Fight for 4230 at CheckThat! 2021: Domain-Specific Preprocessing and Pretrained Model for Ranking Claims by Check-Worthiness

Xinrui Zhou*[1], Bohuai Wu*[1] and Pascale Fung[1]

[1] *The Hong Kong University of Science and Technology, Clearwater Bay, Kowloon, Hong Kong Special Administrative Region of the People's Republic of China*

Abstract

The widespread dissemination of false news on social media has brought negative effects to society. In this paper, we describe a model submitted to the CLEF-2021 CheckThat! Task 1 - English to estimate the check-worthiness of tweets and political debates/speeches. Our official submission was ranked 2^{nd} in subtask 1A with a MAP score of 0.195 and ranked 1^{st} in subtask 1B with a MAP score of 0.402. The main challenges of the task 1 are the imbalanced data and the not standard texts of tweets. We did thorough data preprocessing and mainly focused on combining different pretrained models with a dropout layer and a dense linear layer. We explored and experimented with many combinations of different data preprocessing techniques and augmentation methods. We also tried extracting additional features from metadata and ensembling the best-performance models to further improve. We have developed a preprocessing procedure for tweets, and our experiments show that domain-specific preprocessing and pretrained models can significantly improve the performance. Finally, we submitted the result produced by the BERTweet model with extra dropout layer and classifier layer with preprocessed data for subtask 1A and RoBERTa model fine-tuned on tweets_hate_speech_detection dataset with extra dropout layer and classifier layer for subtask 1B.

Keywords

check-worthiness, data preprocessing, BERTweet, distilRoBERTa

1. Introduction

Social media becomes increasingly popular for information seeking out and consumption due to its low cost, easy access and rapid dissemination of information, however, it also facilitates the release and dissemination of rumors and false information [1]. The detection of fake news on social media presents unique characteristics, and there are huge differences in content, format, and writing style, which makes the existing detection algorithms of traditional news media ineffective or inapplicable, thus posing new challenges to fake news detection [1]. This problem has become more serious and urgent during the epidemic. As the COVID-19 pandemic spread, social media played an important role in socializing, and also a quick channel to seek and share information about the diseases. This enabled an explosion of unchecked information

[*] These authors contributed equally to this work and should be considered as co-first authors.

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

✉ xzhouaz@connect.ust.hk (X. Zhou*); bwual@connect.ust.hk (B. Wu*); pascale@ece.ust.hk (P. Fung)
🌐 https://pascale.home.ece.ust.hk/ (P. Fung)

and the spread of misinformation [2]. More than 400 people in Iran died from drinking toxic substances due to rumors that high-proof alcohol can cure COVID-19, which have widespread influence on social media [3]. The World Health Organization claimed that a massive amount of misleading information on social media during the epidemic had brought an 'infodemic' and severely threatened public health [4]. Thus, fast false claim identification has become a crucial and challenging task, especially in the epidemic period.

Furthermore, the content of false news related to the epidemic is updated and spread quickly on social media, while manual fact-checking is time-consuming and inefficient. Therefore, it is of great significance to carry out automated false news detection to reduce the human burden.

However, even with automated detection, we can't detect every single claim on social networks. In 2020, there were over 500 million tweets sent every day on average [5]. This poses the need to pre-filter and prioritizes what should be passed to following the fact-checking pipeline, which is namely the task of check-worthiness estimation. A leading contribution in check-worthiness estimation has been the CLEF CheckThat! Lab, which has set up the task of heck-worthiness estimation in the four years' editions. In this paper, we present our approaches in tackling subtask 1A and subtask 1B of the CLEF-2021 CheckThat! Lab in English[6]:

Subtask 1A - Check-worthiness of tweets: Given a topic and a stream of potentially related tweets, rank the tweets according to their check-worthiness for the topic.

Subtask 1B - Check-Worthiness of Debates/Speeches: Given a political debate/speech, produce a ranked list of its sentences, ordered by their check-worthiness. This is a ranking task.

To effectively address this challenge, we mainly focus on the pretrained models with a dropout layer and a dense linear layer, and also explored and experimented with many combinations of different data preprocessing and augmentation methods, with additional features and ensembling methods. The contributions of this paper are mainly from two aspects: we developed a useful automatic preprocessing procedure to effectively process tweets before analysis; Secondly, we show that domain-specific preprocessing and pretrained models can significantly improve the performance to filter out check-worthy claims.

