University of Regensburg at CheckThat! 2021: Exploring Text Summarization for Fake News Detection

Philipp Hartl, Udo Kruschwitz

University of Regensburg, Universitätsstraße 31, 93053 Regensburg, Germany

Abstract

We present our submission to the CLEF 2021 CheckThat! challenge. More specifically, we took part in Task 3a, *Multi-class fake news detection of news articles.* The conceptual idea of our work is that (a) transformer-based approaches represent a strong foundation for a broad range of NLP tasks including fake news detection, and that (b) compressing the original input documents into some form of automatically generated summary before classifying them is a promising approach. The official results indicate that this is indeed an interesting direction to explore. They also confirm that oversampling to address the class imbalance was effective to further improve the results. We also note that both abstractive and extractive summarization approaches score way better when we do not apply hypertuning of parameters suggesting that the small scale of the test collection leads to overfitting.

Keywords

Fake News Detection, Text Summarization, Abstractive / Extractive Summarization, CLEF, BERT

1. Introduction

Fake news, misinformation and disinformation is by no means a recent phenomenon, but instead has been around since classical antiquity when manipulated information was used to discredit political opponents or alter battle courses [1]. What did change though over time was the scale and extent of the problem, e.g. initially dissemination happened verbally, but the invention of the printing press marked a major milestone as easy access and distribution of information combined with increasing literacy enabled more people to consume and create information. The advent of social media with *the freedom to publish* marks the birth of a yet another era altogether [2]. The term *fake news* has been particularly prevalent in the mainstream media since the 2016 US election, when a large amount of intentionally false news was spread through social media during the campaign [3]. These platforms operate with a non-restrictive content policy by design and provide various ways for automation which eases the spread of mis- and disinformation. Combined with their enormous user bases (e.g. Facebook with 2.8 billion active users in December 2020 [1]) information is able to reach many people in a very short period

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

philipp1.hartl@stud.uni-regensburg.de (P. Hartl); udo.kruschwitz@ur.de (U. Kruschwitz)
0000-0002-5503-0341 (U. Kruschwitz)

[1] https://investor.fb.com/investor-news/press-release-details/2021/Facebook-Reports-Fourth-Quarter-and-Full-Year-2020-Results/default.aspx

of time. In an age of information pollution (irrelevant, redundant, unsolicited and low-value information [4]) it is therefore important to (semi-) automatically identify such claims and minimize their harm – in particular as humans appear to not be very skilled at identifying disinformation, with typical recognition rates only being slightly better than chance [5].

CheckThat! Lab [6] is an evaluation campaign which is part of the 2021 Cross-Language Evaluation Forum (CLEF) conference and contains three tasks related to fact-checking or fake news detection with two subtasks each. Our team participated in this year's Task 3a whose goal it is to create a system to identify fake news in a multi-label scenario. We built four models based on fine-tuned BERT [7], a highly-popular bidirectional transformer architecture, and abstractive respectively extractive summarization technologies [8, 9]. Our best submitted model (abstractive summarization) was ranked 8th among all 25 participating teams in the lab for this task. Post-hoc runs reveal though that the same runs but *without* hyperparameter tuning lead to substantially improved results (placing our best run 3rd in the ranked list). In this paper, we describe our participation in Task 3a at CLEF 2021 in detail.

