CODE at CheckThat! 2022: Multi-class fake news detection of news articles with BERT

Olivier Blanc[1], Albert Pritzkau[2], Ulrich Schade[2] and Michaela Geierhos[1]

[1]*Research Institute Cyber Defence and Smart Data (CODE), Bundeswehr University Munich, Germany*
[2]*Fraunhofer Institute for Communication, Information Processing and Ergonomics (FKIE), Germany*

Abstract

The following system description presents our approach for detecting fake news in texts. The given task was formulated as a multi-class classification problem. Our approach is based on the combination of two BERT-based classification models: One model determines whether the textual content is relevant to the task; the second model assigns it a truth value. Starting from a pre-trained model for language representation, we fine-tuned these models on the given classification task in supervised training steps using the annotated data provided.

Keywords

Sequence Classification, Deep Learning, Transformers, BERT

1. Introduction

The proliferation of disinformation online has given rise to a lot of research on automatic fake news detection. CLEF 2022 - CheckThat! Lab [1, 2] considers disinformation as a communication phenomenon. By detecting the use of various linguistic features in communication, it takes into account not only the content but also how a subject matter is communicated.

Shared Task 3 of the CLEF 2022 - CheckThat! Lab [3] defines the following subtasks:

Subtask 3A Given the “textual content” of an article in English, specify a credibility level for the content ranging between “true”, “false”, “partially false”, and “other”.

Subtask 3B Solve Subtask 3A by building a transfer learning model, which is trained on English language and applied to German language.

This paper covers our approach on subtask 3A. To build our models, only textual content is given as input. The system we present in this paper is based on the combination of two

CLEF 2022: Conference and Labs of the Evaluation Forum, September 5–8, 2022, Bologna, Italy
olivier.blanc@unibw.de (O. Blanc); albert.pritzkau@fkie.fraunhofer.de (A. Pritzkau); ulrich.schade@fkie.fraunhofer.de (U. Schade); michaela.geierhos@unibw.de (M. Geierhos)
https://www.unibw.de/datascience/blanc (O. Blanc); https://www.fkie.fraunhofer.de/ (A. Pritzkau); https://www.fkie.fraunhofer.de (U. Schade); https://www.unibw.de/datascience/geierhos (M. Geierhos)
0000-0001-7985-0822 (A. Pritzkau); 0000-0002-8180-5606 (M. Geierhos)

CEUR Workshop Proceedings (CEUR-WS.org)

BERT-based text classifiers [4]: A binary classifier trained on the CheckThat! training set [5] that focuses on identifying articles whose content is not relevant for fake news detection (i.e., articles that belong to the "other" category), and a multi-class classifier trained on a larger dataset that focuses on determining the truth value ("truth", "false", and "partially false") of the textual content.

2. Related Work

A comprehensive survey on fake news and on automatic fake news detection has been presented by Zhou and Zafarani [6]. Based on the structure of data reflecting different aspects of communication, they identified four different perspectives on fake news: (1) the false knowledge it carries, (2) its writing style, (3) its propagation patterns, and (4) the credibility of its creators and spreaders.

CLEF2022 CheckThat! - Task 3 emphasizes communicative styles that systematically co-occur with persuasive intentions of (political) media actors. Similar to de Vreese et al. [7], propaganda and persuasion is considered as an expression of political communication content and style. Hence, beyond the actual subject of communication, the way it is communicated is gaining importance [8].

We build our work on top of this foundation by first investigating content-based approaches for information discovery. Traditional information discovery methods are based on content: documents, terms, and the relationships between them [9]. The methods can be considered as general Information Extraction (IE) methods, automatically deriving structured information from unstructured and/or semi-structured machine-readable documents. Communities of researchers contributed various techniques from machine learning, information retrieval, and computational linguistics to the different aspects of the information extraction problem. From a computer science perspective, existing approaches can be roughly divided into the following categories: rule-based, supervised, and semi-supervised. In our case, we followed the supervised approach by reframing the complex language understanding task as a simple classification problem. Text classification also known as text tagging or text categorization is the process of categorizing text into organized groups. By using Natural Language Processing (NLP), text classifiers can automatically analyze human language texts and then assign a set of predefined tags or categories. Historically, the evolution of text classifiers can be divided into three stages: (1) simple lexicon- or keyword-based classifiers, (2) classifiers using distributed semantics, and (3) deep learning classifiers with advanced linguistic features.