2. Related Work

ClaimBuster system is one of the earliest end-to-end systems for check-worthiness estimation and fact-checking [7]. The ClaimBuster system is still ongoing and could detect claims worth checking on the live discourses, social media and news. It used various supervised learning methods, including Multinomial Naive Bayes Classifier (NBC), Support Vector Classifier (SVM) and Random Forest Classifier (RFC). Moreover, it used different features, such as wording embedding with Part-of-Speech (POS) tags, entity types, sentiment and length and other 100 most important features.

Another online system for check-worthiness detection is ClaimRank, which is trained on actual annotations and can work for various kinds of text [8].

In CLEF2020 CheckThat! Lab competition task1, participants investigated more methods and models than the participants in Check That! Lab 2019. There were several models that have been used by the participants, such as BERT [9, 10, 11, 12], RoBERTa [13, 14, 15], BiLSTM [16], CNN, Random Forest [17], Logistic regression and SVM [9]. Many groups already combined

Table 1
Information of Data in Subtask 1A

Dataset	Total	Check-worthy
Train	822	290
Dev	140	60
Test	350	19

other representations, such as FastText, GloVe, Dependencies, POS and Named entities. To deal with the problems coming from the limited amounts of training data, several groups attempted different external data and graph relations. According to the overviews of the task1 [18], the top-ranked team, Accenture, used RoBERTa with extra mean pooling and dropout layer which were more useful than other data preprocessing.

It is worth mentioning that most of the above systems focused on tweets but we need to deal with both tweets, and speeches and debates. We retry some of the features of these systems, and we focus on the preprocessing techniques to fit the tweets and different combinations of various models and above features.

3. Dataset Analysis and Processing

For tweets on subtask 1A, to better understand the relationship between different features and corresponding check worthiness, we did the exploratory data analysis on the training data to explore the dataset. After that, we did data preprocessing according to the analyses to remove some useless words and modify some abbreviations. To deal with the limitation problem of the training data, we tried data augmentation techniques, such as back translation, to produce more data.

Then, we extracted the different features, such as Node2Vec word embeddings, text meta feature and sentence embeddings produced by different models for training.

3.1. Exploratory Data Analysis

In task 1A, organizers provided datasets of tweets collected from a variety of COVID-19- related topics. The selected tweets are manual annotated and considered as check-worthy if it contains a verifiable factual claim and also needs a professional fact-checker to verify.

We did an exploratory analysis on tweet metadata, including word_count, unique_word_count, stop_word_count, punctuation_count, china_mention_count, url_count, wuhan_mention_count, mean_word_length, char_count, hashtag_count, @_count, number_count, among which we found that most of the meta-features have information about a target, as have very different distributions for check-worthy and non- check-worthy tweets, such as stop_word_count, mean_word_length. These features might be useful in models. It looks like check-worthy tweets are written more formally with longer words compared to non-check-worthy tweets.