2. Related Work

Traditionally, fake news detection is modelled as a classification problem but often with varying class numbers. While datasets like FakeNewsNet [10], MM-COVID [11] or ReCOVery [12] provide only two labels and hence see fake news detection as a binary classification, there exist also several datasets which got multiple labels such as FEVER [13], NELA-GT-2019 [14] or the dataset provided by the organizers of this task (see Section 3). Unfortunately, generating comprehensive datasets still takes a lot of work as the ground-truth labels often need to be assigned by, e.g., journalists or domain experts. Fake news detection systems typically adopt one of three general approaches or a combination of them. The most commonly used way is based on the news content which can be either linguistic, auditory (e.g., attached voice recordings) or visual (e.g., images or videos) [15]. This is based on the assumption that real and fantasy statements differ in content style and quality [16]. Therefore, it is possible to successfully differentiate claims solely on their content with either hand-engineered features [17] or deep learning methods [18]. However, approaches which only focus on the news content might miss valuable context information. Hence, feedback-based solutions target secondary information such as user engagements [19] and dissemination networks [20]. These approaches are often used in combination with content-based methods to increase performance [21]. While contextual information can be useful when available, it is often not or only partially available (as reflected by common benchmark collections for fake news detection [22, 13]). While both methods discussed above are limited to a snapshot of features present at the time of training, intervention-based methods try to dynamically interpret real-time dissemination data. These are arguably the least common approaches used at the moment because of their difficult way to evaluate [23]. When used though, they try to intervene the process of fake news spreading through e.g., injecting of true news into social networks [24] or user intervention [25, 26]. In this work we use a solely content-based approach simply because the dataset provided for this challenge has no additional context data. Additionally, gathering of some context data was explicitly forbidden as described in Section 4, so we decided to focus on a text-based solution.

3. Task Description

This year there have been a total of three CheckThat! tasks with two subtasks each [6]. We participated in *Task 3a: Multi-class fake news detection of news articles*, which is a part of Task 3: Fake News Detection. The goal is to *"given the text of a news article, determine whether the main claim made in the article is true, partially true, false, or other"*. The data used in this task is only available in English. As this task is designed as a four-class classification problem, the official evaluation metric introduced by the organizers is the F1-macro score. The F1-macro score is simply the mean of class-wise F1 scores:

$$F1 = 2 * \frac{precision * recall}{precision + recall} \tag{1}$$

$$F1_{macro} = \frac{1}{n} \sum_{i=0}^{N} F1_i \tag{2}$$

Up to five runs were permitted for each team. We submitted three competitive configurations and one baseline run to compare against our own approaches. Further details on all tasks can be found in the task overview [6].

4. Dataset

As this work is part of this year's CLEF CheckThatLab! [6] Task 3a, we used a modified version of the dataset by Shahi [27] provided by the organizers. This dataset also got four different classes to predict as defined in [28]. The distribution of each class in the provided training and test data can be seen in Table 1. The dataset was given in .csv format with four columns:

- public_id — unique identifier of the news article
- title — title/heading of the news article
- text — text content of the news article
- our rating — class of the news article (either *false*, *partially false*, *true* or *other*)

Table 1
Dataset statistics

Dataset	False	Partially False	True	Other
Training	486	235	153	76
Test	113	141	69	41

The training set contains 950 data points including the 50 sample data points released before both batches of data. The provided test set got 364 data points without labels. We received the ground-truth labels separately after the competition had finished (see Table 1). Each group had to submit a .csv file with their predictions separately on Codalab[2]. Additionally, through a data sharing agreement, it was forbidden to identify individuals and the original entries on the fact-checking websites. Therefore, we refrained from finding this information, although it would have been useful for classification purposes as demonstrated on a similar task [17].

[2] https://competitions.codalab.org/competitions/31238

5. Methodology

In the following section we provide an overview on how we prepared the data, the models we used as well as the training and evaluation process. Everything has been implemented in Python and is available on Github.[3]

5.1. Data preparation

We started our preprocessing with first converting all labels to numeric values. We used 0 for *true*, 1 for *false*, 2 for *partially false* and 3 for *other*. As seen in Table 1, the four classes are not equally distributed. We therefore applied *random oversampling* of all classes except the majority class using the imbalanced-learn package [29] with the aim to train a better classifier. Additionally, we generated *abstractive* and *extractive* summaries (we did this offline as in particular the generation of abstractive summaries was time-consuming). Before sending the text into our models we also tokenized and normalized the texts.

