NLP&IR@UNED at CheckThat! 2021: Check-worthiness estimation and fake news detection using transformer models

Juan R. Martinez-Rico[1], Juan Martinez-Romo[1,2] and Lourdes Araujo[1,2]

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

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

Abstract

This article describes the different approaches used by the NLPIR@UNED team in the CLEF2021 Check-That! Lab to tackle the tasks 1A-English, 1A-Spanish and 3A-English. The goal of Task 1A in English is to determine which tweets within a set of COVID-19 related tweets are worth checking. Task 1A in Spanish is similar but in this case the tweets are related to political issues in Spain. In both tasks, transformer models have been used to identify check-worthy tweets, obtaining the first place in the task in English and the fourth place in the task in Spanish. Task 3A is focused on determining the veracity of 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 used two different approaches: a gradient-boosting classifier with TF-IDF and LIWC features, and a transformer model fed with the first tokens of each news article. We got the fourth place out of 25 participants in this task.

Keywords

check-worthiness, fake news detection, transformer models

1. Introduction

Despite the efforts carried out in recent times to combat the proliferation of fake news, these have not stopped growing, taking advantage of events conducive to its dissemination, such as the current pandemic, or the events that occurred in the last presidential elections in the United States. Therefore, the existence of initiatives such as this CheckThat! Lab[1][2], which give researchers in this area of natural language processing the opportunity to propose and share different ideas that can help mitigate this problem, are appreciated.

In this article, we present the approaches used by our team in the tasks of check-worthiness and fake news detection. Since transformer models have become a fundamental tool that adapts to many of the tasks related to natural language processing obtaining state-of-the-art results, we have chosen to take them as our first option in each of the tasks. However, in Task3a we

decided to also use more classical approaches since the size of the news articles to be checked exceeded the input sequence size that is reasonable to define in a transformer model.

We have organized the rest of the article as follows: in section 2 we briefly describe the transformer models, the approach we have used in tasks 1A-English and 1A-Spanish and we comment on the results obtained, in section 3 we explain our approach in the fact-checking task and discuss the results obtained, and section 4 contains our conclusions and future work.

2. Transfomers for Check-Worthiness

2.1. Previous Approaches in the Check-Worthiness Task

Among the approaches that have been used to tackle this task we can highlight the initial work carried out by [3] where they make use of classifiers such as Random Forest, SVM or Multinomial Naive Bayes, and features based on TF-IDF representations, parts of speech tags, sentiment scores, and entity types. To the aforementioned methods [4] add features such as average embedding vector of the sentence, linguistic features that count the number of words in the sentence that belong to a certain lexicon, contextual features such as the position of a sentence with respect to others in a segment of text, discourse features such as the detection of contradictions, and as a classifier uses a Deep Feed-Forward Neural Network. Already within this Check That! Lab we have seen in past editions the use of recurrent neural networks by [5] where each token is represented in three ways: through embeddings, and with part of speech tags and syntactic dependencies encoded as one-hot vectors. In the same edition [6] makes use of character n-gram features with a k-nearest neighbors classifier. More recently in this same Lab, transformer models began to be used for the check-worthiness task by many of the participants [7][8][9]. In the next section we will see a short description of this architecture.

2.2. The Transformer Model

Since its appearance as an alternative to neural machine translation models, transformer models[10] have become a preferred model when compared to other natural language processing techniques, not only in machine translation, but in other tasks such as sequence classification, summarization, named entity recognition, text generation, extractive question answering or language modeling.

A transformer is a deep learning model that "translates" input sequences into output sequences using an encoder-decoder architecture. It uses an attention mechanism to identify the most relevant parts of the input and output sequences. Previous models such as RNNs also use an attention mechanism but are limited by their sequential nature when processing input data. Transformers, by relying solely on the attention mechanism, do not need to process the input sequences in a specific order, allowing them to process these sequences in parallel and thus reducing training times.

The model is fed with training data in the form of sequence pairs (input, target). The first is applied in the encoder block and the second in the decoder block.

In recurrent models, sequences are introduced token by token, thus providing the relative position of each of these tokens in the sequence. Since transformers do not process sequences

in this way, this positional information is provided to the model as a mask added to the input and target sequences.

