IUCL at PAN 2024: Using Data Augmentation for Conspiracy Theory Detection

Notebook for the Oppositional Thinking Analysis Lab at CLEF 2024

Shrirang Mhalgi, Srikar Kashyap Pulipaka and Sandra Kübler

Indiana University, Bloomington, IN, USA

Abstract

Team IUCL used a fine-tuned DeBERTa [1] with a sequence classification head as its basis. DeBERTa was finetuned on an augmented dataset comprising the PAN24 [2] training set and the LOCO corpus [3]. LOCO is a corpus consisting of conspiracy and mainstream texts for a range of conspiracy topics, including COVID related narratives. By adding all LOCO texts, the training set increased by five times to around 20,000 texts. The team utilized a balanced subset of 400 samples from the original dataset as the development set. Data augmentation led to a significant improvement in model convergence, resulting in DeBERTa's performance equaling that of a Llama-2-7B model on the original dataset. The augmented DeBERTA model was evaluated on the PAN24 test set and achieved an MCC score of 0.8388, achieving the best result out of 83 teams.

Keywords

Conspiracy Theory, Data Augmentation, Language Models, Ensembles

1. Introduction

Team IUCL participated in the shared task on distinguishing between conspiracy and critical texts [2] at PAN-CLEF [4]. The classification of critical and conspiracy texts presents a multifaceted challenge since conspiracy theories are generally conveyed by insinuation rather than spelling them out. Thus, this task requires a complex understanding of language and context, as well as sophisticated computational methods capable of discerning subtle patterns and rhetorical strategies.

In our work, we explore different types of classifier, including statistical models, smaller and large language models[1]. We investigate the classification of critical and conspiracy texts by leveraging data augmentation of the training data. We use additional data from the LOCO corpus [3], an automatically collected corpus of conspiracy websites and mainstream documents on the same topics. We also investigate different strategies for data pre-processing and feature extraction, along with the different machine learners and ensembles. We focus on subtask 1, binary classification, and on the English data only. Our best system ranked 1st out of the 83 submissions for this subtask.

The remainder of this paper is structured as follows: Section 2 gives a brief overview of related work, section 3 provides an overview of the data we used. Section 4 describes our methodology, and section 5 discusses our results. We conclude in section 6.

2. Related Work

The surge of conspiracy theories spread over social media and elsewhere has resulted in a renewed interest among scholars to understand how conspiracy theories spread [5], who is susceptible to believing conspiracy theories [6], to determine characteristics of conspiracy language [7], and to determine how to detect conspiracy theory content automatically [8].

CLEF 2024: Conference and Labs of the Evaluation Forum, September 09–12, 2024, Grenoble, France

srmhalgi@iu.edu (S. Mhalgi); spulipa@iu.edu (S. K. Pulipaka); skuebler@iu.edu (S. Kübler)

https://cl.indiana.edu/~skuebler/ (S. Kübler)

0000-0003-0885-5436 (S. Kübler)

[1] For convenience, we group all language models together and call them LMs.

Ethics professor fired for having ethics . ‚Äú My school implores me to be an authority on ethics and I ’m here to tell you it ’s ethically wrong to coerce someone to take a vaccine . ‚Äù Fight back . AFLDS . org / legal

Figure 1: Sample text from the training data.

In terms of computational approaches to conspiracy theory detection, Peskine et al. [9] used finetuned large language models, to detect COVID-19 related conspiracy theories. Fort et al. [8] developed methodology to robustly detect conspiracy theories out of domain. They argue that conspiracy theories are not monolithic, and adherents believe in individual sets of factoids, generally of a range of conspiracy theories. This necessitates a domain independent methodology. They show that by bleaching words typical for individual conspiracy theories, an SVM becomes more robust out of domain. Reiter-Haas et al. [10] investigate the use of semantic analyses in the form of Augmented Feature Representation (AMR) graphs to analyze the framing used in health related conspiracy theories. They find that health-related narratives in conspiracy media are mostly framed as beliefs while mainstream media generally use terms of science.