5.2. Model architecture

All models used are fine-tuned variants of Google's BERT [7] and use the *bert-base-uncased* implementation provided by Wolf et al. [30] in conjunction with a linear layer on top to predict the output. We have chosen BERT because it already has shown good performance in various text classification tasks [31] as well as in fake news detection [32]. Due to limited computational resources we could not use a more sophisticated BERT model like RoBERTa [33]. One of the main drawbacks of BERT-based models is the maximum sequence length each model is able to process which is at a maximum of 512 tokens (word pieces) for BERT. Unfortunately, fake news articles often are a longer than this value [34]. In the provided dataset the mean token length is 806 with at least 55% of texts exceeding the 512 token limit. As these values are calculated with *nltk* [35] and word pieces do not exactly match tokens, the real ratio is even higher (all other token values reported are calculated similarly). By default, BERT-based models simply truncate the text to the desired input length (or apply padding if it is too short). This leads to the loss of potentially important information in the input text. To circumvent this issue we propose three different solutions, all aimed at compressing the original text:

- Modified hierarchical transformer representation
- Extractive summarization
- Abstractive summarization

Hierarchical transformer representations have been introduced by Pappagari et al. [36]. In their work they suggest splitting the input text into smaller text segments with overlapping parts (stride) to represent the structure of the text. In our model we split the text into parts of 500 tokens with a stride length of 50. After getting the BERT embeddings for each text segment we then calculated the mean representation dimensionally and fed this into BERT. The output of BERT is then used to classify the input text. Mean embeddings have been successfully used before by Mulyar et al. [37]

[3] https://github.com/phHartl/CheckThatLab_2021

Another possible solution is to use automatic summarization to get a more condensed text representation. Deep learning models such as BART [8], XLNet [38] or ALBERT [39] perform exceptionally well on summarization tasks like SQuAD [40] or ELI5 [41] - even sometimes surpassing humans. These algorithms are able to reduce the text length by a significant amount if desired, which is ideal for the initial problem with BERT. In our work we use the extractive summarization technology implemented by [42]. Note that while this method is also based on BERT it has no maximum sequence length. To ensure a better summarization quality while keeping the running time reasonable we activated co-reference handling (better contextualization) and used distilBERT [43] as the underlying model. In contrast to [9] we are interested in long sequences and not only the first two sentences for classifying. After manually inspecting different configurations we settled with a summarization ratio of 0.40.

Apart from an *extractive* approach we also implemented an *abstractive* technique based on BART. This model is specifically well suited for text generation, outperforming similar ones on summarization tasks like SQuAD 1.1 [8]. The Huggingface transformers library [30] provides an easy way to use BART-models for sequence generation. Because of the repetitive nature of greedy and beam search [44, 45] we used *Top-K* [46] and *Top-p* sampling [47] for our summaries. The exact model we used is *sshleifer/distilbart-cnn-12-6*[4], which is a smaller BART model trained on a news summarization dataset by Hermann et al. [48]. In our final configuration we used the 100 (Top-K) most likely words and a probability (Top-p) of 95%. Like BERT, BART has a sequence limit of 1024 tokens. Therefore, if the input text was longer than 1000 tokens we used our first approach to ensure all parts of the text are taken into consideration when getting summarized. We also tried to get a summarization ratio of roughly 40% for better comparability to the extractive approach. However, as both approaches are not deterministic this cannot always be guaranteed (also, as noted, both approaches take quite a while to execute, so we saved the results in files once generated). Additionally, due to the late release of the dataset we could not try out many configurations but instead had to use suggested configurations.

Finally, the submitted models all use the hierarchical text representation (even when using text summaries). There is one model for each type of input text aka. no summary, extractive summary or abstractive summary. We also submitted a run without oversampling for better comparability.