2.1. Deep Learning for Information Extraction

Recent work on text classification uses neural networks, particularly Deep Learning (DL). Badjatiya et al. [10] demonstrated that these architectures, including variants of Recurrent Neural Networks (RNN) [11, 12, 13], Convolutional Neural Networks (CNN) [14], or their combination (CharCNN, WordCNN, and HybridCNN), produce state-of-the-art results and outperform baseline methods (character n-grams, TF-IDF, or bag-of-words representations).

2.2. Deep Learning Architectures

Until recently, the dominant paradigm in approaching NLP tasks has been focused on the design of neural architectures, using only task-specific data and word embeddings such as those mentioned above. This led to the development of models, such as Long Short Term Memory (LSTM) networks or Convolution Neural Networks (CNN), that achieve significantly better results in a range of NLP tasks than less complex classifiers, such as Support Vector Machines, Logistic Regression or Decision Tree Models. Badjatiya et al. [10] demonstrated that these approaches outperform models based on character and word n-gram representations. In the same paradigm of pre-trained models, methods like BERT [4] and XLNet [15] have been shown to achieve state-of-the-art performance in a variety of tasks.

Indeed, the usage of a pre-trained word embedding layer to map text into a vector space and then pass it through a neural network, marked a significant step forward in text classification. The potential of pre-trained language models, as e.g. Word2Vec [16], GloVe [17], fastText [18], or ELMo [19], to capture the local patterns of features to benefit text classification, has been described by Castelle [20]. Modern pre-trained language models use unsupervised learning techniques on large texts corpora to gain some primal 'knowledge' of the language structures. These models are usually fine-tuned for a given task with an additional supervised training step using more specific labeled data.

2.3. About BERT

BERT stands for Bidirectional Encoder Representations from Transformers [4]. It is based on the Transformer model architectures introduced by Vaswani et al. [21]. The general approach consists of two stages: first, BERT is pre-trained on vast amounts of text, with an unsupervised objective of masked language modeling and next-sentence prediction. Second, this pre-trained network is then fine-tuned on task specific, labeled data. The Transformer architecture is composed of two parts, an Encoder and a Decoder, for each of the two stages. The Encoder used in BERT is an attention-based architecture for NLP. It works by performing a small, constant number of steps. In each step, it applies an attention mechanism to understand relationships between all words in a sentence, regardless of their respective position. By pre-training language representations, the Encoder yields models that can either be used to extract high quality language features from text data, or fine-tune these models on specific NLP tasks (Classification, Entity Recognition, Question Answering, etc.).

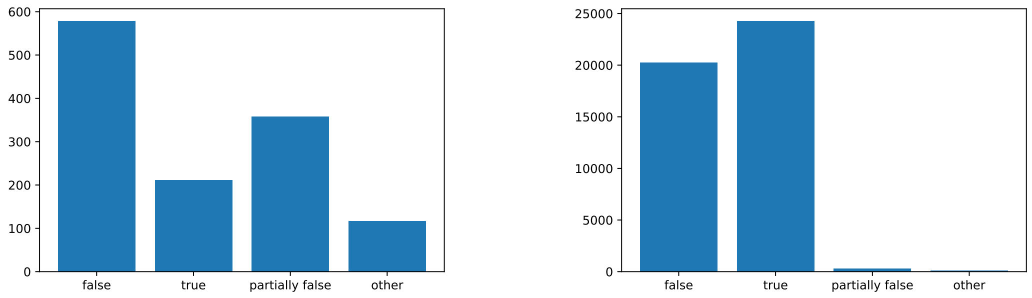
3. Dataset

The dataset for the CLEF2022 CheckThat! - Task 3 was originally developed during the CLEF-2021 CheckThat! campaign [22, 23, 24] and provided by Shahi et al. [25]. The AMUSED framework presented by Shahi [26] was used for data collection. A benchmark classification for the dataset was defined by Shahi and Nandini [27]. The adopted task was framed as multi-class classification problem. Class labels were provided as credibility levels (false, partially false, true, or other) as proposed by Shahi et al. [28]. The initial training dataset consisted of 1,264 documents.

