Reproducibility report on Explainable Automated Fact-Checking for Public Health Claims

Anonymous Author(s)
Affiliation
Address
email

Reproducibility Summary

Scope of Reproducibility

Our work consists of two major parts: (1) Reproducing results from Kotonya & Toni(Authors) (3) (2) Performing experiments to improve test accuracy and other metrics for veracity prediction. We did not use BioBERT(20) model deliberately for veracity prediction as it did not perform well on the defined metrics, as observed in the original paper(3). Authors were doubtful of how good the rouge metric is in conveying the quality of explanations, so they used human evaluation to evaluate the explanations generated. We stuck to rouge score for evaluating the explanations generated.

Methodology

Authors did not publish the code for fine-tuning BERT(10) and SciBERT(7) models for veracity prediction. For explanation generation the authors use a BERT based model which was not made public, so we chose the BART model pre-trained on CNN-DailyMail dataset. We have written a functional and modular code.[1] which is easy to reproduce and comprehend.

Results

The accuracy for veracity prediction using BERT base model (top 5 sentences) was 3% lower than that published by the authors. The accuracy for veracity prediction using SciBERT (top 5 sentences) was 4.73% lower than that published by the authors. SciBERT performed well on all the test metrics for veracity prediction. While the accuracy was close, the macro F1, precision and recall were inconsistent with the authors' claim. For explanation generation, the automated evaluation metric was rouge(21). In case of R1 and RL, we got f1 measure, which was around 30% more than what was mentioned in the paper(3). Improvements were also observed in R2 rouge score. 4.1.4. We also checked some of the explanations that were generated, and the results were up to the mark with gold standard explanations.

What was easy

It was easy to implement the code for veracity prediction using two different BERT models. The model used for summarization was available in the Hugging Face library, pretrained on the same dataset as the authors. Without much effort, we were able to fine-tune the model on our dataset.

What was difficult

The implementation code was not available in author's GitHub repository[2]. We had to implement code ourselves. It was difficult to increase the accuracy of the models to get close to that published by the authors.

Communication with original authors

We tried contacting the authors many times, but unfortunately could not make any contact.

[1] https://github.com/saswat01/Reproduce-Health_Fact_Checking

[2] https://github.com/neemakot/Health-Fact-Checking

1 Introduction

A great amount of progress has been made in the area of automated fact-checking. This includes more accurate machine learning models for veracity prediction and datasets of both naturally occurring (Wang, 2017; Augenstein et al., 2019; Hanselowski et al., 2019) and human-crafted (Thorne et al., 2018) fact-checking claims, against which the models can be evaluated. We introduce a framework for generating explanations and veracity prediction specific to public health fact-checking. We show that gains can be made through the use of in domain data. The second shortcoming we look to address is the paucity of explainable models for fact-checking (of any kind). Explanations have a particularly important roles to play in the task of automated fact checking, especially in health-related claims where domain specific knowledge is required to understand the context. Explainable models can also aid the end users' understanding as they further elucidate claims and their context[3].

This work intends to perform reproducibility, experiments to improve score on evaluation metrics for veracity prediction and perform an ablation study(remove BioBERT for veracity prediction and human evaluation of generated explanations) and validate the metrics to evaluate the experiments. The work also contributes a pipeline of veracity prediction and explanation generation, using which we can get veracity prediction of a claim and explanation verifying the prediction.

2 Scope of reproducibility

As mentioned by the authors, (3) the veracity predictions made on the claims and evidence sentences (top 5) gave the best accuracy when SciBERT was used and BERT(base-uncased) model gave the best precision score. The SciBERT model succeeded on all the test metrics for veracity prediction. But the BERT model did not surpass SciBERT in terms of precision score, which was contrary to what was observed by the authors. We also took a data centric approach and observed the performance of the same transformers on dataset alterations for veracity prediction.