5.3. Experimental setup

For training, we represented each input as *[CLS] + title + [SEP] + text*, where *text* is either the original text or one of the two summaries produced and [CLS] is a classification token and [SEP] is a token to indicate a separator between two sentences. For training, we use an 80/20 training/validation split and optimize hyperparameters based on the loss of the validation set. We used the same initial random state and split for all configurations to provide a better comparability. We used a batch size of 8, an initial learning rate of 5e-5, a weight decay of 0.01 with 500 warm-up steps and three training epochs with an AdamW [49] optimizer. Everything was trained on a single RTX 2080 Ti with 11 GB VRAM using the Huggingface library.

[4] https://huggingface.co/sshleifer/distilbart-cnn-12-6

6. Results

We report three sets of results – (a) official results for all four of our runs, and for comparison we also present results obtained on (b) the development set as well as (c) the test set without hyperparameter tuning (not submitted to the challenge).

First of all, in Table 2 we present the official results as returned to us by the shared task organizers. We marked the best-performing model for each metric in bold.[5] Recall, that hierarchical transformer representation is applied to the source text in *all* of our runs, i.e. the term "original texts" refers to text that has been created this way but *without* subsequently applying abstractive or extractive summarization, respectively.

Table 2
Official results

Model	Accuracy	Precision	Recall	F1-macro
BERT w/o oversampling	0.387	**0.636**	0.300	0.25570
BERT w/ original texts	0.432	0.409	**0.402**	0.40413
BERT w/ extractive summaries[6]	0.370	0.549	0.362	0.32986
BERT w/ abstractive summaries	**0.438**	0.476	0.385	**0.40415**

To contextualise the official results better (and also due to the fact that at this point we do not have official baseline results to compare against), we also report the results on the validation set (see Table 3). The configuration is the same as described in section 5.3 but without hyperparameter tuning (using a 80/20 split of the training data).

Table 3
Performance on validation/dev set without hyperparameter tuning

Model	Accuracy	Precision	Recall	F1-macro
BERT w/o fine-tuning	0.421	0.379	0.370	0.356
BERT w/o oversampling	**0.584**	**0.525**	0.371	0.329
BERT w/ original texts	0.511	0.378	0.379	0.369
BERT w/ extractive summaries	0.568	0.498	**0.463**	**0.459**
BERT w/ abstractive summaries	0.542	0.362	0.397	0.376

Table 4 also follows the same configuration, but has been calculated once the test set was available and does not use hyperparameter tuning either (using all training data and evaluating on the test data).

7. Discussion

First of all we observe that the non-fine-tuned model and the model which has been trained without oversampling the minority classes perform worst in all setups. This is in line with expectations.

[5] Because of the extremely close values in Table 2 we added additional fractional digits.

[6] The value for extractive summarization has been calculated with the official evaluation script afterwards, as there was a problem when uploading the file

Table 4
Performance on test set without hyperparameter tuning

Model	Accuracy	Precision	Recall	F1-macro
BERT w/o fine-tuning	0.251	0.328	0.315	0.251
BERT w/o oversampling	0.379	0.419	0.355	0.333
BERT w/ original texts	0.472	0.487	0.481	0.465
BERT w/ extractive summaries	**0.531**	**0.525**	**0.523**	**0.508**
BERT w/ abstractive summaries	0.489	0.509	0.450	0.459

It however gets more complicated when comparing the other models. The official runs suggest that *BERT w/ abstractive summaries* wins overall by a tiny bit, but is on par with *BERT w/ original texts* (i.e. the original articles hierarchically transformed but without applying summarization). Given that this makes it into 8th place of 25 submissions and the fact that abstractive summarization is becoming more and more competitive, we see this as a clear signal that our general conceptual idea is a promising one.

When taking a look at the official results for *BERT w/ extractive summaries* and *BERT w/o oversampling*, both models are still reasonably well-placed in the rankings. They would have ranked 16th and 18th respectively showing how well a vanilla BERT is pre-trained already.