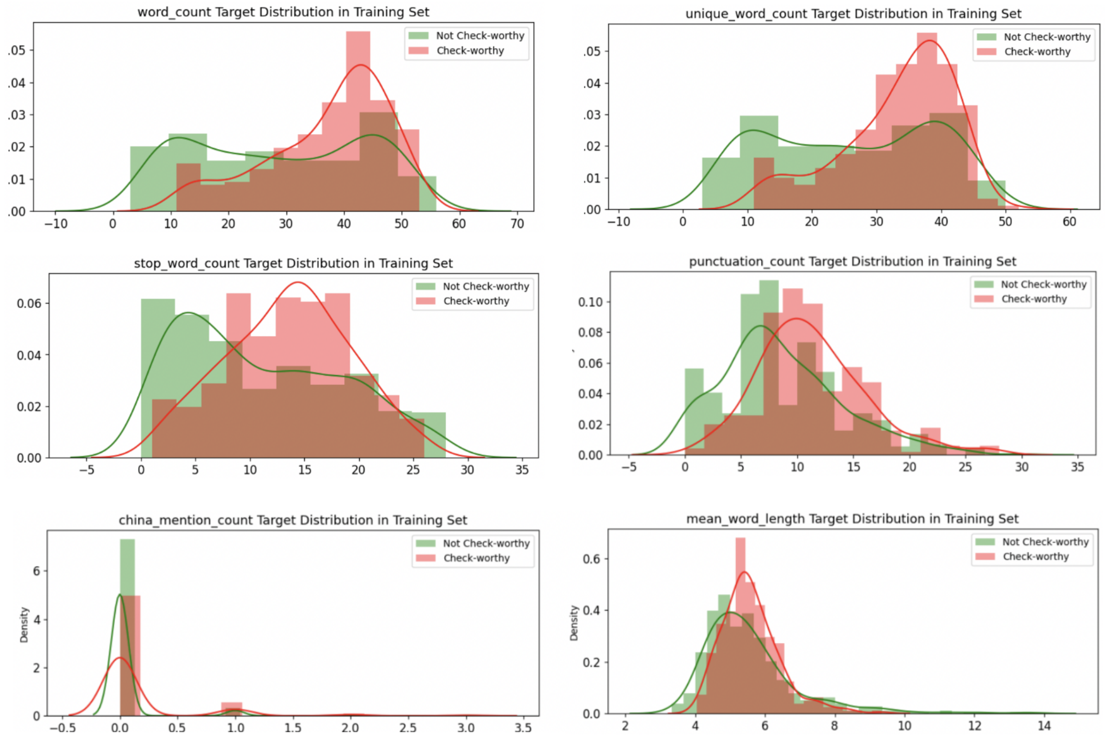

Figure 1: Distributions of word_count, unique_word_count, stop_word_count, punctuation_count, china_mention_count and mean_word_length.

3.2. Data preprocessing

We applied different techniques to preprocess the raw tweet text and developed many handcrafted domain-specific dictionaries. Our preprocessing procedure includes the following processing rules and orders:

- `Normalize all punctuations to English`
- `Clear entity reference`
- `Remove all links`
- `Clean punctuations, non_ASCII except emojis`: In the BERTweet model, each emoji icon is translated into text strings as a word token using the emoji package[1]. Emoji icons could reveal the sentence sentiment and are relevant to check-worthiness.
- `Expanding shortened quantities`: The presence of quantities and numbers can influence the check-worthiness label. We replaced tokens such as 6m, 12k, 8wks with expanded quantities like 6 million, 12 thousand, 8 weeks.
- `Expand contractions`: We used a handcrafted dictionary to expand contractions and correct misspelling in contractions, such as y'all, youve.

[1] Available at https://pypi.org/project/emoji/

- `Unification of all coronavirus synonyms`: The dataset contains different forms of hashtags that refer to COVID_19, such as #covid2019, #CoronaVirus, #coronavirus19, we unified all coronavirus synonyms to the term coronavirus, including different spelling such as #korona, #koronawirus. We also expand all word forms and hashtags that contain the word corona, such as replacing CoronaOutbreak with coronavirus outbreak.
- `Transform slang`: The tweets dataset contains many slangs that can affect the semantic meaning of the sentences. We developed a handcraft dictionary to transform them into the form that models can recognize, such as transform w/o to without, lmao to laughing, RT to retweet.
- `Expand hashtags`: Many hashtags are used directly as subjects or objects in tweets, therefore, expanding them can better help the model understand the semantics. We used a handcraft dictionary to expand them, such as POTUS to the president of the United States. We tried to use the `wordninja` package[2] which splits hashtags into separate words based on probability. However, it has poor performance on our dataset since it can poorly identify which words need to be processed and could mistakenly split words like "the" to "t" and "he".

3.3. Data Augmentation

The size of training data is small and easily results in overfitting. To increase the amount of data and help reduce overfitting, we employ some data augmentation methods.

3.3.1. Back Translation

We translate each tweet text to the temporary destination language (French and Dutch) and then translate back the previously translated text into English to produce new sentences with different expressions of the same meanings. After back translation, we have a training data size 3 times larger.

3.3.2. Synonym Replacement

We also use WordNet to identify relevant synonyms, and randomly select n words to replace them by their synonyms in the tweet text to produce new sentences.

3.4. Features Extraction

3.4.1. Word2vec Word Embedding (WE)

Word embeddings are able to catch semantic and syntactic features of words. Thus, we represent each sentence as the average vector of its words. We use word2vec models pretrained on Google news, which provides a vector size of 300.

[2] Available at https://pypi.org/project/wordninja/

3.4.2. Text Meta Feature (TMF)

Metadata of tweets might be an indicator for check-worthy claims. We used the following information of tweets as features: word count, count of a hashtag, presence of a link, punctuation count.