The encoder block is made up of a stack of *n* identical encoders, each of them with a self-attention layer and a feed fordward neural network. The decoder block is made up of the same number *n* of decorders and each of them is composed of a self-attention layer, an encoder-decoder attention layer and a feed forward neural network.

The self-attention layers allow to identify within the same sequence, which tokens are more relevant for another token that is being considered at that moment. On the contrary, the encoder-decoder attention layer relates tokens of the input and target sequences. The attention layers are not monolithic, but are composed of several attention heads that focus on different portions of the sequence.

The output of the encoder block is the one that feeds all the encoder-decoder attention layers of the decoder block, while the output of the decoder block links with a linear layer and this with a softmax layer that maps each position of the target sequence with the output vocabulary.

What is described above is the original model however, after its presentation a large number of models derived from the transformer architecture have appeared. For example, one of the most successful is BERT[11], which basically eliminates the decoder block present in transformers, and in its training the input sequences are masked in such a way that it processes them bidirectionally.

Another point to highlight is that as part of these architectural-models a series of data-models pre-trained in an unsupervised manner with large datasets have been released. This allows us to easily apply transfer-learning to different tasks such as those mentioned at the beginning of this section.

Next, we will describe how we have used some of these models in the check-worthiness and fake news detection tasks.

2.3. Task 1A English

The objective of Task1a-English[12] is, given a set of tweets in English language related to the COVID-19 topic, to identify which tweets are worth checking by assigning a score to each of them.

To tackle this task we eliminated any metadata present in the tweets and have focused only on the textual information provided.

Taking into account that all the tweets to be evaluated are about COVID-19, we have searched a well-known repository of pre-trained models[1], and we have found one that is trained in tweets related to this topic.

Finally, we have used the BERTweet model[13], a BERT-architecture model initially pre-trained with 850 million tweets in English using the RoBERTa[14] pre-training procedure, to which the same authors performed a second 40-epoch pre-training with 23 million English tweets related to the COVID-19 topic.

To check if actually using a pre-trained model for the same topic and document type had a superior behavior to other pre-trained models and architectures in more neutral datasets, we

---

[1]https://huggingface.co/transformers/

Table 1

Task 1A English - Transformer models analysis: results on dev dataset

Model	Epochs	Batch Size	MAP	F1	P-R	ROC
bertweet-covid19-base-uncased	5	16	**0.849**	0.767	0.848	0.874
bertweet-covid19-base-cased	5	16	**0.845**	0.790	0.843	0.879
bertweet-base	5	10	**0.842**	0.774	0.841	0.873
roberta-base	5	8	**0.793**	0.709	0.791	0.836
funnel-transformer/small	3	8	**0.785**	0.654	0.784	0.783
funnel-transformer/small-base	3	8	**0.785**	0.654	0.784	0.783
funnel-transformer/intermediate	3	8	**0.761**	0.637	0.759	0.768
funnel-transformer/intermediate-base	3	8	**0.761**	0.637	0.759	0.768
distilbert-base-cased	5	8	**0.752**	0.688	0.749	0.790
funnel-transformer/medium	5	8	**0.737**	0.707	0.731	0.820
funnel-transformer/medium-base	5	8	**0.737**	0.707	0.731	0.820
bert-base-cased	5	8	**0.733**	0.672	0.729	0.774
bert-base-multilingual-cased	5	8	**0.726**	0.636	0.722	0.786
albert-base-v2	5	16	**0.694**	0.677	0.691	0.756
distilbert-base-multilingual-cased	5	8	**0.680**	0.697	0.673	0.764

Table 2

Task 1A English - Selected transformer models: results on dev dataset

Model	Epochs	Batch Size	MAP	F1	P-R	ROC
bertweet-covid19-base-uncased	6	14	**0.862**	0.800	0.861	0.874
bertweet-covid19-base-cased	5	14	**0.860**	0.797	0.859	0.883

implemented a grid search procedure in which we varied the number of periods, the size of the lot and the model/architecture used. The rest of the hyperparameters have been kept in the default values that each model has.

