Current language models' poor performance on pragmatic aspects of natural language

Albert Pritzkau[1,*,†], Julia Waldmüller[2,†], Olivier Blanc[2,†], Michaela Geierhos[2] and Ulrich Schade[1]

[1]*Fraunhofer Institute for Communication, Information Processing and Ergonomics (FKIE), Fraunhoferstraße 20, 53343 Wachtberg, Germany*

[2]*Research Institute for Cyber Defence and Smart Data (CODE), University of the Bundeswehr Munich, Werner-Heisenberg-Weg 39, 85577 Neubiberg, Germany*

Abstract

With the following system description, we present our approach for claim detection in tweets. We address both Subtask A, a binary sequence classification task, and Subtask B, a token classification task. For the first of the two subtasks, each input chunk—in this case, each tweet—was given a class label. For the second subtask, a label was assigned to each individual token in an input sequence. In order to match each utterance with the appropriate class label, we used pre-trained RoBERTa (A Robustly Optimized BERT Pretraining Approach) language models for sequence classification. Using the provided data and annotations as training data, we fine-tuned a model for each of the two classification tasks. Though the resulting models serve as adequate baseline models, the exploratory data analysis suggests fundamental problems in the structure of the training data. We argue that such tasks cannot be fully solved if pragmatic aspects of language are ignored. This type of information, often contextual and thus not explicitly stated in written language, is insufficiently represented in the current models. For this reason, we posit that the provided training data is under-specified and imperfectly suited to these classification tasks.

Keywords

Pragmatics, Information Extraction, Text Classification, RoBERTa

1. Introduction

Political rhetoric, propaganda, and advertising are all examples of persuasive discourse. As defined by Lakoff [1], persuasive discourse is the non-reciprocal "attempt or intention of one party to change the behavior, feelings, intentions, or viewpoint of another by communicative means". Thus, in addition to the purely content-related features of communication, the

Forum for Information Retrieval Evaluation, December 15–18, 2023, India

[*] Corresponding author.

[†] These authors contributed equally.

albert.pritzkau@fkie.fraunhofer.de (A. Pritzkau); julia.waldmueller@unibw.de (J. Waldmüller); olivier.blanc@unibw.de (O. Blanc); michaela.geierhos@unibw.de (M. Geierhos); ulrich.schade@fkie.fraunhofer.de (U. Schade)

https://www.fkie.fraunhofer.de (A. Pritzkau); https://go.unibw.de/waldmueller (J. Waldmüller); https://go.unibw.de/blanc (O. Blanc); https://go.unibw.de/geierhos (M. Geierhos); https://www.fkie.fraunhofer.de (U. Schade)

0000-0001-7985-0822 (A. Pritzkau); 0000-0002-8180-5606 (M. Geierhos)

discursive context of utterances plays a central role. The shared task *CLAIMSCAN'2023* [2] on the topic *Uncovering Truth in Social Media through Claim Detection and Identification of Claim Spans* considers claims as a key element of current information campaigns, with the aim to mislead and deceive. The goal of both Subtasks A and B is to develop systems that can effectively detect and identify claims in social media text. The utterance of a particular claim is understood as a communicative phenomenon. This approach assumes that communication depends not only on the meaning of the words in an utterance but also on what speakers intend to communicate with a particular utterance. In linguistics such an approach is adopted by the field of pragmatics. It is not always possible to deduce the function of an utterance from its form. Additional contextual information is often needed. Recent research [3, 4] suggests the possibility that transformer-based networks capture structural information about language, ranging from orthographic, morphological and syntactical up to semantic features. Beyond these features, these architectures remain almost entirely unexplored. This task is an attempt to explore the limits of the prevailing approach, in particular, to investigate the ability of transformers to capture pragmatic features.

The shared task *CLAIMSCAN'2023* defines the following subtasks:

Subtask A. Claim Detection [5]: The task is a binary classification problem, where the objective is to label the given social media post as a claim or non-claim. A claim is an assertive statement that may or may not have evidence.