For explanation generation, we used the top 5 sentences returned by SBERT in veracity prediction stage. The BART based summarization model pretrained on CNN/DailyMail dataset, yielded better results than BERT based extractive abstractive summarization model pretrained on the same dataset as mentioned in the paper (3). Our central aim was to reproduce these results as close as possible to the authors. The results of our experimental findings are shown in Table 4.1.1, 4.1.2 and 4.1.3.

3 Methodology

We have provided the entire code to fine-tune the BERT, SciBERT and DistillBart transformers for veracity prediction and explanation generation. The processed dataset has been put up in the GitHub repository (1), which can be used straightaway for fine-tuning the transformer models on downstream tasks. The modularity of the code makes it comfortable for experimentation. For, e.g., if an individual wants to increase/decrease hyperparameters like batch size, epochs, learning rate etc. they can easily do that. All the instructions have been specified in the GitHub repository (1) regarding the usage of the repository and experimentation. We have made use of PyTorch Lightning, which makes training swifter. Along with that, we train all the models using early stopping with patience of 2 or 3. While training the models, we save the checkpoints when the validation loss is least. It helps to track and save the model with the best weights.

The fine-tuning was performed on Tesla T4 15.84 Gigabyte GPUs provided on the Google Colaboratory platform. The recommended batch size provided by the authors was 16 only the GPUs could support a batch size of 13. We used batch size 13 for fine-tuning both the transformer models for veracity prediction and a batch size of 8 for fine-tuning DistillBart for explanation generation. Computational time and other details have been provided in Section 3.6. We have also provided a convenient test script which can predict label, select evidences from main text and generate an explanation for the claim text.

3.1 Model descriptions

For veracity prediction, we make use of pretrained BERT(base-uncased) model and SciBERT(scibert-scivocab-uncased) model from Hugging Face library. The tokenizers for BERT(base-uncased) and SciBERT(scibert-scivocab-uncased) were also used from the Hugging Face library. For explanation generation, we used DistillBart model and tokenizer pretrained on CNN-DailyMail dataset from the Hugging Face library.

[3] Excerpts taken from paper (3)

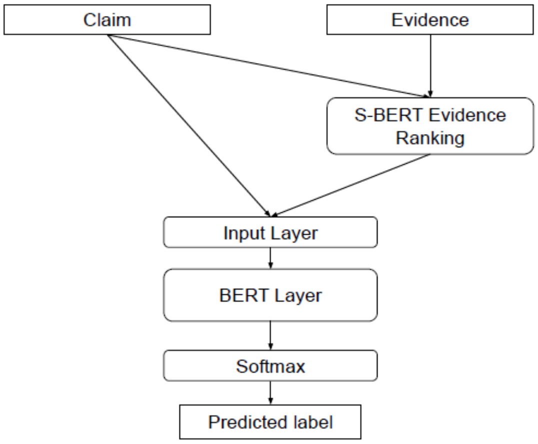


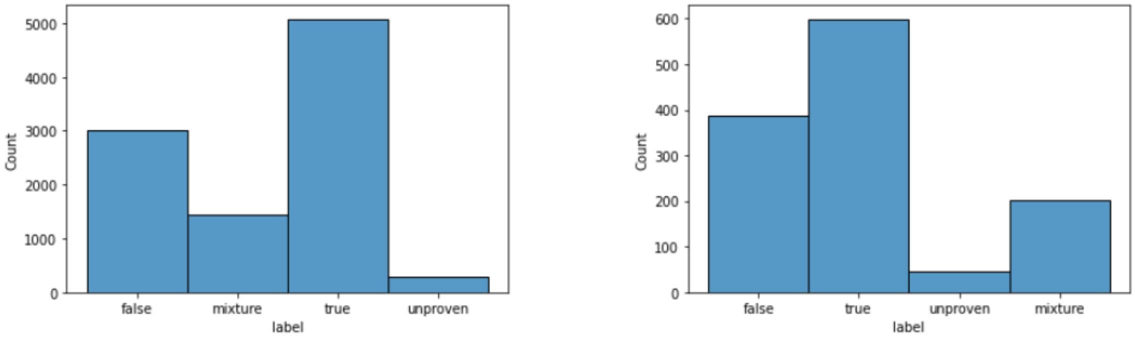
Figure 1: Veracity Prediction model Architecture