In addition to this training data, we collected other data from external sources suggested by the organizers. We used the dataset built for a similar shared task called Fake News Detection Challenge KDD 2020 [29], as well as the Fake News Classification Datasets [30], a collection of similar datasets for fake news classification, which is available on Kaggle. By combining all the data, we obtained a large training dataset consisting of 44,910 labeled articles.

The exploratory analysis started with the investigation of inconsistencies in the dataset. Unexpectedly, ambiguities in the annotation of the documents could be detected. For example, identical documents were found with contradictory annotations "true" vs. "false". In this case, we decided to remove all affected documents from the training data, as otherwise an alternative decision would have led to a inadvertently weighting of the remaining class. After cleaning up these ambiguities, remaining unique duplicates could be easily removed.

Unbalanced class distribution Imbalance in data can exert a major impact on the value and meaning of accuracy and on certain other well-known performance metrics of an analytical model.



(a) original training data

(b) large corpus

Figure 1: Label distribution in the training sets.

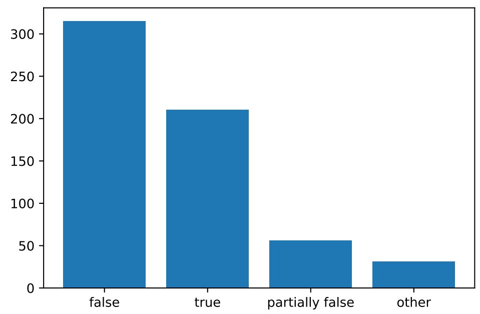


Figure 2: Label distribution in the gold standard.

Figure 1a depicts a clear skew towards false information and Figure 1b towards true information. Furthermore, the "true" class is significantly underrepresented compared to the "partially false" class in the original training data. For the large corpus, it is exactly the opposite.

4. Our Approach

Our approach is based on the combination of two BERT-based text classifiers: a binary classifier that focuses on identifying articles whose content is not relevant to fake news detection (i.e., articles that belong to the category "other"), and a multi-class classifier trained on a large corpus, which focuses on determining the truth value ("true", "false", and "partially false") of the content.

4.1. Baseline Model

As a first attempt, we created a simple BERT-based multi-class classifier model using the Tensorflow/Keras API. The model consists of the following layers:

Preprocessing. This layer lowercases and tokenizes the raw input text and converts it into multiple numeric tensors that will feed the BERT Encoder layer.

BERT Encoder. We use the official BERT model [4] with 12 hidden layers, a hidden size of 768, and 12 attention heads. This model has been pre-trained for English on the Wikipedia and BooksCorpus. All parameters are fine-tuned during training with our dataset.

Dropout. A dropout rate of 0.1 is used for regularization during training.

Linear Classifier. The final layer is a tensor of 4 units, one for each label. The predicted label is determined using the argmax function.

To estimate the performance of our baseline model, we split our dataset with a ration of 82/18 into training and validation set. The model was trained for 10 epochs, minimizing cross entropy loss using AdamW optimizer with an initial learning rate of 3e-5. As shown on the report in Table 1, we achieve an accuracy of 0.96 and a macro F1-score of 0.53 on our validation set. In particular, the baseline model does not return any hit for the label "other" and achieves a low recall of 0.17 for the label "partially true". We tried to improve this in our next experiments.

Table 1
Classification report for our baseline model on our validation dataset.

	precision	recall	F1-score	support
false	0.97	0.95	0.96	3,631
true	0.96	0.98	0.97	4,392
partially false	0.24	0.17	0.20	46
other	0.00	0.00	0.00	15
accuracy			0.96	8,084
macro avg	0.54	0.53	0.53	8,084
weighted avg	0.96	0.96	0.96	8,084

4.2. Text Content Shortening

Transformer-based models are not able to handle long sequences because their self-attention mechanism scales quadratically with sequence length. In particular, our BERT encoder sets a hard limit of 512 tokens. However, most of the text content in our training set exceeds this limit. Therefore, anything beyond this limit is truncated and ignored by our baseline classifier.