Subtask B. Claim Span Identification [6]: The task is to identify the words/phrases that contribute to the claims made in the given social media post. A claim is an assertive statement that may or may not be supported by evidence.

2. Background

The linguistic field of pragmatics regards speaking as acting, or more precisely, as acting with the intention of manipulating the audience. The speech act called assertion [7, 8] means to make a statement so that the audience is informed about something. According to Grice's cooperative principle [9], the information provided must be relevant, helpful, and true in the context of the discourse. Since we are attuned to this principle, false claims are effective if they show no signs of falsehood or duplicity. We simply follow the cooperative principle and take the statement to be true, with all the consequences it implies. Signs of falsehood or duplicity can save us from such a washout. Such signs can be violations of one's own beliefs (e.g., 'Hawaiian wildfire is an attack experiment of a weather weapon conducted by the US military'), a wrong style, e.g. excessive emotion in a news text (e.g., 'Hawaiian wildfire is a scandalous attack experiment of a perfidious weather weapon conducted by the sleazy US military'), or untypical grammatical errors like omitting determiners (e.g., 'Hawaiian wildfire is attack experiment of weather weapon conducted by US military'). However, some of these signs might be overlooked because of our attunement to the cooperative principle in general and Grice's maxim of quality (ibidem) in particular. A system might therefore perform better at detecting false claims.

Task descriptions

This paper describes the participation in both subtasks. The challenge for Subtask A is to decide whether a given tweet contains a claim. Accordingly, the task is formulated as a binary classification problem. Beyond the mere identification of claims, Subtask B involves the delineation of text intervals containing said claims. For each token in a tweet, it is to be examined whether it is part of a claim, and subsequently, the claim span is to be determined. The model should then predict the indices of the span intervals for each tweet.

Exploratory Data Analysis

The organizers of the CodaLab competition *CLAIMSCAN'2023* have released the datasets for both subtasks. Each subtask dataset consists of a training set, a development set, and a test set, all focused on discussions related to the COVID-19 pandemic.

The labeled data for Subtask A, obtained from 8,483 tweets, includes both the training set of 6,986 tweets and the developer set of 1,497 tweets resulting in a ratio of 82:18. Assuming that the split is already validated, we did not apply any resampling. Both sets consist of the tweets in plain text with an additional binary label *claim* or *non-claim*. While the definition of a claim was given as *a claim is an assertive statement that may or may not have evidence*, we observed questionable annotations of the training set. For example, the tweet

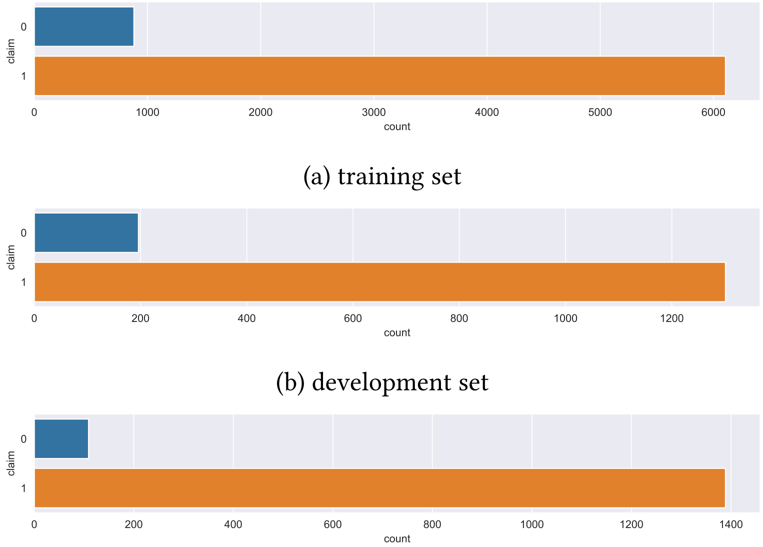
'Older but still relevant: Health products that make false or misleading claims to prevent, treat or cure #COVID19 may put your health at risk via HealthCanada #cdnhealth https://t.co/9dFNXaV3gW'

is labeled as a *non-claim*. However, the tweet

'coronavirus altnews founder shekhar gupta and others spread unverified claims by a fake twitter account'

is marked as a *claim*. For the purpose of submission, an unlabeled test set consisting of 1,489 tweets was used.