3. Data and Data Augmentation

3.1. Data

Since we are participating in subtask 1, the binary classification, and on the English data, we describe only that dataset here. The datasets provided by the shared task organizers were collected from the Telegram platform, specifically addressing discussions surrounding the COVID-19 pandemic. The training set comprises approximately 4 000 texts, each annotated as either CRITICAL or CONSPIRACY. The training set is imbalanced, with 66% of the texts being critical and 33% conspiracy texts.

To assess the performance of our models, we partitioned the original training set and created a development set, randomly choosing 200 texts per category. This balanced distribution ensured a robust evaluation across both the categories. Furthermore, a test set consisting of 1 000 texts was made available by the shared task organizers.

3.2. Data Augmentation

To provide more data to our language model (LM) approach, we augmented the data and used the all the texts (mainstream and conspiracy) from the LOCO [3] corpus. LOCO is a corpus consisting of conspiracy and mainstream texts for a range of conspiracy topics, including COVID related narratives. The corpus was collected automatically, using the conspiracy score by `factCheck.org` to find websites with conspiracy content, and using a google search of the same seed terms for finding mainstream texts. The corpus covers 47 seeds, including COVID19 and coronavirus, but also 5G, Elvis death, or Illuminati. We assigned the conspiracy texts the label CONSPIRACY and the mainstream texts CRITICAL. This changes the class distribution in the training data to 25.1% conspiracy texts and 74.9% critical texts.

The reason for choosing conspiracy texts across all conspiracy theories present in LOCO was to develop a more generalized LM based system capable of capturing a wide range of conspiracy texts, rather than focusing specifically on COVID-related texts.

4. Methodology

4.1. Data Exploration

After a brief skimming of the training data, we found that the URLs were tokenized, with spaces separating the URL path within the domain. Some of the URLs were shortened, and certain texts contained a combination of both standard URLs and shortened URLs.

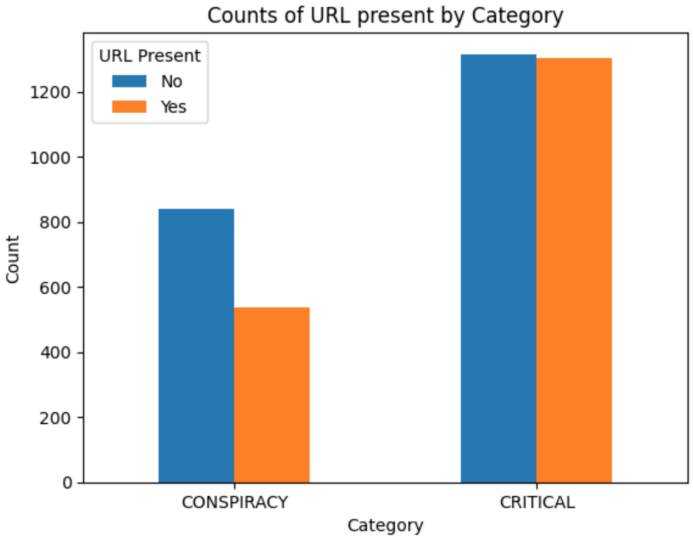


Figure 2: Distribution of URLs in the PAN dataset.

For example, consider the text in Figure 1. This text shows a split URL and incorrectly rendered quotes. Cleaning such a dataset proved to be a challenging task in itself. We used regular expressions to cover as many of the URLs as possible, but there are some that our method did not identify.

4.2. Data Preprocessing

For the statistical models we initiated data preprocessing by converting all text to lowercase. Subsequently, we replaced URLs with the token 'URL' to mitigate their influence and enhance the dataset quality, using regular expressions. A frequency investigation of the words in the classes shows a higher prevalence of URLs in the conspiracy texts. However, upon examination, we found that both categories contained URLs, as depicted in Figure 2.

Utilizing the 'en_core_web_sm' pipeline from spaCy[2], we then removed stopwords to evaluate their impact on traditional models. Additionally, we extracted metadata such as sentence lengths and word counts. We performed part of speech tagging using the POS tagger provided in spaCy. Each word was then replaced with its corresponding POS tag.