To solve this problem, we experimented with feeding our classifier a shorter version of text content. We first try to process the input text using the BERT Extractive Summarizer Python module [31] to create summaries with a maximum length of 500 tokens. We are also experimenting with truncating long texts by simply cutting out the middle and keeping the first 250 tokens at the beginning and the last 250 tokens at the end of each document. In case of short text with less than 500 tokens, the overlapping text segments are duplicated.

4.3. Two Models Approach

In order to get more hits for the label "other", we trained a second binary classifier model that focuses on detecting documents from this category. This model has the same layout as the multi-class classifier and was trained with the small dataset that was provided for this subtask [5] in which the proportion of the label "other" is more important than in our large training set.

The final prediction result is obtained by combining the predictions of the two models: If the binary classifier assigns the label "other" to a document, then this label is selected without considering the label predicted by the multi-class classifier. If this is not the case, the category predicted by the multi-class classifier is retained.

4.4. Early Stopping and F1-Score Monitoring

Finally, we also tuned the training loop in TensorFlow by monitoring accuracy and macro F1-score obtained on the validation data at the end of each epoch. The training loop was stopped earlier if no improvement was observed in any of these value in the last 5 epochs. At the end of the training, the model weight was selected that had the best macro F1-score on the validation data from all iterations, usually at the expense of a small hit on accuracy.

4.5. Preliminary Results on our Test Dataset

Table 2 shows the evaluation results we obtained with our own test data. The different text shortening heuristics are represented: head only, summarization, and head+tail truncation. We can see that all of these variants provide reasonable accuracy. The summarization heuristic gives the lowest macro F1-score. The BERT Extractive Summarizer that we use was primarily

Table 2
Preliminary results on our test dataset.

	head only	summarization	head+tail
accuracy	0.957817	0.964944	0.958436
F1-score	0.604506	0.464823	0.632744

developed for the creation of summaries of lecture courses, and is thus not particularly suited to process the news articles of our dataset. On the other hand, head+tail truncation provides the best macro F1-score of 0.63. This is the variant we keep for our submission and to compare our result with the gold standard.

4.6. Results and Discussion

Table 3
Classification report for the final comparison with the gold standard.

	precision	recall	F1-score	support
false	0.594937	0.746032	0.661972	315
other	0.026316	0.032258	0.028986	31
partially false	0.139073	0.375000	0.202899	56
true	0.535714	0.071429	0.126050	210
accuracy			0.444444	612
macro avg	0.324010	0.306180	0.254977	612
weighted avg	0.504100	0.444444	0.404007	612

The official evaluation results on the test set are shown in the Tables 3 and 4. We observe a significant degradation in performance for both accuracy and F1-score compared to our evaluation on our own test dataset. This suggests a big discrepancy between the gold standard and our training dataset. Further exploratory data analysis would be required.

We focused on appropriate combinations of Deep Learning methods as well as their hyper-parameter settings. Even without extensive pre-processing of the training data, we already obtain competitive results and strong baseline models that, when fine-tuned, significantly outperform models trained from scratch.

When improving the pre-trained baseline models, class imbalance seems to be one of the main challenges. This can be clearly seen in Figure 3. The poor performance, especially for the categories "true" and "other", correlates with the distribution of training data across these categories.

A commonly used tactic for dealing with imbalanced datasets is assigning weights to each label. Alternative solutions for dealing with imbalanced datasets in supervised machine learning include undersampling or oversampling. Undersampling considers only a subset of an overpopulated class to obtain a balanced dataset. With the same goal, oversampling creates copies of the unbalanced classes.

5. Conclusion and Future Work

With the above findings, we achieve state-of-the-art performance in text classification on our validation dataset. The performance decreases significantly on the test data due to a too large gap between the gold standard and our extended training dataset. Nevertheless, BERT has proven to be a powerful language representation model for multi-class text classification. In future work, we plan to investigate more recent neural architectures for language representation such as T5 [32], GPT-3 [33], or its open competitor OPT-175B [34].