Looking beyond the official results, we observe some wide variation of scores though. While *BERT w/ extractive summaries* performs better than other approaches when not using hyperparameter tuning (see Table 4), it scores way worse when hyperparameter tuning is in place (Table 2). In fact, *not* applying hyperparameter tuning would rank the system in 3rd position of the ranked list of 25 runs with an F1-macro of 0.508. This seems to be an indication of overfitting happening internally. The validation set in general seems to be not well suited to learn with, as all tuned models perform better when applying them to the test dataset directly (this is also the case, when the training set is exactly the same). All this raises some concerns about the size, robustness and generalisability of the test collection. This is by no means a novel finding, and some researchers go as far as to call the current (commonly applied) NLP evaluation approach to be broken [50]. We conclude that we will have to test our methodology on a wide range of additional collections to gain a better understanding of its strengths and weaknesses.

One last point to note, there seems to be only little difference in performance when using *BERT w/ original texts* or *BERT w/ abstractive summaries*. Interestingly, the respective models achieve very similar performance independently of the dataset and experimental setup used.

7.1. Limitiatons

Due to the nature of such challenges there was not much time to try different experimental setups. Especially abstractive summarization generation has a lot of different parameters to work with. Unfortunately, one iteration for those alone takes about half a day of computing time on our system. While we always tried to use the recommended configurations when possible, we could only use BERT with a maximum batch size of 8. It would have been interesting to see, whether batch sizes of 16 or greater make a significant difference in performance. Previous work on parameter tuning of BERT suggests this [51]. While BERT itself is a very sophisticated system, an approach using an even better system like RoBERTa [33] or XLNet [38] could outperform it.

This has already been proven in their respective papers on other NLP tasks. The substantial difference in performance between the official results (Table 2) and our reruns on the test set (Table 4) indicate that the chosen experimental setup might either not have been ideal for this task or the data sets were simply too small. While hyperparameter tuning is often useful, in this case we achieve better results without it. However, this could also be because of the validation/dev set we acquired. As seen in Table 3 all models perform worse here than on the actual test set. This indicates a bad seed for our validation set we optimized on. Also, a summarization ratio of 0.40 was picked quite arbitrarily which might or might not restrict the full potential of summarizations.

7.2. Future Work

In future it would certainly be interesting to explore more configurations and applications of automatic summarization. We believe summarization has the potential to enable better transferable knowledge. This could be useful for a variety of classification tasks as many models often only work in a certain domain. Therefore, it would be interesting to compare models trained on automatic summarizations and compare their performance in different domains working as a kind of "normalization" technique. We expect summarization of texts to limit overfitting in the future. With the results of Table 4 in mind, we hypothesize that there is a lot of room for improvement still available. We plan to apply our approaches on more datasets in the future and try to optimize the tuning further.

8. Conclusions

We presented an approach for fake news detection that is based on the powerful paradigm of transformer-based embeddings and utilises text summarization as the main text transformation step before classifying a document. The results suggest that this is indeed a worthwhile direction of work and in future work we plan to explore this further. We note that using oversampling has a strong positive effect on system performance. What we did also observe was that the performance obtained on different datasets and based on different models of hypertuning varied substantially. One way forward is to apply our framework to larger datasets to see how robust extractive and abstractive summarisation might be for the task at hand.

Acknowledgements

This work was supported by the project *COURAGE: A Social Media Companion Safeguarding and Educating Students* funded by the Volkswagen Foundation, grant number 95564.

References

[1] J. M. Burkhardt, History of Fake News, Library Technology Reports 53 (2017) 5–9.

[2] R. Baeza-Yates, B. Ribeiro-Neto (Eds.), Modern Information Retrieval, 2nd ed., Addison-Wesley, 2010.

[3] H. Allcott, M. Gentzkow, Social media and fake news in the 2016 election, Journal of economic perspectives 31 (2017) 211–36.

[4] L. Orman, Fighting Information Pollution with Decision Support Systems, Journal of Management Information Systems 1 (1984) 64–71. URL: https://doi.org/10.1080/07421222.1984.11517704. doi:10.1080/07421222.1984.11517704, publisher: Routledge _eprint: https://doi.org/10.1080/07421222.1984.11517704.