3.4.3. Sentence embedding (SE)

After getting the word embeddings from the BERT-based pretrained model, we used the original word embedding matrix, average embedding for word embeddings of all words and the concatenated embeddings of all words as the three different sentences embeddings.

4. Models

In this part, we used the word embeddings and other features with the various models to train the training data and test using the validation dataset. We compared different methods to get a whole performance picture about different natural language processing methods for check worthiness tasks.

4.1. Bert-based classification model

BERT [19], RoBERTa [20], BERTWWM and BERTweet [21] models are the principal models that we have used to train the training dataset for subtask1A and the distilRoBERTa [22] models for subtask1B.

For BERT, we used the BERT-large pretrained model with 24 transformers, 1024 hidden sizes and 16 self-attention heads with totally 336M parameters on lower-case English texts. Similarly, the RoBERTa and BERTWWM also used the same architecture with the BERT-large pretrained model with 355M parameters and 336M parameters respectively.

BERTweet is the first public large-scale pretrained language model for tweets with the BERT BASE architecture and RoBERTa training procedure [20]. BERTweet produces better performance than the previous state-of-the-art models on POS tagging, named-entity recognition and text classification tasks on the English tweets. Therefore, our models mainly used the pretrained BERTweet model released by VinAI. And also, our final model is the BERTweet model with preprocessing for data for subtask 1A.

The distilRoBERTa is the distilled version of RoBERTa models, which is faster than the original RoBERTa model. And for subtask 1B, we mainly used the distilroberta-finetuned-tweets-hate-speech model which is the distilroberta-base model architecture fine-tuned on the tweets-hate-speech dataset released by mrm8488.

4.2. Ensembling Models

For ensembling different natural language processing tools, we also tried several ensembling models. The first model is the combination of 4 top models with voting or weights. Second, we fed the sentence embedding got from the BERTweet model into the AdaBoost regressor [23] or logistic regression [24]. The third one is that we put prediction values from the Bert-based

Table 2
Original Baselines

Model	MAP
Random Baseline	0.4795
Ngram Baseline	0.5916

Table 3
BERT-based Models With One Extra Dropout And Classifier Layer.

Model	MAP	P@3	P@5	P@10
BERT	0.7074	1	1	1
BERTWWM	0.8030	1	1	1
RoBERTa	0.7765	1	1	1
BERTweet	0.8136	1	1	1
BERTweet w/ preprocessed	**0.8753**	1	1	1
BERTweet w/ augmented	0.8205	1	1	1

classification model and metadata as features and combined them with sentence embeddings as the new sentence representation then fed the new embeddings into the AdaBoost regressor, logistic regression or SVM [25].

5. Experiments

In this part, our projects will present the experiments that have been done for subtask1A. The results will include the measurement of precision@K (K is 3, 5, 10) and Mean Average Precision (MAP) comparison among two original baselines and different models, followed by the analyses for the improvement.

5.1. Experiments for Subtask 1A

For subtask1A, our experiments can be divided into 3 parts, including, the comparison among BERT-based models with one dropout and classifier layer, all BERT-based models with KFold algorithm, and different ensembling models.

5.1.1. BERT-based Models

In this part, we used the original BERT-large model, BERTWWM, RoBERTa-large, and BERTweet models with one extra dropout layer and one classifier layer which is a dense linear layer. Moreover, we also trained the models with preprocessed data and augmented datasets.

Originally, there are two official baselines, Random baseline, and Ngram baseline which use random guess and ngram prediction respectively. Table 2 shows the MAP results of Random Baseline and Ngram Baseline.

Table 4
BERT-based Models With KFold algorithm.

Model	MAP	P@3	P@5	P@10
BERT	0.7234	1	1	1
BERTWWM	0.7752	1	1	1
RoBERTa	0.8005	1	1	1
BERTweet	0.8332	1	1	1
BERTweet w/ preprocessed	**0.8370**	1	1	1

Table 5
Ensembling Models.

Model	MAP	P@3	P@5	P@10
BERTWWM +SE(mean) +Adaboost	0.7243	1	1	1
BERTWWM +SE(concat) +Adaboost	0.7134	1	1	1
Node2vec +LR	0.6454	1	1	1
BERTWWM +Node2vec+LR	0.7055	1	1	1
BERTWWM pred +Node2vec +LinearSVC	0.7759	1	1	1
Voting	**0.8547**	1	1	1

In Table 3, for each model, we trained 3 epochs with 16 batch size and 128 max sequence length. The learning rates we have used are 3e-5 and 5e-5. According to the experiments, the BERTweet model with one dropout layer and one classifier layer achieves the highest accuracy.

