runs with different random seeds. These results have been separated into three groups: the transformer models with a sequence size of 128 tokens without expansion of the *training* dataset, the Longformer model, and the transformer models with a sequence size of 128 tokens in which the *training* dataset has been expanded. Using the Longformer, the highest average F1 (0.486) is obtained with a sequence size of 512 tokens and the two configurations of this model clearly outperform all the models in the first group. This average value of F1 measure in the best Longformer configuration is exceeded (0.496) by the funnel transformer model with expansion of the *training* dataset and leaving a size of 10% in the *dev* dataset.

Table 3 shows the results of the ensemble classifier also averaged for executions on ten different random seeds. On this occasion, and given the large number of hyper-parameters evaluated in the grid search, we have ordered the results by average F1 measure, showing only the most significant configurations, although the rank column allows us to place each configuration displayed in its absolute position.

In this way, we can see that the best result (0.477) is obtained when the ensemble is configured with two hidden layers, a dropout of 0, the activation function *tanh*, the transformer model *bert-base-uncased* is loaded, and the two sets of discrete features: LIWC and Stanza are used in the FFNN input.

This average value of F1 is lower than that of the funnel transformer model alone, but higher than that obtained with the loaded model (*bert-base-uncased*) when used alone. In our tests we have detected that the ensemble with the funnel transformer model behaves quite unstable when it is executed with different random seeds, which means that the average value of the ensemble classifier is lower than expected with this configuration.

On the other hand, in tables 4 and 5 we can see, respectively, the best values obtained for any random seed in transformer models and in the ensemble classifier.

Table 2Average results on our *test* dataset with transformer models

Configuration	Precision.	Recall	F1
funnel 128 tokens	0.483	0.453	0.448
bert cased 128 tokens	0.464	0.423	0.409
bert uncased 128 tokens	0.464	0.432	0.420
albert 128 tokens	0.487	0.447	0.448
longformer max length 512	0.517	0.488	0.486
longformer max length 1024	0.533	0.490	0.484
funnel dev split 20%	0.489	0.481	0.479
bert cased dev split 20%	0.441	0.441	0.438
bert uncased dev split 20%	0.447	0.439	0.437
albert dev split 20%	0.443	0.437	0.435
funnel dev split 10%	0.503	0.496	0.496
bert cased dev split 10%	0.464	0.457	0.455
bert uncased dev split 10%	0.471	0.465	0.464
albert dev split 10%	0.473	0.463	0.463

Table 3Average results on our *test* dataset with ensemble classifier sorted by F1

Configuration	Precision.	Recall	F1	Rank
bert uncased+all, 2 hidden layers, tanh, dropout=0	0.485	0.479	0.477	1
bert uncased+Stanza, 2 hidden layers, sigmoid, dropout=0.2	0.477	0.477	0.474	2
bert uncased+Stanza, 2 hidden layers, sigmoid, dropout=0	0.477	0.478	0.473	3
albert+LIWC, 1 hidden layer, relu, dropout=0.5	0.480	0.475	0.473	4
albert+LIWC, 1 hidden layer, sigmoid, dropout=0.5	0.480	0.475	0.473	5
albert+all, 2 hidden layers, tanh, dropout=0.2	0.483	0.472	0.472	10
albert+Stanza, 1 hidden layer, relu, dropout=0	0.478	0.473	0.472	12
funnel+all, 2 hidden layers, relu, dropout=0.5	0.466	0.467	0.461	90
funnel+LIWC, 2 hidden layers, sigmoid, dropout=0.5	0.457	0.469	0.458	98
bert cased+all, 2 hidden layers, relu, dropout=0.2	0.462	0.459	0.457	99
bert cased+Stanza, 2 hidden layers, relu, dropout=0.2	0.465	0.458	0.456	101
bert uncased+LIWC, 1 hidden layer, sigmoid, dropout=0	0.448	0.452	0.441	158
funnel+Stanza, 2 hidden layers, relu, dropout=0.2	0.437	0.457	0.438	164
bert cased+LIWC, 1 hidden layer, relu, dropout=0	0.435	0.453	0.435	175

In this case, the Longformer model and the transformer models with extended *training* dataset obtain a very similar F1 measure: 0.536 for the Longformer with a maximum sequence size of 1024 tokens and a random seed of 42, and 0.535 for the funnel transformer model with a *dev* partition of 10% and random seed 79. The transformer models without expansion of the *training* dataset fall behind the Longformer model although they surpass some configurations that use the extended dataset.