We also investigated content word bleaching, in the assumption that words that are very specific for a given conspiracy theory will be detrimental to detecting other conspiracy theories. Following Fort et al. [8], we used Latent Dirichlet Allocation (LDA) [11] from the gensim library[3] to cluster the training texts and extract the words that are most closely related to a topic. We clustered the texts into a single cluster as we knew that all texts were related to COVID-19. We then retrieved the top 25 topic words and replaced those words in the training data by the word TOPIC.

These preprocessing steps resulted in the creation of different types of texts, namely:

1. **text**: raw text
2. **POS tagged**: words replaced by their corresponding POS tags
3. **preprocessed**: lower cased text with URLs replaced
4. **no stopwords**: stop words removed
5. **bleached**: top 25 words replaced in the preprocessed text.

Due to time limitations and the size of the LOCO corpus, we only performed the experiments using preprocessed data on the original training set for statistical models.

[2] https://spacy.io
[3] https://radimrehurek.com/gensim/

Table 1
Individual statistical models and techniques used to create the best performing ensembles; evaluated on the development set.

Text	Features	Model	MCC
bleached	TF-IDF	SVC	0.7407
no stopwords	BOW	LR	0.7350
no stopwords	TF-IDF	SVC	0.7144
text	TF-IDF	SVC	0.7143
text	BOW	LR	0.7115
no stopwords	BOW	XGBoost	0.7072
text	BOW	Naïve Bayes	0.7050
no stopwords	TF-IDF	LR	0.7014
bleached	BOW	Naïve Bayes	0.7008
no stopwords	BOW	Random Forest	0.6998
bleached	TF-IDF	Random Forest	0.6980

4.3. Statistical Models

We used the enriched texts and word embeddings to train Multinomial Naïve Bayes, Random Forest, XGBoost, Logistic Regression, and SVC models.

We chose the Naïve Bayes classifier for its simplicity and ability to handle missing data values. The Support Vector Classifier (SVC) excels at handling high-dimensional spaces and is robust against overfitting. Random Forest, an ensemble learning method, is also robust to overfitting and provides feature importance ranking, which helps identify the most influential features. Logistic Regression and Multinomial Naïve Bayes classifiers are both easy to interpret and computationally efficient. XGBoost offers a highly efficient and scalable approach to handle data. All models, except for XGBoost, were implemented using scikit-learn. For XGBoost, we used the Python XGBoost module to train our systems. We optimized the classifiers using grid search and 5-fold cross-validation.

We also trained an ensemble of the top 15 models that performed best on our development set. This process resulted in the creation of seven different ensembles, trained on the top 3, 5, 7, 9, 11, 13, and 15 models. Among these, the ensemble of the top 11 models achieved the highest MCC score.

To create the best performing ensemble, we used the experiments reported in section 5.2. We show the models selected for the best performing ensemble of 11 models in Table 1.

4.4. Feature Extraction for the Statistical Models

We employed traditional features, using TF-IDF with the maximum feature parameter varied at 1000, 5000, and 10000, and bag of words with n-gram ranges from 1 to 5, as well as modern embedding-based approaches to create features for the statistical models.

Initially, we employed BERT-based embeddings to train our statistical models. However, BERT models can handle a maximum sequence length of 512 tokens, which was insufficient for our needs [12]. Therefore, we transitioned to the Longformer model, which supports sequences up to 4096 tokens in length without requiring extensive memory and computational resources [13]. To extract word embeddings, we processed the text with the Longformer model and utilized the output from the final hidden layer as our embeddings. We applied these feature extraction techniques to the five types of texts generated during data preprocessing to train the statistical models.

While training the statistical models, we found that using the POS-tagged text led to significant information loss. Therefore, we discontinued the POS based models early on. The experiments included the four remaining text representations, three feature extraction techniques, and six types of models.

Table 2
Official competition results, evaluated on the test data.