Furthermore, we expect great opportunities for transfer learning from the areas such as argumentation mining [35] and offensive language detection [36]. To deal with data scarcity as a general challenge in Natural Language Processing, we examine the application of concepts such as active learning, semi-supervised learning [37] as well as weak supervision [38].

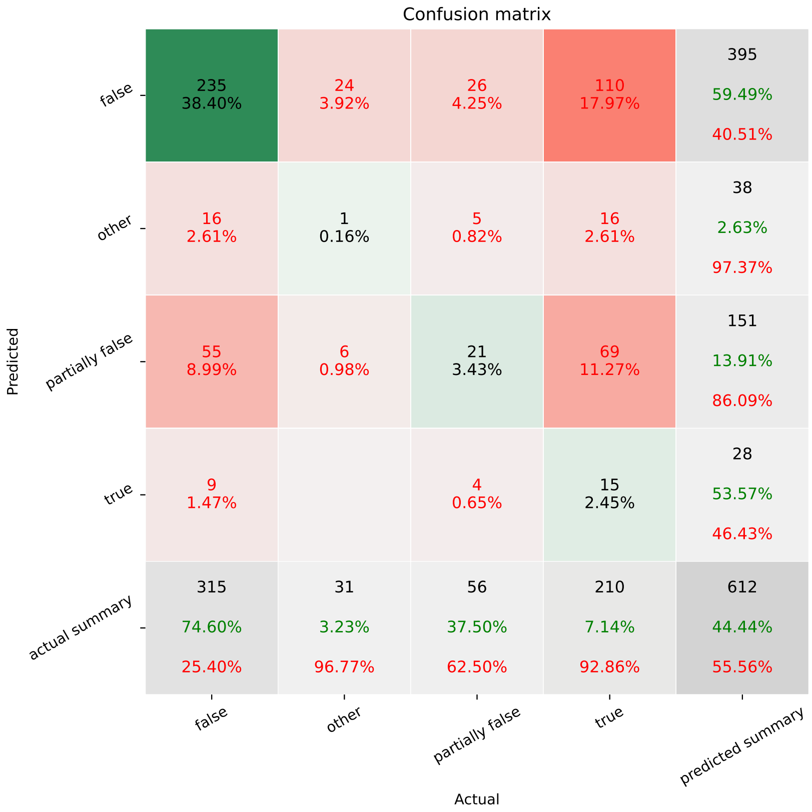
References

[1] P. Nakov, A. Barrón-Cedeño, G. Da San Martino, F. Alam, J. M. Struß, T. Mandl, R. Míguez, T. Caselli, M. Kutlu, W. Zaghouani, 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,

Table 4
Results for subtask 3A.

Rank	Team	Accuracy	F1-macro
1	iCompass	0.5473856209150327	0.33913726061970056
2	nlpiruned	0.5408496732026143	0.3324961059439111
3	awakened	0.5310457516339869	0.323094873671759
4	UNED	0.5441176470588235	0.3154167141734794
5	NLytics	0.5130718954248366	0.30760313138292816
6	SCUoL	0.5261437908496732	0.3046600458365164
7	hariharanrl	0.5359477124183006	0.2980435129438832
8	CIC	0.47549019607843135	0.28590932238045674
9	ur-iw-hnt	0.5326797385620915	0.2832669322709163
10	BUM	0.4722222222222222	0.27598221355575114
11	boby232	0.47549019607843135	0.2754227301661777
12	HBDCI	0.5081699346405228	0.273395238614303
13	DIU_SpeedOut	0.5212418300653595	0.2707056214947176
14	DIU_Carbine	0.4722222222222222	0.257884103161851
15	**CODE**	**0.4444444444444444**	**0.2549765772812493**
16	MNB	0.5065359477124183	0.25068001668752604
17	subMNB	0.5065359477124183	0.25068001668752604
18	fosil	0.4624183006535948	0.25051008810710806
19	Text_Minor	0.37745098039215685	0.23470704319654845
20	DLRG	0.5130718954248366	0.19871476054314866
21	DIU_Phoenix	0.2777777777777778	0.15930171516454703
22	AIT_FHSTP	0.19934640522875818	0.15489957496769197
23	DIU_SilentKillers	0.25980392156862747	0.1529300428217984
24	DIU_Fire71	0.27450980392156865	0.13281469514373465
25	AI Rational	0.09803921568627451	0.11650059103012848