1. The pretrained BERT(base-uncased) model has 12-layer, 768-hidden, 12-heads, 110M parameters. It was pretrained on lower-cased English text.
2. The pretrained SciBERT model is a BERT model pretrained on papers taken from Semantic Scholar of Corpus size 1.14M papers and 3.1B tokens.
3. The pretrained DistillBart 12-6 model is pretrained on CNN- DailyMail dataset, which has 300k unique news articles written by journalists at CNN and the Daily Mail. It has 306M parameters.
4. A smaller DistillBart model pretrained on the same CNN-DailyMail dataset. It has 230M parameters.

We have provided the above-mentioned fine-tuned models in the repository (1).

3.2 Datasets

The Pubhealth dataset constructed by the authors (3) contains 11,832 claims for fact-checking. The claims were related to several topics like biomedical research, government healthcare policies and other health related stories. The dataset is divided into three splits train, dev and test dataset. Main features in the dataset was claim ID, claim sentence, main text containing article text, explanation, and label. The distribution of labels in train, dev and test dataset is displayed in Figure 2. The dataset had some NA values for some features. We dropped the rows containing NA values. After dropping the rows we had 9806 observations in train dataset, 1235 observations in test dataset and 1217 observations in validation dataset.



(a) Label distribution for train dataset

(b) Label distribution for test dataset

Figure 2: Distribution of labels in dataset

We also created a derived dataset using authors code6.3 for preprocessing and then applying SBERT to the main text to get top k sentences for veracity prediction according to their cosine similarity with respect to the contextualized representation of the claim sentence.

The dataset in raw and processed form are available in our GitHub(1) repository.

3.3 Hyperparameters

Details regarding the hyperparameters like batch size, number of epochs, learning rate are mentioned in the original research paper by the authors. Although, it helped us to reproduce the paper, information about other hyperparameters (for e.g., token length) and detailed methodology for fine-tuning task for veracity prediction was not offered. We used the same learning rate as mentioned by the authors, i.e., 1e-6. As was mentioned earlier, the GPU provided on the Google Colaboratory platform could not support a batch size of 16 (recommended by the authors) we used batch sizes of 8, 10, 12 and 13 for experimentation. Out of which, a batch size of 13 established the best results; a batch size of 12 also showed synonymous results. We altered epochs from 4 to 7 and got 5 as the most appropriate number of epochs. Hence, the number of epochs for fine-tuning the models was altered from 4 (recommended by the authors) to 5 as it gave best results in our case. Encode plus tokenizer with maximum length of 512 was used for fine-tuning both the transformers. We optimized our model using Cross-Entropy Loss (recommended by authors) for veracity prediction.

For all experiments related to veracity prediction, hyperparameter trials were done for each possibility of batch sizes {10, 12, 13} and number of epochs {4, 5, 7}.

In the case of explanation generation, authors didn't give any details for hyperparameter tuning. We enforced manual search with a couple of combination of hyperparameter values. We used 5e-5 as the learning rate, batch size of 8, number of epochs was 3. We used Adam W optimizer with maximum input length of 512 and the maximum output length was 128.

3.4 Experimental setup and code

For veracity prediction models were evaluated on test dataset using macro F1, precision, recall, and accuracy metrics. Linear scheduler with warm up was used to train the language models to decrease chances of early over-fitting or skewness. For cleaning the top 5 evidence sentences, we used regex library provided by the Python language. Precise instructions have been provided in the repository(1) to assist you to train the models and test them, along with discerning the test metrics.

For explanation generation, the performance on the test dataset was assessed using rouge score metrics. The training script is easy to run aiding arguments, with option to save the model which can be used later to evaluate on the test data. We have provided a test script in the GitHub repository, which lets you perform veracity prediction and generate explanation for the claim sentence simultaneously.