System	MCC	macro F1	F1 conspiracy	F1 critical	Rank
IUCL	0.8388	0.9194	0.8947	0.9441	1
AI_Fusion	0.8303	0.9147	0.8866	0.9429	2
baseline-BERT	0.7964	0.8975	0.8632	0.9318	(17-18)
IUCL 2	0.7845	0.8896	0.8610	0.9181	(27-28)

4.5. Finetuned Language Models

We used the transformers library by Hugging Face [14] to finetune our language models. The language models were loaded in 4 bit precision to reduce memory usage and speed up training. We used the Parameter Efficient Finetuning (PEFT) [15] library to load the models. Our experiments focused primarily on the DeBERTa [1] and Llama-2-7b [16] models. DeBERTAa is a transformer-based model developed by Microsoft, which builds on the BERT architecture by introducing a new attention mechanism. Llama-2-7b is a transformer-based LM developed by Meta AI, which is trained on a large corpus of text. Both language models were finetuned for three epochs.

Both language models were trained on the first 512 tokens per text. DeBERTa also predicted on the first 512 tokens while Llama-2-7b performed best when predicting on the first 2048 tokens. Both language models were first trained only on the PAN24 corpus, followed, where applicable, by training on an augmented training set that included the LOCO corpus.

Although the LMs can handle longer sequences, computational constraints limited the sequence length to 512 tokens for training. However, we conducted one experiment using 2048 tokens for training, which did not yield better performance compared to 512 tokens. Even with this limitation of training on 512 tokens, we performed inference on 2048 tokens for the models based on Llama-2-7b and Llama-3-8b architectures to capture a much larger context. This led to a significant increase in the performance of the LMs.

The hyperparameters used for the submitted system using DeBERTa were as follows: learning rate of 5e-5, token length of 512, gradient accumulation steps of 8, and a batch size of 1. The Low-Rank Adaptation parameters used for the PEFT library were as follows: Rank (r) of 64, alpha of 16, a dropout of 0.1 and a task type of Sequence Classification. All linear layers were updated during training. No bias was used.

4.6. Evaluation

The organizers used the Matthews Correlation Coefficient (MCC) [17] as the evaluation metric to assess the systems. We adopted the same metric to perform internal evaluation of our statistical models, language models, and ensembles.

5. Results

Here, we first provide an overview of the official results of our two systems then we will discuss results of our internal experiments with the statistical models, the LMs, and the ensembles, on our development data.

5.1. Official Results

We were allowed to submit two systems. For the first system, we chose a DeBERTa model finetuned on the entire PAN and LOCO datasets. Fr the second system, we chose an ensemble that includes the augmented DeBERTa system plus Llama2 finetuned on the PAN dataset, plus the statistical models of the best ensemble of traditional models. The latter was chosen since it performed best in our internal evaluation (see next section).

Table 3
The best results from the statistical models, evaluated on our development set. The best results for each model are highlighted in bold.

Model	Preprocessing	Features	MCC
Logistic Regression	text	BOW	0.7115
		TF-IDF	0.6812
		embeddings	0.6652
	preprocessed	BOW	0.6957
		TF-IDF	0.6843
		embeddings	0.6372
	no stopw.	BOW	**0.7350**
		TF-IDF	0.7014
		embeddings	0.6372
	bleached	BOW	0.6945
		TF-IDF	0.6832
		embeddings	0.6041
SVC	text	BOW	0.5634
		TF-IDF	0.7143
		embeddings	0.6539
	preprocessed	BOW	0.6024
		TF-IDF	0.6754
		embeddings	0.6547
	no stopw.	BOW	0.5078
		TF-IDF	0.7144
		embeddings	0.6201
	bleached	BOW	0.6030
		TF-IDF	**0.7407**
		embeddings	0.6043

The official results in Table 2 show that our system using DeBERTa with the augmented training set reached an MCC of 0.8388, and was ranked the best system in the competition, out of 83 teams. Our second submitted system reached an MCC of 0.7845. It was not officially ranked but places between ranks 27 and 28 on the official ranking[4], thus not reaching the baseline BERT system.

5.2. Results on Development Data

5.2.1. Statistical Models

The results for the two best performing models among the statistical models, SVC and Logistic Regression, are shown in Table 3. The best result is reached by using the SVC with TF-IDF and bleached topic words. The second highest result is reached by Logistic Regression when using a standard bag of words after bleaching topic words.