Table 5 shows the results of the grid search for the most significant configurations ordered by F1 measure. Here, the extreme behavior of the ensemble classifier is clearly seen when the

Table 4Best results on our *test* dataset with transformer models

Configuration	Precision	Recall	F1	Seed
funnel 128 tokens	0.512	0.490	0.488	33
bert cased 128 tokens	0.578	0.471	0.470	85
bert uncased 128 tokens	0.523	0.499	0.496	0
albert 128 tokens	0.520	0.469	0.480	54
longformer max length 512	0.530	0.529	0.516	54
longformer max length 1024	0.564	0.524	0.536	42
funnel dev split 20%	0.520	0.527	0.521	42
bert cased dev split 20%	0.462	0.459	0.457	0
bert uncased dev split 20%	0.464	0.461	0.462	42
albert dev split 20%	0.474	0.465	0.465	85
funnel dev split 10%	0.532	0.549	0.535	79
bert cased dev split 10%	0.481	0.467	0.470	63
bert uncased dev split 10%	0.498	0.487	0.491	0
albert dev split 10%	0.502	0.498	0.496	12

Table 5Best results on our *test* dataset with ensemble classifier sorted by F1

Configuration	Precision	Recall	F1	Seed	Rank
funnel+LIWC, 2 hidden layers, sigmoid, dropout=0	0.544	0.559	0.546	79	1
funnel+LIWC, 2 hidden layers, sigmoid, dropout=0.5	0.542	0.560	0.544	79	2
funnel+Stanza, 2 hidden layers, sigmoid, dropout=0	0.542	0.557	0.543	79	3
funnel+all, 2 hidden layers, sigmoid, dropout=0	0.540	0.558	0.541	79	4
funnel+all, 2 hidden layers, sigmoid, dropout=0.2	0.536	0.553	0.541	79	5
bert uncased+all, 2 hidden layers, tanh, dropout=0	0.529	0.526	0.518	79	116
albert+LIWC, 2 hidden layers, sigmoid, dropout=0.2	0.506	0.516	0.507	12	165
albert+Stanza, 2 hidden layers, relu, dropout=0.2	0.515	0.509	0.506	33	166
albert+all, 1 hidden layer, relu, dropout=0	0.524	0.512	0.504	12	183
bert uncased+Stanza, 2 hidden layers, relu, dropout=0	0.514	0.508	0.498	79	238
bert cased+all, 2 hidden layers, relu, dropout=0.2	0.499	0.494	0.496	0	249
bert cased+Stanza, 2 hidden layers, sigmoid, dropout=0	0.498	0.497	0.493	0	281
bert uncased+LIWC, 1 hidden layer, relu, dropout=0	0.503	0.496	0.492	0	304
bert cased+LIWC, 2 hidden layers, sigmoid, dropout=0.5	0.579	0.493	0.490	85	355

funnel transformer model is loaded: the best and worst results (the latter are not shown in the table) are for this configuration. In addition, the best value of F1 (0.543) associated with the configuration where the ensemble uses two hidden layers, a dropout of 0, the activation function *sigmoid*, and the discrete LIWC features, clearly exceeds that obtained by the transformers alone (0.536).

Seeing these results on our *test* dataset and after verifying that the average and maximum values differ mainly due to the behavior of the funnel transformer model when it is loaded in the ensemble classifier, we have chosen to send our submissions guided by the highest absolute

value on any random seed.