For Subtask B, the size of the training set was 6,044, the development set had 756 tweets resulting in a ratio of 89:11. The test dataset contained 755 entries. In contrast to Task A, here, in addition to the tweet text and the claim label, the start index and the end index of the token spans corresponding to the claims were also provided. Of these, 7,585 spans were annotated as claims, meaning that some tweets contained more than one claim. As in Subtask A, we made several notable observations regarding the labeled training data. We observed an instance of an impossible annotation in line 19 of the training set. This anomaly raised questions regarding the quality of data and the need for quality control mechanisms when building the dataset. Furthermore, during our analysis of annotation spans, it was revealed that 235 '@' mentions and 16 URLs (starting with "https://...") were present in the annotated text. We discovered that colons appeared to be the most indicative feature for identifying the beginning of a claim, with 846 instances manifesting this pattern within the training dataset. Additionally, the data analysis suggests the utilization of keyword-based sampling in the construction of the training dataset. This is particularly evident from Figure 3. This is supported, for example, by the fact that the account name of Donald Trump (@realdonaldtrump) appears in the top 30 most frequent words (see Figure 3b). Surprisingly, we found that cleaning the training data resulted in a poorer performance of the model.



(a) training set

(b) development set

(c) test set

Figure 1: Label distribution and class imbalance for Subtask A.

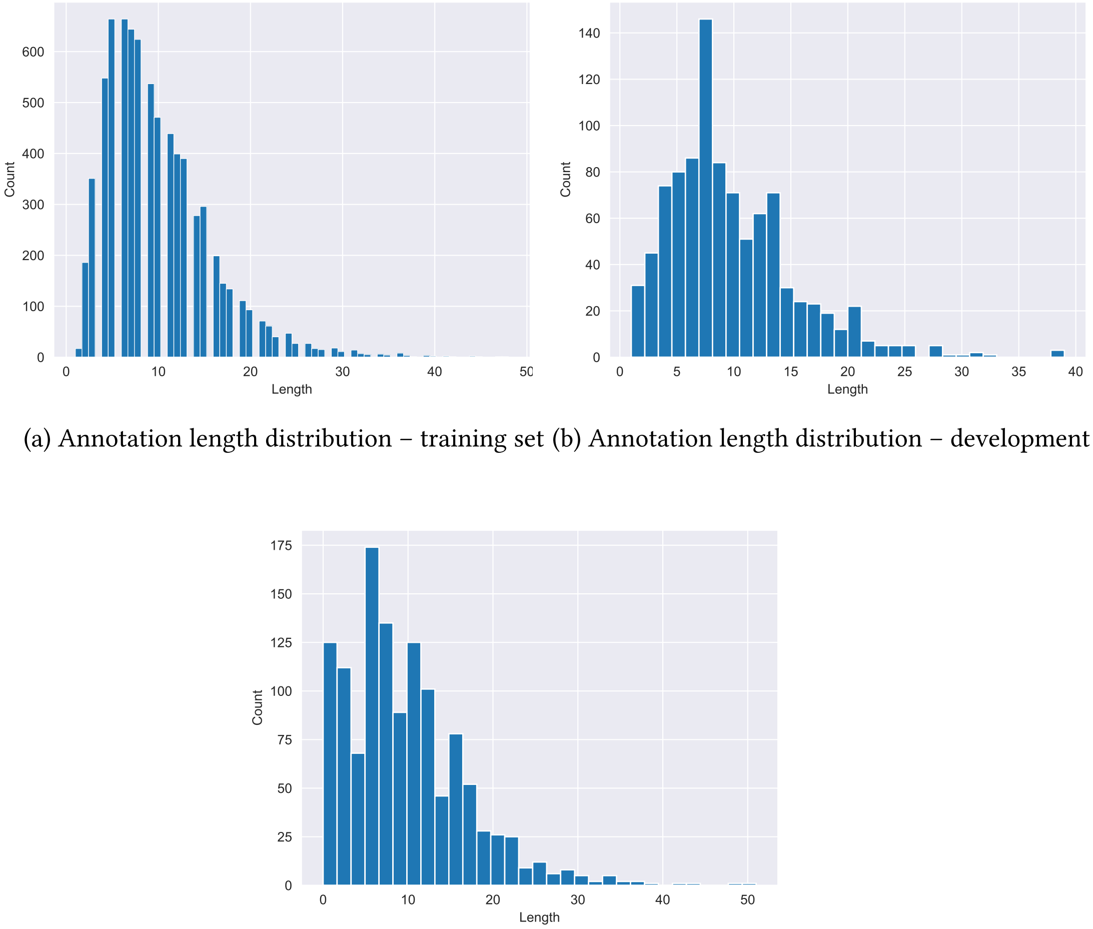
The value and meaning of accuracy and other well-known performance metrics of an analytical model can be greatly affected by data imbalance. As shown in Figures 1a and 1b, the class distribution is skewed. This poses a challenge for the balanced learning of the model, as the non-claim class is significantly underrepresented. When comparing the distributions of annotation length in the training set, development set, and test set, as shown in Figures 2, it becomes apparent that these significantly deviate from each other and, in some cases, exhibit a strong concentration of data points within specific groups.