3.5 Extended Experiments

Apart from reproducing the paper, experiments were conducted, to answer the following questions:

- Can managing class imbalance in the dataset lead to better performance of models on evaluation metrics for veracity prediction ?
- Can text cleaning improve model performance for veracity prediction ?
- Can any other hyperparameter improve model performance for veracity prediction ?

To handle class imbalance in the dataset, synonym matching (replacing maximum 15 words in the top 5 evidence sentences using synonyms) was used as the augmentation technique. Most observations lied under the "True" label, so we made the "Mixture" and "Unproven" label observations equivalent to the "False" label, which had observations next to the "True" label. We conducted experiments on the augmented dataset using the BERT model, 4.1.2 discusses the results. The SciBERT model did not give satisfactory results on the augmented data, so we did not perform extensive experimentation utilizing it.

Text cleaning was performed using the regex library provided by the Python language. We removed redundant text, i.e., square brackets, links, punctuation, and words containing numbers from the top 5 evidence sentences. The SciBERT model fine-tuned on the clean top 5 evidence sentences performed flawlessly on all the evaluation metrics 4.1.3.

It was observed that the average token length was 125 and a maximum token length of 512 was observed. There were a handful of tokens whose token length was 512. We reduced the token length from 512 to 350 to fine-tune BERT and

SciBERT on the clean data. We chose 350 as the trusted token length, as it was capable of representing the token length of 95% of the sentences. It gave the best results and the results are discussed in Section 4. A 2% gain in accuracy was discovered when the SciBERT model was fine-tuned on clean top 5 evidence sentences along with a token length of 350.

3.6 Computational requirements

As the experiments are done on an easily accessible environment, there are no specific requirements one needs to reproduce and implement our work. You need a laptop/PC and an internet connection to perform everything that we have published in the report and in the GitHub repository (1).

The computational time for various models and other relevant details have been provided in the tables below.

Model	Train time(in sec.)	Train time(in hours)
BERT(epoch 4)	3420 sec	0.95 hours
BERT(epoch 5)	5400 sec	1.5 hours
BERT(epoch 7)	7200 sec	2 hours
SciBERT(epoch 4)	3600 sec	1 hours
SciBERT(epoch 5)	5940 sec	1.65 hours

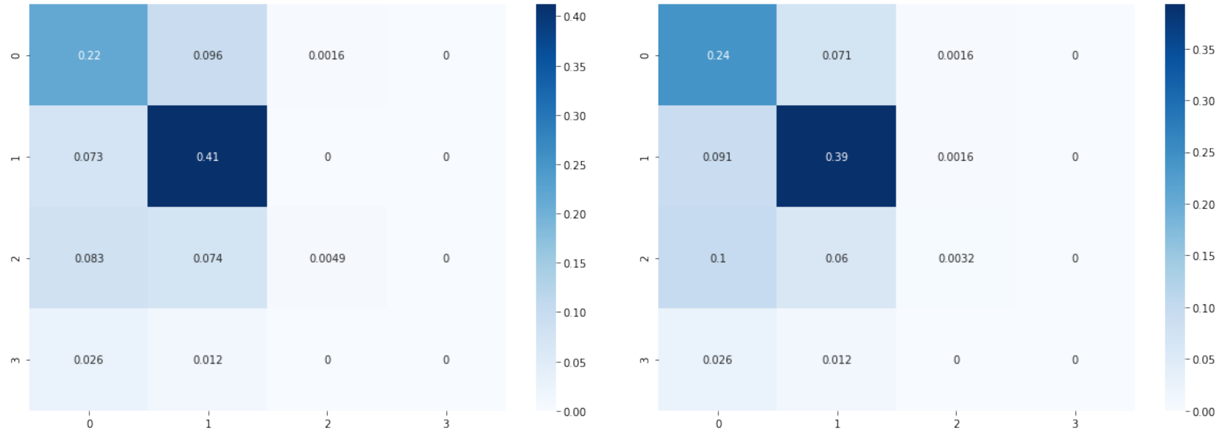
Table 1. Training computational time for veracity prediction