Among the transformer models we have tested are BERT, ALBERT[15], RoBERTa, DistilBERT[16], and Funnel-Transformer[17]. Table 1 shows the best results obtained for each model for the mean average precision, F1, precision-recall curve and ROC curve measurements, sorted by mean average precision.

As we can see, the best behavior is obtained with the model that is pre-trained in tweets related to the COVID-19 topic.

Therefore we select the first two models *bertweet-covid19-base-uncased* and *bertweet-covid19-base-cased* and we test various values of the *epsilon* parameter obtaining the best results with the value 2.5×10^{-9}. These results are shown in Table 2.

We also found that although we always initialized the Python, NumPy, and PyTorch random number generators with the same seeds, the same results did not always appear for a given set of parameters. Therefore, to make the final shipments, we do not join the training and dev datasets to have a larger one with which to train the models, but we train the models with

Table 3
Task 1A Spanish - Transformer models analysis: results on dev dataset

Model	Epochs	Batch Size	MAP	F1	P-R	ROC
Electra mrm8488-electricidad-base-discriminator	3	16	**0.495**	0.384	0.492	0.885
BERT Geotrend-bert-base-es-cased	3	8	**0.474**	0.439	0.472	0.874
BERT dccuchile-bert-base-spanish-wwm-cased	3	16	**0.467**	0.458	0.465	0.879
RoBERTa mrm8488-RuPERTa-base	3	8	**0.376**	0.341	0.372	0.836
Electra mrm8488-electricidad-base-generator	5	8	**0.325**	0.130	0.318	0.830

the training dataset and evaluate them with the dev dataset, repeatedly executing the same configurations of parameters and selecting the test files to send from the best results obtained on the dev dataset, assuming that an initial random configuration that behaved well in the dev dataset would also do so in the test dataset.

2.4. Task 1A Spanish

In this version of Task 1A, the set of tweets is defined in Spanish language and these tweets are related to issues of Spanish politics.

As in Task 1A English, we have used several transformer models to evaluate which one best suits these types of tweets. The tested models have been BERT, Electra[18] and RoBERTa.

After a preliminary grid search with different pre-trained models in Spanish and different values of batch size and epochs, keeping the rest of the hyperparameters in their default values, we obtained the results shown in Table 3. The best results are shown for each pre-trained model.

Since the model *Electra mrm8488-electricity-base-discriminator*[2] is the one with a slightly higher result, it is the one we selected for a more exhaustive search for parameters. This Electra model is pre-trained with 20GB of the Spanish-language Oscar corpus[19].

We also realized, extracting the vocabulary from this pre-trained model, that among the first 1000 tokens there were 971 unused tokens of type [unusedNNN].

To see if these tokens could be useful, we pulled all the out-of-vocabulary tokens of the training dataset. From this set of words, we manually selected those that seemed most relevant to us and had three or more appearances, mainly the names of politicians, political parties, the media, and hashtags used in electoral campaigns. In total, the list consisted of 197 tokens.

With this list, we create a dictionary to group tokens that correspond to the same concept. For example, *#PINParental*, *pin* and *parental* were matched with the same *PINParental* token.

In this dictionary, we substitute the tokens on the right side by tokens [unusedNNN] to obtain a match between the out-of-vocabulary tokens with the unused tokens of the model, and both in the training loop and in the evaluation loop we did the replacement of the out-of-vocabulary tokens using this dictionary.

Unfortunately, the results obtained with this strategy were not as expected, obtaining better results without substituting out-of-vocabulary tokens. The best results obtained after repeated

---

[2]https://huggingface.co/mrm8488/electricidad-base-discriminator

Table 4
Task 1A Spanish - Selected transformer models: results on dev dataset

Model	Epochs	Batch Size	MAP	F1	P-R	ROC
mrm8488-elect-base-discr. without replacement	3	12	**0.514**	0.480	0.512	0.878
mrm8488-elect-base-discr. without replacement	3	14	**0.510**	0.472	0.506	0.892
mrm8488-elect-base-discr. without replacement	3	16	**0.509**	0.390	0.506	0.892
mrm8488-elect-base-discr. with replacement	3	18	**0.466**	0.277	0.463	0.870
mrm8488-elect-base-discr. with replacement	6	18	**0.458**	0.417	0.456	0.839
mrm8488-elect-base-discr. with replacement	4	10	**0.452**	0.419	0.449	0.872