3. System overview

In this study, we evaluate and compare a sequence classification approach on the given data with different augmentations. The comparison is performed at the level of trained models on the same dataset. The different evaluation paradigms result from applying the sequence classifier heads to a pre-trained model as a base model. We suggest that contextual information leads to a qualitative difference in the scores.

3.1. Pre-trained language representation

At the core of any solution to a given task is a pre-trained language model derived from BERT [10]. BERT stands for Bidirectional Encoder Representations from Transformers. It is

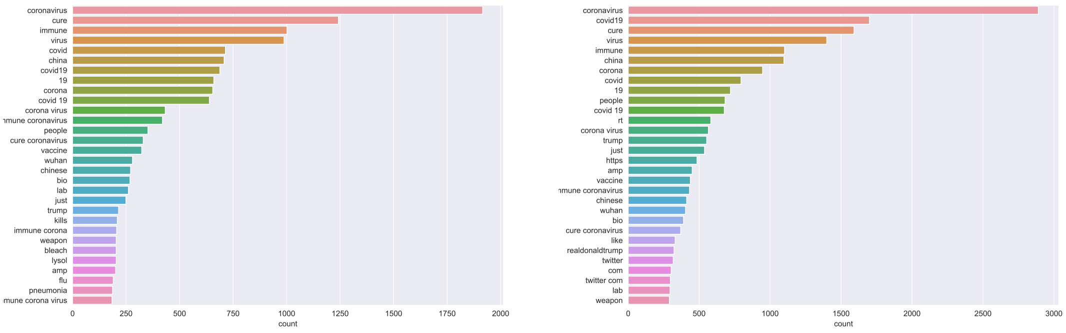


(a) Annotation length distribution – training set (b) Annotation length distribution – development set

(c) Annotation length distribution – test set

Figure 2: Comparison of the annotation length distributions of the training, development, and test sets for Subtask B.

based on the transformer model architectures introduced by Vaswani et al. [11]. The general approach consists of two stages. First, BERT is pre-trained on large amounts of text, with the unsupervised goal of masked language modeling and next sentence prediction. Second, this pre-trained network is then fine-tuned on task-specific, labeled data. The transformer architecture consists of two parts, an encoder and a decoder, for each of the two stages. The encoder used in BERT is an attention-based architecture for NLP. It works by performing a small, constant number of steps. In each step, it applies an attention mechanism to understand the relationships between all the words in a sentence, regardless of their respective positions. By pre-training language representations, the encoder yields models that can either be used to extract high-quality language features from text data or to fine-tune these models for specific NLP tasks (classification, entity recognition, question answering, etc.). We rely on RoBERTa



(a) Term distributions in annotation spans.