5.1.2. KFold BERT-based Models

According to Table 4, we used 20 splits of StratifiedKFold to train the same BERT-based models.

5.1.3. Ensembling Models

Table 5 shows the several methods of ensembling. We used BERTWWM sequence embedding with mean or concatenation method with AdaBoost regressor. Also, we put other text meta-features. Also node2vec was combined into a logistic regression model. Moreover, we tried to consider the prediction value as a novel feature and fed it into the LinearSVC with the node2vec sequence embedding. Finally, we combined some models with different voting weights.

Table 6
DistilRoBERTa for Subtask 1B

Model	MAP
distilroberta + dropout + classifier	0.1696

Table 7
Final Models for Subtask 1A and 1B

Tasks	Subtask 1A	Subtask 1B
Model	BERTweet + Dropout + Classfier w/ preprocessed data	distilroberta + dropout + classifier
MAP	0.195	0.402
P@3	0.333	0.833
P@5	0.400	0.750
P@10	0.400	0.600

5.2. Experiments for Subtask 1B

For subtask 1B, according to the experiments for subtask 1A, we simply used the distilroberta-finetuned-tweets-hate-speech model with one dropout layer and one classifier layer. And the Table 6 shows the average MAP for 9 different speeches.

6. Results and Discussion

According to Table 3, it cannot be denied that BERT-based models have a strong ability to deal with the classification task. Among BERT, BERTWWM, RoBERTa and BERTweet models, RoBERTa, BERTWWM and BERTweet are more powerful than the basic BERT model. It is due to the development of the masking pattern used by RoBERTa and BERTWWM, and the domain-specific pretraining by the BERTweet model. And also, through the experiments of using augmented data and preprocessed data, both the augmentation and preprocessing can help the models to better understand the training data, especially the preprocessing, however, the Text Meta Features (TMFs) are not very effective.

After comparing the Table 3 and Table 4, the Kfold for training can improve most of the BERT-based models, although the improvement for the BERTweet model with preprocessed data is not large. But Kfold can still be used for a limited dataset with feature fine-tuning.

According to the comparison between Table 5 and Table 3, it shows that one dropout layer with one classifier layer is more useful than a simple adaptive boosting algorithm or logistic regression. The experiments showed that the model with the highest accuracy will dominate other models. Therefore, the ensembling model will have higher accuracy if the weight assigned to the highest model is larger.

Final models for subtask 1A and subtask 1B submitted are shown in Table 7.

7. Conclusion and Future Work

In this paper, we present our models and efforts in Task 1 of CLEF2021 Check That! Lab. For subtask 1A, we used three main methods, BERT-based models with extra dropout layer and classifier layer, KFold algorithm and ensembling models. We adopted various data preprocessing and augmentation techniques to help the system improve the accuracy. The main contributions of this paper are: firstly, the development of a useful automatic preprocessing procedure to effectively process tweets before analysis; Secondly, we show that domain-specific preprocessing and pretrained models can significantly improve the performance to filter out check-worthy claims.

In the final submission of Subtask 1a, our final system is the BERTweet model with one dropout layer and one classifier layer without KFold algorithm on the preprocessed training dataset, which eventually ranked 2^{nd} (out of 9 groups) based on the official evaluation metric. For subtask 1B, we used the distilRoBERTa-finetuned-tweets-hate-speech model followed by one dropout layer and one classifier layer, which finally ranked 1^{st} based on the official evaluation metric.

In future work, we plan to experiment with more ensembling techniques as well as with more extra features such as sentence sentiments, POS tags, social characteristics on tweets like the number of retweets, likes and so on.

Acknowledgments

This research is partially supported by Yu Tiezheng, Dai Wenliang and Ji Ziwei. And also, Zhou Zhuorui supports one of the ensembling models and few ideas about the processing.

References

[1] K. Shu, A. Sliva, S. Wang, J. Tang, H. Liu, Fake news detection on social media: A data mining perspective, ACM SIGKDD explorations newsletter 19 (2017) 22–36.

[2] S. B. Naeem, R. Bhatti, A. Khan, An exploration of how fake news is taking over social media and putting public health at risk, Health Information & Libraries Journal (2020).