Table 5
Task 1A - Submission official results

Task	MAP	MRR	RP	P@1	P@3	P@5	P@10	P@20	P@30
1A Spanish	**0.492**	1.000	0.475	1.000	1.000	1.000	0.800	0.800	0.620
1A English	**0.224**	1.000	0.211	1.000	0.667	0.400	0.300	0.200	0.160

runs with different batch sizes and epochs are shown in Table 4, along with the best results obtained by substituting tokens.

To send the submissions to this version in Spanish of subtask 1A, the same strategy was used as in the English version: training the model repeatedly for the same parameters and send the configurations with the best values in the dev dataset.

2.5. Task 1A Results

Finally, two submissions were made for the Spanish version of Task 1A and three submissions for the English version. The official evaluation measure was mean average precision (MAP). In Spanish we obtained the fourth position among six participants while in English we obtained the first position among ten participants. The results are shown in Table 5.

3. Fake News Detection Task

3.1. Previous Approaches in the Fake News Detection Task

The approaches to the detection of fake news that have been made so far can be divided into three groups: knowledge-based methods, content-based method and context-based methods.

In the former, each claim is compared with a source of evidence that supports that claim. The source of evidence can be a knowledge graph[20] in which case we must extract subject-predicate-object triples from the claim and verify their existence in the graph, or we can be use as a source of evidence the information retrieved from a query to a search engine[21], having then to compare the information obtained with the claim using techniques such as similarity, stance detection, contradiction detection, etc.

Content-based methods only use the textual information present in the document. The features obtained can be latent, such as word or sentence embeddings, or explicit such as TF-IDF vectors, bag of words vectors, word counts[22], psycho-linguistic features[23], etc. Transformers and RNNs can also be considered as a content-based method that uses latent features.

In context-based features the information surrounding the claim is used to verify its degree of truthfulness. Examples of these features can be those based on propagation[24], based on the user's reputation[25], based on their profile[26], etc.

3.2. Task 3A - English

For the fake news detection task in English[27], from a set of news articles we have to classify each item in one of the following categories: true, partially true, false, or other[28][29][30], taking into account the main claim of the news article.

The organizers provided three different training datasets[31], so we joined these three datasets and left 20% as a dev dataset for a total of 760 training instances and 190 validation instances.

To tackle this task we have used two different approaches. The first of them is, as in the tasks dedicated to determining the check-worthiness of a sentence, to use transformer models to check if the latent features that these models extract from the documents can be related to their veracity.

The second approach is to use the more classical ensemble methods together with various types of features such as TF-IDF and LIWC.

3.2.1. Transformer approach

A grid search has been carried out with four different transformer models: ALBERT, BERT, DistilBERT and Funnel-Transformer, and different batch sizes and number of epochs.

Given that one of the limitations of the transformer models is the length of the sequence that they accept as input, we have assumed that the relevant information for each news article is likely to be found at the beginning of it. In this way we have extracted the first 150 and 200 tokens as input for the models. We have also tried to use the first 150 tokens of the article title as input. As some instances had no title, in those cases we have used the first 150 tokens of the article text. The four possible class values have been converted to integer values so that they could be processed correctly.

The Table 6 shows the best results obtained for each transformer model. Given that this is a multi-class classification, we have used precision, coverage and F1 as evaluation measures, taking this last measure as the main one. As can be seen, the title of the article does not seem to contain enough information about its veracity, and a longer sequence length provides better results, as expected.