In this way, we have made a submission with the best value obtained with the Longformer model, with the best value obtained with the transformers with expanded training dataset, and our final submission has been the one corresponding to the ensemble classifier shown in first position in Table 5.

With this latest submission we have obtained the second best F1 measure (0.3324) among the 25 participating teams with a difference of 0.0066 compared to the team ranked first.

5. Conclusions and Future Work

To tackle the task of detecting fake news in this edition of the CheckThat! Lab, our team has evaluated three strategies that involve the use of transformer models. The first one is the use of an architecture such as the Longformer model, which is specially designed to work with long input sequences without penalizing the performance obtained during the training process. This capability seems adequate to the task, given that we are working with news items that can have thousands of words.

Comparing its results with those of transformer models with smaller sequence sizes, we find that both in absolute measures of F1 and averaged over different random seeds, the Longformer model has superior performance.

The second strategy used has been to expand the *training* dataset in chunks of 128 words, which is the sequence size that we have configured by default in all models. For this, we have assumed that if an article is *true*, *partially false*, or *false*, it will still have the same truth value if we chop it up, since the latent features that the transformer model extracts must be similar. Making this assumption, our results show that in certain configurations this expansion is beneficial, surpassing even the Longformer model in the average F1 measurement.

Finally, we wanted to test if the combination of discrete features and a multi-layer classifier together with a transformer model with its typical input add an additional ability to classify news items according to their truth value. For this goal, we have developed an ensemble classifier that is loaded with two pre-trained models separately in the *training* dataset: an FFNN that has as inputs the features extracted by the LIWC text analysis tool together with features elaborated from the information returned by the NLP Stanza tool and, on the other hand, a transformer model. With this architecture, we have performed a grid search, verifying that in absolute values using the LIWC features and the sequence of words from the article as inputs obtains the best F1 value among all the evaluated strategies.

In the future, we plan to continue investigating other ways of integrating different models in an ensemble classifier, as well as including knowledge-based features that, on the one hand, help to identify this type of news, and on the other hand, allow to justify the predictions of the classifier.

Acknowledgments

This work has been partially supported by the Spanish Ministry of Science and Innovation within the projects DOTT-HEALTH (PID2019-106942RB-C32) and RAICES (IMIENS 2022).