[3] B. Trew, Hundreds dead in iran from drinking methanol amid fake reports it cures coronavirus, 2020. URL: https://www.independent.co.uk/news/world/middle-east/iran-coronavirus-methanol-drink-cure-deaths-fake-a9429956.html.

[4] Infodemic, 2020. URL: https://www.who.int/health-topics/infodemic/the-covid-19-infodemic#tab=tab_1.

[5] D. Sayce, The number of tweets per day in 2020, 2020. URL: https://www.dsayce.com/social-media/tweets-day/.

[6] 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.

[7] N. Hassan, G. Zhang, F. Arslan, J. Caraballo, D. Jimenez, S. Gawsane, S. Hasan, M. Joseph, A. Kulkarni, A. K. Nayak, et al., Claimbuster: The first-ever end-to-end fact-checking system, Proceedings of the VLDB Endowment 10 (2017) 1945–1948.

[8] I. Jaradat, P. Gencheva, A. Barrón-Cedeño, L. Màrquez, P. Nakov, Claimrank: Detecting check-worthy claims in arabic and english, arXiv preprint arXiv:1804.07587 (2018).

[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 preprint arXiv:2007.10534 (2020).

[10] R. Alkhalifa, T. Yoong, E. Kochkina, A. Zubiaga, M. Liakata, Qmul-sds at checkthat! 2020: determining covid-19 tweet check-worthiness using an enhanced ct-bert with numeric expressions, arXiv preprint arXiv:2008.13160 (2020).

[11] Y. S. Kartal, M. Kutlu, Tobb etu at checkthat! 2020: Prioritizing english and arabic claims based on check-worthiness, Cappellato et al.[10] (2020).

[12] C.-G. Cusmuliuc, L.-G. Coca, A. Iftene, Uaics at checkthat! 2020: Fact-checking claim prioritization, Cappellato et al.[18] (2020).

[13] 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 preprint arXiv:2009.02431 (2020).

[14] A. Nikolov, G. D. S. Martino, I. Koychev, P. Nakov, Team alex at clef checkthat! 2020: Identifying check-worthy tweets with transformer models, arXiv preprint arXiv:2009.02931 (2020).

[15] T. Sachin Krishan, S. Kayalvizhi, D. Thenmozhi, K. Rishi Vardhan, Ssn nlp at checkthat! 2020: Tweet check worthiness using transformers, convolutional neural networks and support vector machines (2020).

[16] J. Martinez-Rico, L. Araujo, J. Martinez-Romo, Nlp&ir@ uned at checkthat! 2020: A preliminary approach for check-worthiness and claim retrieval tasks using neural networks and graphs, Cappellato et al.[10] (2020).

[17] T. McDonald, Z. Dong, Y. Zhang, R. Hampson, J. Young, Q. Cao, J. Leidner, M. Stevenson, The university of sheffield at checkthat! 2020: Claim identification and verification on twitter, Cappellato et al.[10] (2020).

[18] S. Shaar, A. Nikolov, N. Babulkov, F. Alam, A. Barrón-Cedeno, T. Elsayed, M. Hasanain, R. Suwaileh, F. Haouari, G. Da San Martino, et al., Overview of checkthat! 2020 english: Automatic identification and verification of claims in social media, Cappellato et al.[10] (2020).

[19] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, CoRR abs/1810.04805 (2018). URL: http://arxiv.org/abs/1810.04805. arXiv:1810.04805.

[20] 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 preprint arXiv:1907.11692 (2019).

[21] D. Q. Nguyen, T. Vu, A. T. Nguyen, Bertweet: A pre-trained language model for english tweets, CoRR abs/2005.10200 (2020). URL: https://arxiv.org/abs/2005.10200. arXiv:2005.10200.

[22] V. Sanh, L. Debut, J. Chaumond, T. Wolf, Distilbert, a distilled version of BERT: smaller,

faster, cheaper and lighter, CoRR abs/1910.01108 (2019). URL: http://arxiv.org/abs/1910.01108. arXiv:1910.01108.

[23] D. P. Solomatine, D. L. Shrestha, Adaboost. rt: a boosting algorithm for regression problems, in: 2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No. 04CH37541), volume 2, IEEE, 2004, pp. 1163–1168.

[24] R. E. Wright, Logistic regression. (1995).

[25] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, B. Scholkopf, Support vector machines, IEEE Intelligent Systems and their applications 13 (1998) 18–28.