3.2.2. Ensemble approach

In this second approach we use the random forest[32] and gradient boosting[33] classifiers. We extracted the text of each article and processed it with the LIWC2015[34] text analysis tool,

Table 6

Task 3A - Transformer models analysis: results on dev dataset

Model	Epochs	Batch Size	Input	Prec.	Rec.	F1
albert-base-v2	9	8	Text 200	0.445	0.424	**0.427**
funnel-transformer-intermediate	7	8	Text 200	0.436	0.409	**0.402**
albert-base-v2	8	8	Text 150	0.418	0.398	**0.397**
funnel-transformer-intermediate	9	8	Text 150	0.405	0.394	**0.387**
bert-base-cased	9	8	Text 200	0.383	0.386	**0.382**
distilbert-base-cased	6	8	Text 200	0.397	0.371	**0.374**
bert-base-cased	10	8	Text 150	0.370	0.368	**0.362**
distilbert-base-cased	9	8	Text 150	0.351	0.345	**0.345**
distilbert-base-cased	6	8	Title 150	0.354	0.367	**0.344**
bert-base-cased	6	8	Title 150	0.375	0.375	**0.340**
funnel-transformer-intermediate	8	8	Title 150	0.423	0.329	**0.322**
albert-base-v2	6	8	Title 150	0.335	0.341	**0.316**

obtaining a total of 93 discrete features[3] such as *Analytic*, *Clout*, *Authentic*, *Tone*, etc. The use of LIWC in this task is motivated by the premise that false articles may have certain linguistic features that are not present in legitimate articles, and this can be reflected in the results offered by this tool. We also extract the TF-IDF vectors as features from the text of the articles.

To build the latest feature set, for each article we do a Google search using the article title as query terms.

From the first 20 results obtained, we extract the domain names from each URL and concatenate them, separating them with spaces, constructing text strings with the shape *"www.politifact.com www.reuters.com www.nytimes.com apnews.com ..."*. With these strings we also build a TF-IDF representation. Thus, we assume that if domain names of sites dedicated to fact-checking appear among the first 20 results, that article is at least suspected of containing some controversy. On the other hand, if the domain names are from prestigious media, the original article, true or false, may be important.

To select the proper configuration, we keep the LIWC features fixed, and we try to optionally concatenate the text TF-IDF features and the domain names TF-IDF features.

In Random Forest the number of estimators has been established at 100, the maximum depth of the tree at 1000 and as a criterion to evaluate the split quality *gini* has been used. In Gradient Boosting the number of estimators has also been set to 100 and as a loss function *deviance* has been used. The result of these tests is shown in Table 7.

As can be seen, the Gradient Boosting classifier is superior to Random Forest in all feature configurations. It is also able to take advantage of the information provided by all the concatenated features, while the Random Forest classifier obtains the best result when only the LIWC features are used.

---

[3]These are all the features that this tool provides.

Table 7
Task 3A - Ensemble models and features analysis: results on dev dataset

Model	Domain	Text	LIWC	Prec.	Rec.	F1
Gradient Boosting	true	true	true	0.428	0.369	**0.366**
Gradient Boosting	false	true	true	0.419	0.366	**0.364**
Gradient Boosting	false	false	true	0.420	0.346	**0.338**
Gradient Boosting	true	false	true	0.393	0.343	**0.334**
Random Forest	false	false	true	0.386	0.335	**0.319**
Random Forest	false	true	true	0.574	0.325	**0.303**
Random Forest	true	true	true	0.524	0.306	**0.277**
Random Forest	true	false	true	0.462	0.274	**0.226**

Table 8
Task 3A - Submissions official results

Model	Prec.	Rec.	F1
Gradient Boosting + Domain + Text + LIWC	0.5055	0.4805	**0.4680**
Albert-base + sequence lenght 150	0.4653	0.4109	**0.4237**
Albert-base + sequence lenght 200	0.3779	0.3742	**0.3691**

3.3. Task 3A Results

In this task we have made three submissions. The first one has been generated by Gradient Boosting with the three types of features: LIWC, domain names TF-IDF and text TF-IDF. The second submission we have done with Albert transformer with *albert-base* language model and the article text as input with a sequence length of 150. Moreover, for primary submission we have used the same type of transformer but with a sequence length of 200.

With the best of these submissions we have achieved an F1-macro measure of 0.468 which places us in fourth position among 25 participants.