[5] V. L. Rubin, On deception and deception detection: Content analysis of computer-mediated stated beliefs, Proceedings of the American Society for Information Science and Technology 47 (2010) 1–10. Publisher: Wiley Online Library.

[6] P. Nakov, G. D. S. 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: D. Hiemstra, M. Moens, J. Mothe, R. Perego, M. Potthast, F. Sebastiani (Eds.), Advances in Information Retrieval - 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28 - April 1, 2021, Proceedings, Part II, volume 12657 of *Lecture Notes in Computer Science*, Springer, 2021, pp. 639–649. URL: https://doi.org/10.1007/978-3-030-72240-1_75. doi:10.1007/978-3-030-72240-1_75.

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

[8] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, L. Zettlemoyer, BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 7871–7880. URL: https://www.aclweb.org/anthology/2020.acl-main.703. doi:10.18653/v1/2020.acl-main.703.

[9] Q. Li, W. Zhou, Connecting the Dots Between Fact Verification and Fake News Detection, in: Proceedings of the 28th International Conference on Computational Linguistics, International Committee on Computational Linguistics, Barcelona, Spain (Online), 2020, pp. 1820–1825. URL: https://www.aclweb.org/anthology/2020.coling-main.165. doi:10.18653/v1/2020.coling-main.165.

[10] 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. URL: https://www.liebertpub.com/doi/abs/10.1089/big.2020.0062. doi:10.1089/big.2020.0062, publisher: Mary Ann Liebert, Inc., publishers.

[11] Y. Li, B. Jiang, K. Shu, H. Liu, MM-COVID: A Multilingual and Multidimensional Data Repository for CombatingCOVID-19 Fake News, arXiv:2011.04088 [cs] (2020). URL: http://arxiv.org/abs/2011.04088, arXiv: 2011.04088 version: 1.

[12] X. Zhou, A. Mulay, E. Ferrara, R. Zafarani, ReCOVery: A Multimodal Repository for COVID-19 News Credibility Research, in: Proceedings of the 29th ACM International Conference

on Information & Knowledge Management, Association for Computing Machinery, New York, NY, USA, 2020, pp. 3205–3212. URL: https://doi.org/10.1145/3340531.3412880.

[13] J. Thorne, A. Vlachos, C. Christodoulopoulos, A. Mittal, FEVER: a Large-scale Dataset for Fact Extraction and VERification, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 809–819. URL: https://www.aclweb.org/anthology/N18-1074. doi:10.18653/v1/N18-1074.

[14] M. Gruppi, B. D. Horne, S. Adalı, NELA-GT-2019: A Large Multi-Labelled News Dataset for The Study of Misinformation in News Articles, arXiv:2003.08444 [cs] (2020). URL: http://arxiv.org/abs/2003.08444, arXiv: 2003.08444.

[15] X. Zhou, J. Wu, R. Zafarani, SAFE: Similarity-Aware Multi-modal Fake News Detection, in: H. W. Lauw, R. C.-W. Wong, A. Ntoulas, E.-P. Lim, S.-K. Ng, S. J. Pan (Eds.), Advances in Knowledge Discovery and Data Mining, Lecture Notes in Computer Science, Springer International Publishing, Cham, 2020, pp. 354–367. doi:10.1007/978-3-030-47436-2_27.

[16] U. Undeutsch, Beurteilung der glaubhaftigkeit von aussagen, Handbuch der psychologie 11 (1967) 26–181.

[17] C. Yuan, Q. Ma, W. Zhou, J. Han, S. Hu, Early Detection of Fake News by Utilizing the Credibility of News, Publishers, and Users based on Weakly Supervised Learning, in: Proceedings of the 28th International Conference on Computational Linguistics, International Committee on Computational Linguistics, Barcelona, Spain (Online), 2020, pp. 5444–5454. URL: https://www.aclweb.org/anthology/2020.coling-main.475. doi:10.18653/v1/2020.coling-main.475.