Model	Test time(in sec.)	Test time(in minutes)
distilbart-cnn-12-6 (epoch 3)	6300 sec	1.75 hours
distilbart-cnn-6-6 (epoch 3)	4500 sec	1.25 hours

Table 2. Training computational time for summarization

4 Results

For calculation of all the metrics for veracity prediction, Scikit-learn library was used. The SciBERT model gives the best accuracy, F1 score, precision, and recall on the Pubhealth dataset for veracity prediction. It supports the original claim of the authors except that their BERT model gave better precision than SciBERT which was not observed in our experiments. Also, from all the experiments we conducted, SciBERT model gave the best results when the top 5 evidence sentences was cleaned and the token length was shrunk. BERT model gave the best results when it was trained using the best hyperparameters, as discussed 3.3. Also, it was observed that BERT results were not very different when it was fine-tuned on clean top 5 evidence sentences. Distillbart based model gave better results than ExplanerFC-Expert model used by authors. The rogue scores are given in the table 4.1.4 along with authors' best model.



(a) Bert

(b) Scibert

Figure 3: (a)BERT and (b)SciBERT confusion matrix

4.1 Results reproducing original paper

It can be comprehended that SciBERT surpasses BERT on every evaluation metric in section 4.1.1.

4.1.1 Veracity prediction best results

Both models were fine-tuned using the best hyperparameters, except the token length. It can be sighted that accuracy metrics is proximate to the original assertions of the authors but the precision, accuracy and F1 score is contrary from what was originally verified by the authors. SciBERT performs unexcelled on all the metrics for veracity prediction. It partially supports the authors' assertions, as the precision of the BERT model is not better than the SciBERT model.

Model	Pr.	Rc.	F1	Acc.
BERT(top 5)	0.35	0.39	0.35	63%
SciBERT(top 5)	**0.44**	**0.41**	**0.37**	**65%**

Table 3. BERT fine-tuned using token length 512, SciBERT fine-tuned using token length 350

4.1.2 Augmentation Result using BERT(top 5) with batch size 12

We experiment the phenomenon of epoch size on fine-tuning the BERT model on the dataset for veracity prediction. Text augmentation improved the precision, recall and F1 score of the BERT model but was not improving the accuracy of the model. Data augmentation also facilitated the BERT model to categorize the labels more accurately, particularly the "Mixture" and "Unproven" labels.

Epochs	Pr.	Rc.	F1	Acc.
4	0.41	0.39	0.37	57%
5	0.42	0.35	0.34	59%
7	0.45	0.37	0.35	59%

Table 4. BERT metrics when trained on augmented dataset using synonym replacement, token length 512

4.1.3 Text Cleaning Result on language models for veracity prediction

Model	Pr.	Rc.	F1	Acc.
BERT	0.31	0.38	0.34	62%
SciBERT	0.44	0.41	0.37	65%

Table 5. Language models trained using the best hyperparameters with token length 350

4.1.4 Results for explanation generation with batch size 8

The metric used to evaluate the explanation quality was rouge.

Model	Metric	precision	recall	F1
distilbart-cnn-12-6	R1	0.472	0.451	0.461
	R2	0.181	0.173	0.177
	RL	0.409	0.392	0.4
distilbart-cnn-6-6	R1	0.459	0.438	0.447
	R2	0.165	0.158	0.161
	RL	0.395	0.377	0.385
ExplanerFC-Expert	R1	-	-	0.323
	R2	-	-	0.135
	RL	-	-	0.27

Table 6. Explanation generation results

5 Ablation Study

As discussed in the original research paper (3), BioBERT v1.1 and BioBERT v1.0 did not show any significant performance on the metrics mentioned for veracity prediction. We excluded the BioBERT transformer model for the reproducibility of veracity prediction.