Table 8 shows our reproduction of the results obtained by the three submissions. Unlike what happened in the dev dataset, with the test dataset the best model has been the Gradient Boosting classifier that uses the features based on LIWC, domain names TF-IDF and text TF-IDF. This tells us that although transformer models can perform well in the fake news detection task with little or no feature engineering, the use of text analysis tools like LIWC along with other handcrafted features can still be useful for profiling fake news.

4. Conclusions and Future Work

In this edition of CheckThat! Lab, our team has explored the two main tasks in detecting fake news: the selection of sentences or tweets to verify and the verification of these elements themselves.

Regarding the check-worthiness task, we have verified that the transformer models can

extract the latent features present in the tweets more efficiently than other methods, although it is necessary to carefully choose the appropriate data model for the task, with large performance differences between some models and others.

Our participation in the English version of this task has been very positive, obtaining the first position, while in the Spanish version we have been in fourth place. We have also detected that in Spanish the mean average precision on the dev dataset (0.495) was much lower than that obtained in English (0.849). This may be due to the fact that the dataset used is not specifically pre-trained on tweets or on Spanish politics.

In the task of detecting fake news we have participated with two different approaches. On the one hand, we have used transformer models trying to extract linguistic features that identify fraudulent articles, and expecting good behavior from them. On the other hand, we have used a fairly simple Gradient Boosting classifier that uses linguistic features extracted through the LIWC tool, TF-IDF text features, and a TF-IDF representation of domain names retrieved from a Google search. We have used this second system as contrastive submission since its results were inferior to those of the transformer models. However, in the test dataset the best performance was obtained with this last model.

Being our first participation in a fake news detection task, the result was positive, obtaining fourth place among 25 participants.

We think that although it can always be improved, the check-worthiness task can be approached reasonably well by means of transformers models, so our future work will be mainly devoted to investigating alternative methods to those used in this laboratory to tackle the task of fact-checking and detection of fake news, for example using knowledge methods to verify claims.

Acknowledgments

References

[1] 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.

[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, 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.

[3] N. Hassan, C. Li, M. Tremayne, Detecting check-worthy factual claims in presidential debates, in: Proceedings of the 24th acm international on conference on information and knowledge management, 2015, pp. 1835–1838.

[4] P. Gencheva, P. Nakov, L. Màrquez, A. Barrón-Cedeño, I. Koychev, A context-aware approach for detecting worth-checking claims in political debates, in: Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017, 2017, pp. 267–276.

[5] C. Hansen, C. Hansen, J. G. Simonsen, C. Lioma, The Copenhagen Team Participation in the Check-Worthiness Task of the Competition of Automatic Identification and Verification of Claims in Political Debates of the CLEF-2018 CheckThat! Lab (2018) 8.

[6] B. Ghanem, M. Montes-y Gomez, F. Rangel, P. Rosso, UPV-INAOE-Autoritas - Check That: Preliminary Approach for Checking Worthiness of Claims (2018) 6.

[7] E. Williams, P. Rodrigues, V. Novak, Accenture at CheckThat! 2020: If you say so: Post-hoc fact-checking of claims using transformer-based models, arXiv:2009.02431 [cs] (2020). URL: http://arxiv.org/abs/2009.02431, arXiv: 2009.02431.

[8] A. Nikolov, G. D. S. Martino, I. Koychev, P. Nakov, Team Alex at CLEF CheckThat! 2020: Identifying Check-Worthy Tweets With Transformer Models, arXiv:2009.02931 [cs] (2020). URL: http://arxiv.org/abs/2009.02931, arXiv: 2009.02931.

[9] G. S. Cheema, S. Hakimov, R. Ewerth, Check_square at CheckThat! 2020: Claim Detection in Social Media via Fusion of Transformer and Syntactic Features, arXiv:2007.10534 [cs] (2020). URL: http://arxiv.org/abs/2007.10534, arXiv: 2007.10534.

[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] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).

[12] S. Shaar, M. Hasanain, B. Hamdan, Z. S. Ali, F. Haouari, M. K. Alex Nikolov, F. A. Yavuz Selim Kartal, G. Da San Martino, A. Barrón-Cedeño, R. Míguez, T. Elsayed, P. Nakov, Overview of the CLEF-2021 CheckThat! Lab Task 1 on Check-Worthiness Estimation in Tweets and Political Debates, in: Working Notes of CLEF 2021—Conference and Labs of the Evaluation Forum, CLEF '2021, Bucharest, Romania (online), 2021.