Overall, we find that the best performing settings depend on the model choice. For Logistic Regression, the bag of words outperforms TF-IDF features as well as word embeddings, and stopword removal works well. For SVC, TF-IDF features are best, followed by word embeddings. Here, bleaching topics improves results, while the other preprocessing strategies have a mixed influence, depending on the feature representation.

Our preliminary results showed that the removal of URLs has an overall negative impact on statistical classifiers. There are exceptions (e.g., when using the SVC with embeddings or using Logistic Regression with TF-IDF features), but there are no consistent trends.

The MCC results mostly depend on the model's success on the conspiracy class. This makes sense as the conspiracy is the minority class, and thus harder to predict. However, the F-scores for this class

[4] https://pan.webis.de/clef24/pan24-web/oppositional-thinking-analysis.html

Table 4
Results of the experiments using LMs.

Model	Augmentation	MCC
Llama2-7b	No	0.8601
Llama-2-7b (no stopwords)	No	0.8051
Llama2-7b	Yes	**0.8607**
Llama3-8b	Yes	0.8253
DeBERTa Large	No	0.7951
DeBERTa Large	Yes	0.8358

Table 5
Results of the ensembles.

Type of models	number of models	MCC
Llama2-7b (non-augm.) + DeBERTa (augm.)	2	0.8955
IUCL2	3	**0.9000**
Statistical	1	0.7407
	3	0.7333
	5	0.7452
	7	0.7553
	9	0.7496
	11	**0.7686**
	13	0.7630
	15	0.7630

tend to be around 2-3 points lower that the corresponding F-scores for the critical class. Thus, the data imbalance does not hamper the systems much.

5.2.2. Language Models

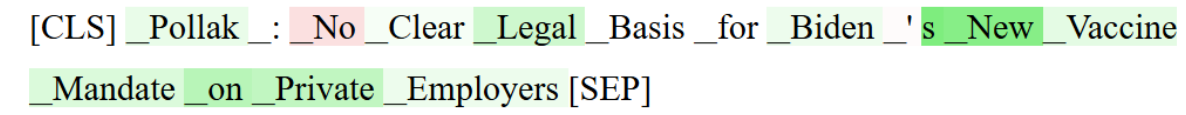
The results of our experiments using LMs, evaluated on the development data, are shown in Table 4. They show that the best results are reached by using Llama 2. For this model, augmenting the training data results in a minor improvement from an MCC of 0.8601 to 0.8607. For DeBERTa, in contrast, augmenting the training data results in a sizable increase from an MCC of 0.7951 to 0.8358. Additionally, we found that augmenting the language models with the LOCO data accelerates convergence and improves results.

The increase for DeBERTa when using the augmented training data is surprising in that the LOCO data used for augmenting the training data differ from the training data in the text type (websites vs. Telegram), they cover a wider range of conspiracy theories, and the texts were sampled to highlight the distinction between conspiracy and mainstream content, rather than conspiracy and critical thinking. The fact that DeBERTa can successfully use the additional training data leads us to the assumption that there is a language of conspiracy theories, with a distinct style that this LM can learn to recognize.

5.2.3. Ensembles

The results of the ensembles, either combining LMs with statistical models or statistical models only, are shown in Table 5. These results show that the best ensemble of 11 classifiers gains about 2.6 points on the best performing single classifier. However, this combination does not reach the ensemble including LMs. Our second submission system, IUCL2, reaches an MCC of 0.90 on our development set. Note that this is about 4 points higher than the results of the augmented Llama 2 and about 6.5 points higher than the augmented DeBERTa model. This trend is in contrast to the results on the test data, where the single DeBERTa model performed significantly better.

Critical:



Conspiracy:

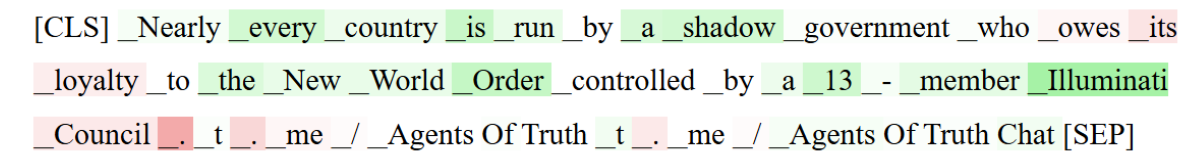


Figure 3: Example of the saliency of words in a critical and a conspiracy text.