References

- [1] P. Nakov, A. Barrón-Cedeño, G. Da San Martino, F. Alam, J. M. Struß, T. Mandl, R. Míguez, T. Caselli, M. Kutlu, W. Zaghoulani, C. Li, S. Shaar, G. K. Shahi, H. Mubarak, A. Nikolov, N. Babulkov, Y. S. Kartal, J. Beltrán, The CLEF-2022 CheckThat! Lab on Fighting the COVID-19 Infodemic and Fake News Detection, in: M. Hagen, S. Verberne, C. Macdonald, C. Seifert, K. Balog, K. Nørvåg, V. Setty (Eds.), *Advances in Information Retrieval*, Springer International Publishing, Cham, 2022, pp. 416–428.
- [2] P. Nakov, A. Barrón-Cedeño, G. Da San Martino, F. Alam, J. M. Struß, T. Mandl, R. Míguez, T. Caselli, M. Kutlu, W. Zaghoulani, C. Li, S. Shaar, G. K. Shahi, H. Mubarak, A. Nikolov, N. Babulkov, Y. S. Kartal, J. Beltrán, M. Wiegand, M. Siegel, J. Köhler, Overview of the CLEF-2022 CheckThat! Lab on Fighting the COVID-19 Infodemic and Fake News Detection, in: *Proceedings of the 13th International Conference of the CLEF Association: Information Access Evaluation meets Multilinguality, Multimodality, and Visualization, CLEF '2022*, Bologna, Italy, 2022.
- [3] J. Köhler, G. K. Shahi, J. M. Struß, M. Wiegand, M. Siegel, T. Mandl, M. Schütz, Overview of the CLEF-2022 CheckThat! Lab Task 3 on Fake News Detection, in: *Working Notes of CLEF 2022—Conference and Labs of the Evaluation Forum, CLEF '2022*, Bologna, Italy, 2022.
- [4] 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*. Publisher: Elsevier.
- [5] G. K. Shahi, D. Nandini, FakeCovid—A multilingual cross-domain fact check news dataset for COVID-19, *arXiv preprint arXiv:2006.11343 (2020)*.
- [6] V. Pérez-Rosas, B. Kleinberg, A. Lefevre, R. Mihalcea, Automatic Detection of Fake News, in: *Proceedings of the 27th International Conference on Computational Linguistics*, 2018, pp. 3391–3401.
- [7] J. W. Pennebaker, R. L. Boyd, K. Jordan, K. Blackburn, The development and psychometric properties of LIWC2015, *Technical Report*, 2015.
- [8] V. L. Rubin, T. Lukoianova, Truth and deception at the rhetorical structure level, *Journal of the Association for Information Science and Technology 66 (2015) 905–917*. Publisher: Wiley Online Library.
- [9] J. P. Dickerson, V. Kagan, V. Subrahmanian, Using sentiment to detect bots on Twitter: Are humans more opinionated than bots?, in: *2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014)*, IEEE, China, 2014, pp. 620–627. URL: <http://ieeexplore.ieee.org/document/6921650/>. doi:10.1109/ASONAM.2014.6921650.
- [10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, \. Kaiser, I. Polosukhin, Attention is all you need, in: *Advances in neural information processing systems*, 2017, pp. 5998–6008.
- [11] M. Schütz, A. Schindler, M. Siegel, K. Nazemi, Automatic fake news detection with pre-trained transformer models, in: *International Conference on Pattern Recognition*, Springer, 2021, pp. 627–641.
- [12] 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), 2019, pp. 4171–4186.
- [13] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, R. Soricut, ALBERT: A Lite BERT for Self-supervised Learning of Language Representations, arXiv:1909.11942 [cs] (2020). URL: <http://arxiv.org/abs/1909.11942>, arXiv: 1909.11942.
- [14] 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, arXiv:1907.11692 [cs] (2019). URL: <http://arxiv.org/abs/1907.11692>, arXiv: 1907.11692.
- [15] V. Sanh, L. Debut, J. Chaumond, T. Wolf, DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter, arXiv:1910.01108 [cs] (2020). URL: <http://arxiv.org/abs/1910.01108>, arXiv: 1910.01108.
- [16] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, Q. V. Le, XLNet: Generalized Autoregressive Pretraining for Language Understanding, arXiv preprint arXiv:1906.08237 (2019).
- [17] K. Shu, D. Mahudeswaran, S. Wang, D. Lee, H. Liu, Fakenewsnet: A data repository with news content, social context, and spatiotemporal information for studying fake news on social media, *Big data* 8 (2020) 171–188. Published in Volume: 8 Issue 3: June 1, 2020.
- [18] S. Gundapu, R. Mamidi, Transformer based Automatic COVID-19 Fake News Detection System, arXiv:2101.00180 [cs] (2021). URL: <http://arxiv.org/abs/2101.00180>, arXiv: 2101.00180.
- [19] D. Mehta, A. Dwivedi, A. Patra, M. Anand Kumar, A transformer-based architecture for fake news classification, *Social Network Analysis and Mining* 11 (2021) 39. URL: <https://link.springer.com/10.1007/s13278-021-00738-y>. doi:10.1007/s13278-021-00738-y.
- [20] I. Beltagy, M. E. Peters, A. Cohan, Longformer: The Long-Document Transformer, arXiv:2004.05150 [cs] (2020). URL: <http://arxiv.org/abs/2004.05150>, arXiv: 2004.05150.
- [21] P. Qi, Y. Zhang, Y. Zhang, J. Bolton, C. D. Manning, Stanza: A Python Natural Language Processing Toolkit for Many Human Languages, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, Association for Computational Linguistics, Online, 2020, pp. 101–108. URL: <https://www.aclweb.org/anthology/2020.acl-demos.14>. doi:10.18653/v1/2020.acl-demos.14.