For evaluation of explanations, authors used two methods. The second one was human evaluation, in which authors asked humans to assess the quality of the gold and generated explanations. We could not do human evaluation of explanations. Authors were skeptical about how good rouge score comprehends the usefulness or the quality of the explanations, so they also performed human evaluations of generated explanations. Authors calculated coherence to assess the quality of the explanations. We stuck to rouge metric for our exclusive evaluation criteria for explanation generation.

6 Discussion

The experimental results discussed above supports the overall claims by authors. For veracity prediction, we could reproduce the results claimed by authors for SciBERT model by using clean top 5 evidence sentences and token length of 350. These details were not provided in the paper and were revealed using different experimental setups. Unfortunately, we could not connect to the authors to substantiate our methodology, but the experimental results convinced us to conclude this approach suitable. We could have also experimented with different types of augmentation techniques to discover how the models would have performed. Also, we could have made the maximum token length closer to the average token length to record the empirical observations of model performance. It may have been possible that experimenting on these varied scenarios would have concurred in metrics closer to that published by the authors.

For the explanation generation task, the authors used two different types of evaluation method. One was automatic evaluation using rouge metric. Rouge metric is considered to be the best metric when it comes to summarization tasks. As discussed in the ablation study part, authors did use human evaluation. We could have increased the k value above 5 to check if that generates better results. We used 128 as the max output token, as the average length of gold standard explanations were more or less the same.

6.1 What was easy

As the authors provided the script in their GitHub repository6.3 to extract top 5 evidence sentences from main text, it helped us a lot to kick-start the implementation of transformer models for veracity prediction. Also, the authors provided a clear architecture[1] for veracity prediction that helped us to understand the flow of the whole process for veracity prediction.

6.2 What was difficult

The fine-tuning procedure was not explained in detail by the authors, due to which it took significant amount of time and experiments to search the most suitable hyperparameters for fine-tuning the models for veracity prediction. The instructions provided by the authors about the abstractive-extractive model which they used were difficult to follow.

6.3 Communication with original authors

We tried reaching out to the authors by email. Unfortunately, we could not connect to them. We apprehend they maybe busy. We have also sent them this report for verification and expect their response.

References

[1] `https://github.com/saswat01/Reproduce-Health_Fact_Checking`

[2] `https://github.com/neemakot/Health-Fact-Checking`

[3] Kotonya, Neema, and Francesca Toni. Explainable automated fact-checking: A survey. arXiv preprint arXiv:2011.03870 (2020)

[4] Naser Ahmadi, Joohyung Lee, Paolo Papotti, and Mohammed Saeed. 2019. Explainable fact checking with probabilistic answer set programming. arXiv preprint arXiv:1906.09198, abs/1906.09198.

[5] Pepa Atanasova, Jakob Grue Simonsen, Christina Lioma, and Isabelle Augenstein. 2020. Generating fact checking explanations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7352–7364, Online. Association for Computational Linguistics.

[6] Isabelle Augenstein, Christina Lioma, Dongsheng Wang, Lucas Chaves Lima, Casper Hansen, Christian Hansen, and Jakob Grue Simonsen. 2019. MultiFC: A real-world multi-domain dataset for evidence-based fact checking of

claims. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4685–4697, Hong Kong, China. Association for Computational Linguistics.

[7] Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages* 3615– 3620, Hong Kong, China. Association for Computational Linguistics.

[8] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

[9] Jeanne Sternlicht Chall and Edgar Dale. 1995. Readability revisited: The new Dale-Chall readability formula. *Brookline Books.*

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

[11] William Ferreira and Andreas Vlachos. 2016. Emergent: a novel data-set for stance classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1163–1168, San Diego, California. Association for Computational Linguistics.

[12] Mohamed H Gad-Elrab, Daria Stepanova, Jacopo Urbani, and Gerhard Weikum. 2019. Exfakt: a framework for explaining facts over knowledge graphs and text. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 87–95. ACM.

[13] Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. AllenNLP: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1– 6, Melbourne, Australia. Association for Computational Linguistics.