[13] D. Q. Nguyen, T. Vu, A. T. Nguyen, BERTweet: A pre-trained language model for English Tweets, arXiv preprint arXiv:2005.10200 (2020).

[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] 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.

[16] 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.

[17] Z. Dai, G. Lai, Y. Yang, Q. V. Le, Funnel-Transformer: Filtering out Sequential Redundancy for Efficient Language Processing, arXiv:2006.03236 [cs, stat] (2020). URL: http://arxiv.org/abs/2006.03236, arXiv: 2006.03236.

[18] K. Clark, M.-T. Luong, Q. V. Le, C. D. Manning, ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators, arXiv:2003.10555 [cs] (2020). URL: http://arxiv.org/abs/2003.10555, arXiv: 2003.10555.

[19] P. J. O. Suárez, L. Romary, B. Sagot, A Monolingual Approach to Contextualized Word Embeddings for Mid-Resource Languages, Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (2020) 1703–1714. URL: http://arxiv.org/abs/2006.06202. doi:10.18653/v1/2020.acl-main.156, arXiv: 2006.06202.

[20] G. L. Ciampaglia, P. Shiralkar, L. M. Rocha, J. Bollen, F. Menczer, A. Flammini, Computational Fact Checking from Knowledge Networks, PLOS ONE 10 (2015) e0128193. URL: http://dx.plos.org/10.1371/journal.pone.0128193. doi:10.1371/journal.pone.0128193.

[21] G. Karadzhov, P. Nakov, L. Marquez, A. Barron-Cedeno, I. Koychev, Fully Automated Fact Checking Using External Sources, arXiv:1710.00341 [cs] (2017). URL: http://arxiv.org/abs/1710.00341, arXiv: 1710.00341.

[22] J. T. Hancock, L. E. Curry, S. Goorha, M. Woodworth, On lying and being lied to: A linguistic analysis of deception in computer-mediated communication, Discourse Processes 45 (2007) 1–23. Publisher: Taylor & Francis.

[23] R. Mihalcea, C. Strapparava, The lie detector: Explorations in the automatic recognition of deceptive language, in: Proceedings of the ACL-IJCNLP 2009 Conference Short Papers, Association for Computational Linguistics, 2009, pp. 309–312.

[24] J. Zhang, L. Cui, Y. Fu, F. B. Gouza, Fake news detection with deep diffusive network model, arXiv preprint arXiv:1805.08751 (2018).

[25] P. Nakov, T. Mihaylova, L. Màrquez, Y. Shiroya, I. Koychev, Do not trust the trolls: Predicting credibility in community question answering forums, in: Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017, 2017, pp. 551–560.

[26] K. Shu, X. Zhou, S. Wang, R. Zafarani, H. Liu, The Role of User Profile for Fake News Detection, arXiv:1904.13355 [cs] (2019). URL: http://arxiv.org/abs/1904.13355, arXiv: 1904.13355.

[27] 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.

[28] 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.

[29] G. K. Shahi, D. Nandini, FakeCovid – A Multilingual Cross-domain Fact Check News Dataset for COVID-19, in: Workshop Proceedings of the 14th International AAAI Conference on Web and Social Media, 2020. URL: http://workshop-proceedings.icwsm.org/pdf/2020_14.pdf.

[30] G. K. Shahi, AMUSED: An Annotation Framework of Multi-modal Social Media Data, arXiv preprint arXiv:2010.00502 (2020).

[31] 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.

[32] L. Breiman, Random forests, Machine learning 45 (2001) 5–32. Publisher: Springer.

[33] J. H. Friedman, Stochastic gradient boosting, Computational Statistics & Data Analysis 38 (2002) 367–378. URL: https://linkinghub.elsevier.com/retrieve/pii/S0167947301000652. doi:10.1016/S0167-9473(01)00065-2.

[34] J. W. Pennebaker, R. L. Boyd, K. Jordan, K. Blackburn, The development and psychometric properties of LIWC2015, Technical Report, 2015.