	false	other	partially false	true	predicted summary
false	235 38.40%	24 3.92%	26 4.25%	110 17.97%	395 59.49% 40.51%
other	16 2.61%	1 0.16%	5 0.82%	16 2.61%	38 2.63% 97.37%
partially false	55 8.99%	6 0.98%	21 3.43%	69 11.27%	151 13.91% 86.09%
true	9 1.47%		4 0.65%	15 2.45%	28 53.57% 46.43%
actual summary	315 74.60% 25.40%	31 3.23% 96.77%	56 37.50% 62.50%	210 7.14% 92.86%	612 44.44% 55.56%

Figure 3: Confusion matrix for subtask 3A on the gold standard.

T. Caselli, M. Kutlu, W. Zaghouani, 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, 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] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding (2018). arXiv:1810.04805.

[5] G. K. Shahi, J. M. Struß, T. Mandl, J. Köhler, M. Wiegand, M. Siegel, Ct-fan-22 corpus: A multilingual dataset for fake news detection (version 3), https://doi.org/10.5281/zenodo.6508748, 2022.

[6] X. Zhou, R. Zafarani, Fake News: A Survey of Research, Detection Methods, and Opportunities, ACM Comput. Surv 1 (2018). arXiv:1812.00315.

[7] C. H. de Vreese, F. Esser, T. Aalberg, C. Reinemann, J. Stanyer, Populism as an Expres-

sion of Political Communication Content and Style: A New Perspective, International Journal of Press/Politics 23 (2018) 423–438. URL: http://journals.sagepub.com/doi/10.1177/1940161218790035. doi:10.1177/1940161218790035.

[8] U. Schade, F. Meißner, A. Pritzkau, S. Verschitz, Prebunking als Möglichkeit zur Resilienzsteigerung gegenüber Falschinformationen in Online-Medien, in: N. Zowislo-Grünewald, N. Wörmer (Eds.), Kommunikation, Resilienz und Sicherheit, Konrad-Adenauer-Stiftung, Berlin, 2021, pp. 134–155.

[9] J. Leskovec, K. Lang, Statistical properties of community structure in large social and information networks, Proceedings of the 17th international conference on World Wide Web. ACM (2008) 695–704. URL: http://dl.acm.org/citation.cfm?id=1367591.

[10] P. Badjatiya, S. Gupta, M. Gupta, V. Varma, Deep learning for hate speech detection in tweets, in: 26th International World Wide Web Conference 2017, WWW 2017 Companion, International World Wide Web Conferences Steering Committee, 2017, pp. 759–760. doi:10.1145/3041021.3054223. arXiv:1706.00188.

[11] L. Gao, R. Huang, Detecting online hate speech using context aware models, in: International Conference Recent Advances in Natural Language Processing, RANLP, volume 2017-Septe, Association for Computational Linguistics (ACL), 2017, pp. 260–266. doi:10.26615/978-954-452-049-6-036. arXiv:1710.07395.

[12] J. Pavlopoulos, P. Malakasiotis, I. Androutsopoulos, Deeper attention to abusive user content moderation, in: EMNLP 2017 - Conference on Empirical Methods in Natural Language Processing, Proceedings, Association for Computational Linguistics, Stroudsburg, PA, USA, 2017, pp. 1125–1135. URL: http://aclweb.org/anthology/D17-1117. doi:10.18653/v1/d17-1117.

[13] G. K. Pitsilis, H. Ramampiaro, H. Langseth, Effective hate-speech detection in Twitter data using recurrent neural networks, Applied Intelligence 48 (2018) 4730–4742. doi:10.1007/s10489-018-1242-y. arXiv:1801.04433.