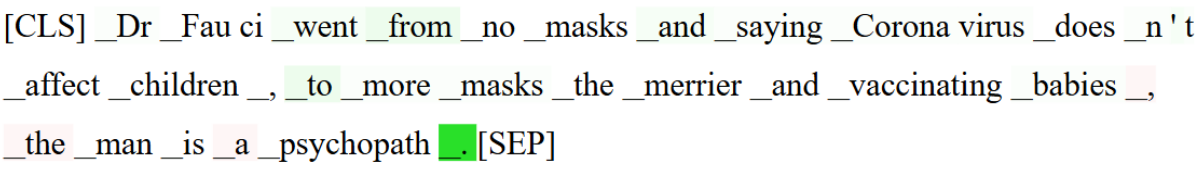


Figure 4: Example of a text falsely classified as critical.

5.3. Post-result Analysis and Model Interpretation

We conducted a post-result analysis to interpret the model performance of our winning model and check the saliency of individual words. We utilize the Integrated Gradients method [18] to calculate the importance of each word in the text. We then visualize the importance of each word in the text using the Transformers-Interpret and Captum libraries.

Our analysis shows that the model pays attention to words in the text that are often interpretable. We show two examples from the development set visualizing the saliency in the text in Figure 3. The critical example shows that the tokens “New” and ”on_Private” are highly indicative of the critical class while ‘No” would point to conspiracy. In the conspiracy text, the term “Illuminati” is the best indicator of the conspiracy class.

Figure 4 shows an example from the test set where our system classified a text as critical. This is an example where there are no solid indicators as to whether a conspiracy belief is present, and the only token receiving attention is the period, showing that this is a difficult example to classify.

6. Conclusion and Future Work

We conducted an extensive analysis of various statistical and language models, evaluating their individual performance, and the effectiveness of their ensembles, to classify texts into conspiracy and critical texts. We found that statistical models tend to underperform in comparison to LMs: the best statistical model is about 10 points below the best LM, and ensembling them increases results by about 2 points. However, this ensemble is only 3 points below the non-augmented DeBERTa.

On the official test set, our best performing model is a DeBERTa model with augmented training data. Our findings also indicate that ensembles combining traditional models with LMs outperformed individual LM results on the development set but not on the test set. This disparity suggests that finetuning sensitivity to data characteristics may have influenced performance, as gains observed in the development set did not fully generalize to the test set.

For future work, we plan to further explore and interpret the model results to gain insights into

how the language of conspiracy theories differs from that of critical texts. Another avenue of research involves investigating the uniformity of labeling and annotations, as well as the impact of this on the classification results.

Acknowledgments

This work is based on research in part supported by US National Science Foundation (NSF) Grant #2123618.

This research was supported in part by Lilly Endowment, Inc., through its support for the Indiana University Pervasive Technology Institute.

References

[1] P. He, X. Liu, J. Gao, W. Chen, DeBERTa: Decoding-enhanced BERT with disentangled attention, arXiv:2006.03654, 2021. arXiv:2006.03654.

[2] D. Korenčić, B. Chulvi, X. Bonet Casals, M. Taulé, P. a. F. Rosso, Overview of the oppositional thinking analysis PAN task at CLEF 2024, in: Working Notes of CLEF 2024 – Conference and Labs of the Evaluation Forum, Grenoble, France, 2024.

[3] A. Miani, T. Hills, A. Bangerter, LOCO: The 88-million-word language of conspiracy corpus, Behavior Research Methods (2021). URL: https://doi.org/10.3758/s13428-021-01698-z.