[14] Peter Grabitz, Yuri Lazebnik, Joshua Nicholson, and Sean Rife. 2017.Science with no fiction: measuring the veracity of scientific reports by citation analysis. BioRxiv, page 172940.

[15] Lucas Graves. 2018. Boundaries not drawn: Mapping the institutional roots of the global fact-checking movement. *Journalism Studies*, 19(5):613–631.

[16] Andreas Hanselowski, Christian Stab, Claudia Schulz, Zile Li, and Iryna Gurevych. 2019. A richly annotated corpus for different tasks in automated factchecking. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL), pages 493–503, Hong Kong, China. Association for Computational Linguistics.*

[17] Naeemul Hassan, Fatma Arslan, Chengkai Li, and Mark Tremayne. 2017. Toward automated factchecking: Detecting check-worthy factual claims by claimbuster. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 1803–1812.*

[18] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in neural information processing systems, pages 1693–1701.*

[19] J Peter Kincaid, Robert P Fishburne Jr, Richard L Rogers, and Brad S Chissom. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. *Technical report, Naval Technical Training Command Millington TN Research Branch.*

[20] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

[21] Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

[22] Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing* (EMNLP-IJCNLP), pages 3730–3740, Hong Kong, China. Association for Computational Linguistics.

[23] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

[24] Yixin Nie, Haonan Chen, and Mohit Bansal. 2019. Combining fact extraction and verification with neural semantic matching networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6859–6866.

[25] Ankur Parikh, Oscar Tackstr ¨ om, Dipanjan Das, and ¨ Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2249–2255, Austin, Texas. Association for Computational Linguistics.

[26] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Volume 1 (Long Papers), pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

[27] Kashyap Popat, Subhabrata Mukherjee, Andrew Yates, and Gerhard Weikum. 2018. DeClarE: Debunking fake news and false claims using evidence-aware deep learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 22–32, Brussels, Belgium. Association for Computational Linguistics.

[28] Justus J Randolph. 2005. Free-marginal multirater kappa (multirater k [free]): An alternative to fleiss' fixed-marginal multirater kappa. Online submission. Nils Reimers and Iryna Gurevych. 2019. SentenceBERT: Sentence embeddings using Siamese BERTnetworks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing* (EMNLP-IJCNLP), pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

[29] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

[30] Kai Shu, Limeng Cui, Suhang Wang, Dongwon Lee, and Huan Liu. 2019a. defend: Explainable fake news detection. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery Data Mining*, KDD '19, pages 395– 405, New York, NY, USA. ACM.

[31] Kai Shu, Deepak Mahudeswaran, Suhang Wang, Dongwon Lee, and Huan Liu. 2019b. Fakenewsnet: A data repository with news content, social context and spatialtemporal information for studying fake news on social media.

[32] Kacper Sokol and Peter Flach. 2019. Desiderata for interpretability: Explaining decision tree predictions with counterfactuals. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 10035–10036.

[33] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. 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), pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

[34] Bernard Turnock. 2012. Public Health: What It Is and How It Works. Jones Bartlett Publishers, Gaithersburg, Md.

[35] Andreas Vlachos and Sebastian Riedel. 2014. Fact checking: Task definition and dataset construction. In *Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science*, pages 18–22, Baltimore, MD, USA. Association for Computational Linguistics.

[36] William Yang Wang. 2017. "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), pages 422–426, Vancouver, Canada. Association for Computational Linguistics.

[37] Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Volume 1 (Long Papers), pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

[38] Wanjun Zhong, Jingjing Xu, Duyu Tang, Zenan Xu, Nan Duan, Ming Zhou, Jiahai Wang, and Jian Yin. 2019. Reasoning over semantic-level graph for fact checking.

[39] Dimitrina Zlatkova, Preslav Nakov, and Ivan Koychev. 2019. Fact-checking meets fauxtography: Verifying claims about images. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing* (EMNLP-IJCNLP), pages 2099–2108, Hong Kong, China. Association for Computational Linguistics.