[18] L. Cui, S. Wang, D. Lee, SAME: sentiment-aware multi-modal embedding for detecting fake news, in: Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM '19, Association for Computing Machinery, New York, NY, USA, 2019, pp. 41–48. URL: https://doi.org/10.1145/3341161.3342894. doi:10.1145/3341161.3342894.

[19] K. Shu, X. Zhou, S. Wang, R. Zafarani, H. Liu, The role of user profiles for fake news detection, in: Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM '19, Association for Computing Machinery, New York, NY, USA, 2019, pp. 436–439. URL: https://doi.org/10.1145/3341161.3342927. doi:10.1145/3341161.3342927.

[20] K. Shu, D. Mahudeswaran, S. Wang, H. Liu, Hierarchical Propagation Networks for Fake News Detection: Investigation and Exploitation, Proceedings of the International AAAI Conference on Web and Social Media 14 (2020) 626–637. URL: https://ojs.aaai.org/index.php/ICWSM/article/view/7329.

[21] K. Shu, L. Cui, S. Wang, D. Lee, H. Liu, dEFEND: Explainable Fake News Detection, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, ACM, Anchorage AK USA, 2019, pp. 395–405. URL: https://dl.acm.org/doi/10.1145/3292500.3330935. doi:10.1145/3292500.3330935.

[22] W. Y. Wang, “Liar, Liar Pants on Fire”: A New Benchmark Dataset for Fake News Detection, in: Proceedings of the 55th Annual Meeting of the Association for Computational

Linguistics (Volume 2: Short Papers), Association for Computational Linguistics, Vancouver, Canada, 2017, pp. 422–426. URL: https://www.aclweb.org/anthology/P17-2067. doi:10.18653/v1/P17-2067.

[23] K. Sharma, F. Qian, H. Jiang, N. Ruchansky, M. Zhang, Y. Liu, Combating Fake News: A Survey on Identification and Mitigation Techniques, arXiv:1901.06437 [cs, stat] (2019). URL: http://arxiv.org/abs/1901.06437, arXiv: 1901.06437.

[24] M. Farajtabar, J. Yang, X. Ye, H. Xu, R. Trivedi, E. Khalil, S. Li, L. Song, H. Zha, Fake News Mitigation via Point Process Based Intervention, arXiv:1703.07823 [cs] (2017). URL: http://arxiv.org/abs/1703.07823, arXiv: 1703.07823.

[25] Y. Papanastasiou, Fake News Propagation and Detection: A Sequential Model, Management Science 66 (2020) 1826–1846. URL: https://pubsonline.informs.org/doi/10.1287/mnsc.2019.3295. doi:10.1287/mnsc.2019.3295, publisher: INFORMS.

[26] J. Kim, B. Tabibian, A. Oh, B. Schölkopf, M. Gomez-Rodriguez, Leveraging the Crowd to Detect and Reduce the Spread of Fake News and Misinformation, in: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM '18, Association for Computing Machinery, New York, NY, USA, 2018, pp. 324–332. URL: https://doi.org/10.1145/3159652.3159734. doi:10.1145/3159652.3159734.

[27] G. K. Shahi, Amused: An annotation framework of multi-modal social media data, arXiv preprint arXiv:2010.00502 (2020).

[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] G. Lemaître, F. Nogueira, C. K. Aridas, Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning, The Journal of Machine Learning Research 18 (2017) 559–563. Publisher: JMLR. org.

[30] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, HuggingFace's Transformers: State-of-the-art natural language processing, arXiv preprint arXiv:1910.03771 (2019).

[31] M. Ostendorff, P. Bourgonje, M. Berger, J. Moreno-Schneider, G. Rehm, B. Gipp, Enriching BERT with Knowledge Graph Embeddings for Document Classification, arXiv:1909.08402 [cs] (2019). URL: http://arxiv.org/abs/1909.08402, arXiv: 1909.08402.