[4] A. A. Ayele, N. Babakov, J. Bevendorff, X. B. Casals, B. Chulvi, D. Dementieva, A. Elnagar, D. Freitag, M. Fröbe, D. Korenčić, M. Mayerl, D. Moskovskiy, A. Mukherjee, A. Panchenko, M. Potthast, F. Rangel, N. Rizwan, P. Rosso, F. Schneider, A. Smirnova, E. Stamatatos, B. Stein, M. Taulé, D. Ustalov, X. Wang, M. Wiegmann, S. M. Yimam, E. Zangerle, Overview of PAN 2024: Multi-author writing style analysis, multilingual text detoxification, oppositional thinking analysis, and generative ai authorship verification - condensed lab overview, in: Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Fifteenth International Conference of the CLEF Association, Grenoble, France, 2024.

[5] J. Uscinski, A. Enders, C. Klofstad, M. Seelig, H. Drochon, K. Premaratne, M. Murthi, Have beliefs in conspiracy theories increased over time?, PLOS ONE 17 (2022) e0270429.

[6] C. Klofstad, O. Christley, A. Diekman, S. Kübler, A. Enders, J. Funchion, S. Littrell, M. Murthi, K. Premaratne, M. Seelig, D. Verdear, S. Wuchty, H. Drochon, J. Uscinski, Belief in white replacement, Politics, Groups, and Identities (2024). doi:10.1080/21565503.2024.2342834.

[7] I. Sakki, L. Castrén, Dehumanization through humour and conspiracies in online hate towards Chinese people during the COVID-19 pandemic, British Journal of Social Psychology 61 (2022).

[8] M. Fort, Z. Tian, E. Gabel, N. Georgiades, N. Sauer, D. Dakota, S. Kübler, Bigfoot in big tech: Detecting out of domain conspiracy theories, in: Proceedings of the Conference on Recent Advances in NLP (RANLP), Varna, Bulgaria, 2023, pp. 353–363.

[9] Y. Peskine, D. Korenčić, I. Grubisic, P. Papotti, R. Troncy, P. Rosso, Definitions matter: Guiding GPT for multi-label classification, in: Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, 2023, pp. 4054–4063. URL: https://aclanthology.org/2023.findings-emnlp.267. doi:10.18653/v1/2023.findings-emnlp.267.

[10] M. Reiter-Haas, B. Klösch, M. Hadler, E. Lex, Framing analysis of health-related narratives: Conspiracy versus mainstream media, arXiv:2401.10030, 2024.

[11] D. Blei, A. Ng, M. Jordan, Latent dirichlet allocation, The Journal of Machine Learning Research 3 (2001) 601–608.

[12] 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, Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186. URL: https://aclanthology.org/N19-1423. doi:10.18653/v1/N19-1423.

[13] I. Beltagy, M. E. Peters, A. Cohan, Longformer: The long-document transformer, arXiv:2004.05150, 2020. arXiv:2004.05150.

[14] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest, A. Rush, Transformers: State-of-the-art natural language processing, in: Q. Liu, D. Schlangen (Eds.), Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Association for Computational Linguistics, Online, 2020, pp. 38–45. URL: https://aclanthology.org/2020.emnlp-demos.6. doi:10.18653/v1/2020.emnlp-demos.6.

[15] S. Mangrulkar, S. Gugger, L. Debut, Y. Belkada, S. Paul, B. Bossan, Peft: State-of-the-art parameter-efficient fine-tuning methods, https://github.com/huggingface/peft, 2022.

[16] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. C. Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P. S. Koura, M.-A. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. M. Smith, R. Subramanian, X. E. Tan, B. Tang, R. Taylor, A. Williams, J. X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov, T. Scialom, Llama 2: Open foundation and fine-tuned chat models, 2023. arXiv:2307.09288.

[17] D. Chicco, N. Tötsch, G. Jurman, The Matthews correlation coefficient (MCC) is more reliable than balanced accuracy, bookmaker informedness, and markedness in two-class confusion matrix evaluation, BioData Mining 14 (2012).

[18] M. Sundararajan, A. Taly, Q. Yan, Axiomatic attribution for deep networks, in: D. Precup, Y. W. Teh (Eds.), Proceedings of the 34th International Conference on Machine Learning, volume 70 of *Proceedings of Machine Learning Research*, PMLR, 2017, pp. 3319–3328. URL: https://proceedings.mlr.press/v70/sundararajan17a.html.