[14] Z. Zhang, D. Robinson, J. Tepper, Detecting Hate Speech on Twitter Using a Convolution-GRU Based Deep Neural Network, in: Lecture Notes in Computer Science, volume 10843 LNCS, Springer, 2018, pp. 745–760. doi:10.1007/978-3-319-93417-4_48.

[15] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, Q. V. Le, XLNet: Generalized Autoregressive Pretraining for Language Understanding, Technical Report, 2019. arXiv:1906.08237.

[16] T. Mikolov, Q. V. Le, I. Sutskever, Exploiting Similarities among Languages for Machine Translation (2013). arXiv:1309.4168.

[17] J. Pennington, R. Socher, C. D. Manning, GloVe: Global vectors for word representation, in: EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference, 2014, pp. 1532–1543. doi:10.3115/v1/d14-1162.

[18] A. Joulin, E. Grave, P. Bojanowski, T. Mikolov, Bag of tricks for efficient text classification, in: 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017 - Proceedings of Conference, volume 2, 2017, pp. 427–431. URL: https://github.com/facebookresearch/fastText. doi:10.18653/v1/e17-2068. arXiv:1607.01759.

[19] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, Deep Contextualized Word Representations, Association for Computational Linguistics (ACL),

2018, pp. 2227–2237. doi:10.18653/v1/n18-1202. arXiv:1802.05365.

[20] M. Castelle, The Linguistic Ideologies of Deep Abusive Language Classification, 2019, pp. 160–170. doi:10.18653/v1/w18-5120.

[21] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, volume 2017-Decem, 2017, pp. 5999–6009. arXiv:1706.03762.

[22] 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.

[23] 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.

[24] 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.

[25] G. K. Shahi, J. M. Struß, T. Mandl, Task 3: Fake News Detection at CLEF-2021 CheckThat!, CLEF˜'2021, Zenodo, Bucharest, Romania (online), 2021. doi:10.5281/zenodo.4714517.

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

[27] 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.

[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.

[29] K. Shu, Fake News Detection Challenge KDD 2020, https://www.kaggle.com/competitions/fakenewskdd2020/overview/final-poster-and-presentation, 2020.

[30] Fakenews Classification Datasets, https://www.kaggle.com/datasets/liberoliber/onion-notonion-datasets, 2020.

[31] D. Miller, Leveraging BERT for extractive text summarization on lectures, CoRR abs/1906.04165 (2019). URL: http://arxiv.org/abs/1906.04165. arXiv:1906.04165.

[32] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer, arXiv 21 (2019) 1–67. arXiv:1910.10683.

[33] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin,

S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language models are few-shot learners, 2020. arXiv:2005.14165.

[34] S. Zhang, S. Roller, N. Goyal, M. Artetxe, M. Chen, S. Chen, C. Dewan, M. Diab, X. Li, X. V. Lin, et al., Opt: Open pre-trained transformer language models, arXiv preprint arXiv:2205.01068 (2022).

[35] M. Stede, Automatic argumentation mining and the role of stance and sentiment, Journal of Argumentation in Context 9 (2020) 19–41. URL: https://www.jbe-platform.com/content/journals/10.1075/jaic.00006.ste. doi:10.1075/jaic.00006.ste.

[36] M. Zampieri, S. Malmasi, P. Nakov, S. Rosenthal, N. Farra, R. Kumar, Predicting the type and target of offensive posts in social media, in: NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference, volume 1, Association for Computational Linguistics, Stroudsburg, PA, USA, 2019, pp. 1415–1420. URL: http://aclweb.org/anthology/N19-1144. doi:10.18653/v1/n19-1144. arXiv:1902.09666.

[37] S. Ruder, B. Plank, Strong Baselines for Neural Semi-supervised Learning under Domain Shift, ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers) 1 (2018) 1044–1054. arXiv:1804.09530.

[38] A. Ratner, S. H. Bach, H. Ehrenberg, J. Fries, S. Wu, C. Ré, Snorkel: rapid training data creation with weak supervision, in: VLDB Journal, volume 29, Springer, 2020, pp. 709–730. doi:10.1007/s00778-019-00552-1.