[32] J. Ding, Y. Hu, H. Chang, BERT-Based Mental Model, a Better Fake News Detector, in: Proceedings of the 2020 6th International Conference on Computing and Artificial Intelligence, Association for Computing Machinery, New York, NY, USA, 2020, pp. 396–400. URL: https://doi.org/10.1145/3404555.3404607.

[33] 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).

[34] W. Souma, I. Vodenska, H. Aoyama, Enhanced news sentiment analysis using deep learning methods, Journal of Computational Social Science 2 (2019) 33–46. Publisher: Springer.

[35] E. Loper, S. Bird, Nltk: The natural language toolkit, arXiv preprint cs/0205028 (2002).

[36] R. Pappagari, P. Żelasko, J. Villalba, Y. Carmiel, N. Dehak, Hierarchical Transformers for Long Document Classification, arXiv:1910.10781 [cs, stat] (2019). URL: http://arxiv.org/abs/1910.10781, arXiv: 1910.10781.

[37] A. Mulyar, E. Schumacher, M. Rouhizadeh, M. Dredze, Phenotyping of Clinical Notes with

Improved Document Classification Models Using Contextualized Neural Language Models, arXiv:1910.13664 [cs] (2020). URL: http://arxiv.org/abs/1910.13664, arXiv: 1910.13664.

[38] 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).

[39] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, R. Soricut, Albert: A lite bert for self-supervised learning of language representations, arXiv preprint arXiv:1909.11942 (2019).

[40] P. Rajpurkar, J. Zhang, K. Lopyrev, P. Liang, Squad: 100,000+ questions for machine comprehension of text, arXiv preprint arXiv:1606.05250 (2016).

[41] A. Fan, Y. Jernite, E. Perez, D. Grangier, J. Weston, M. Auli, Eli5: Long form question answering, arXiv preprint arXiv:1907.09190 (2019).

[42] D. Miller, Leveraging BERT for extractive text summarization on lectures, arXiv preprint arXiv:1906.04165 (2019).

[43] 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.

[44] A. K. Vijayakumar, M. Cogswell, R. R. Selvaraju, Q. Sun, S. Lee, D. Crandall, D. Batra, Diverse beam search: Decoding diverse solutions from neural sequence models, arXiv preprint arXiv:1610.02424 (2016).

[45] L. Shao, S. Gouws, D. Britz, A. Goldie, B. Strope, R. Kurzweil, Generating high-quality and informative conversation responses with sequence-to-sequence models, arXiv preprint arXiv:1701.03185 (2017).

[46] A. Fan, M. Lewis, Y. Dauphin, Hierarchical neural story generation, arXiv preprint arXiv:1805.04833 (2018).

[47] A. Holtzman, J. Buys, L. Du, M. Forbes, Y. Choi, The curious case of neural text degeneration, arXiv preprint arXiv:1904.09751 (2019).

[48] K. M. Hermann, T. Kočiskŷ, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, P. Blunsom, Teaching machines to read and comprehend, arXiv preprint arXiv:1506.03340 (2015).

[49] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, arXiv preprint arXiv:1711.05101 (2017).

[50] S. R. Bowman, G. E. Dahl, What will it take to fix benchmarking in natural language understanding?, in: K. Toutanova, A. Rumshisky, L. Zettlemoyer, D. Hakkani-Tür, I. Beltagy, S. Bethard, R. Cotterell, T. Chakraborty, Y. Zhou (Eds.), Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021, Association for Computational Linguistics, 2021, pp. 4843–4855. URL: https://www.aclweb.org/anthology/2021.naacl-main.385/.

[51] M. Guderlei, M. Aßenmacher, Evaluating Unsupervised Representation Learning for Detecting Stances of Fake News, in: Proceedings of the 28th International Conference on Computational Linguistics, International Committee on Computational Linguistics, Barcelona, Spain (Online), 2020, pp. 6339–6349. URL: https://www.aclweb.org/anthology/2020.coling-main.558. doi:10.18653/v1/2020.coling-main.558.