(b) Term distributions in full text.

Figure 3: Comparison of the term distributions in the annotation spans and in the full text of the messages (top 30) in the training set for Subtask B.

[12], a pre-trained encoder model that builds on BERT's language masking strategy. However, it modifies key hyperparameters in BERT, such as removing BERT's next-sentence pre-training objective and training with much larger mini-batches and learning rates. In addition, RoBERTa has been trained on an order of magnitude more data than BERT, for a longer period of time. This allows RoBERTa representations to generalize downstream tasks even better than BERT.

3.2. Binary Sequence Classification Problem

Model Architecture – NLytics. Subtask A is considered to be a binary classification problem. The models for the experimental setup were based on RoBERTa. For the classification task, fine-tuning is first performed using RobertaForSequenceClassification [13] — $RoBERTa_{LARGE}$ — as the pre-trained model. RobertaForSequenceClassification optimizes for a regression loss (Binary Cross-Entropy Loss) using an AdamW optimizer [14] with an initial learning rate set to 2e-5. After a warm-up period during which the learning rate increases linearly from 0 to the initial learning rate, the optimizer is scheduled to decrease the actual learning rate linearly to 0. The training was started with 20 training epochs each. However, this relatively high number is significantly reduced by an early stopping callback that monitors the performance of the model on the validation dataset. A patience of 5 epochs is set for this callback. For this setup, fine-tuning was done on an NVIDIA TESLA V100 GPU using the Pytorch [15] framework with a vocabulary size of 50,265 and an input size of 512.

Model Architecture – CODE. The experimental setup and approach for the binary classification problem are almost identical to the one above. Instead of RoBERTa, we used BERT [10]. Therefore, we fine-tuned the model using BertForSequenceClassification. This model was also trained for five epochs, following the same approach described above. A NVIDIA GeForce RTX 3090 GPU with 24GB of memory was used for fine-tuning using Pytorch [15].

3.3. Token Classification Problem

Tagging format. We have transformed the initial span markup into the IOB (Inside, Outside, Begin) tagging format. Since we have only one possible entity class, each token can be assigned one of the tags given by *O-claim*, *B-claim*, and *I-claim*.

Model Architecture – NLytics. Subtask B is considered to be a token classification problem. We have fine-tuned a RoBERTa model to predict the above IOB tags for each token in the input sentence. In the default configuration, each token is classified independently of the surrounding tokens. Although the surrounding tokens are taken into account in the contextualized embeddings, there is no modeling of the dependency between the predicted labels: for example, an I tag cannot logically follow an O tag. Since RoBERTa does not model the dependencies between the predicted tokens, we further added a linear-chain Conditional Random Field (CRF) model [16] as an additional layer, in order to model the dependency between the predicted labels of individual tokens. Since the sequence of an O tag following an I tag does not occur in the training set, it assigns a very low probability to the transition from an O tag to an I tag by observation. The CRF receives the logits for each input token, and makes a prediction for the entire input sequence, taking into account the dependencies between the labels, similar to Lample et al. [17]. Note that RoBERTa works with byte pair encoding (BPE) units, while the CRF needs to work with whole words. Thus, only head tokens were used as input to the CRF, and any word continuation tokens were omitted. The models for the experimental setup are based on RoBERTa. For the classification task, fine-tuning is first performed using RobertaForSequenceClassification [13] — $RoBERTa_{LARGE}$ — as the pre-trained model. RobertaForSequenceClassification optimizes for a regression loss (Binary Cross-Entropy Loss) using an AdamW optimizer [14] with an initial learning rate set to 2e-5. After a warm-up period during which the learning rate increases linearly from 0 to the initial learning rate, the optimizer is scheduled to decrease the actual learning rate linearly to 0. The training was started with 20 training epochs each. However, this relatively high number is significantly reduced by an early stopping callback that monitors the performance of the model on the validation dataset. A patience of five epochs is set for this callback. For this setup, fine-tuning was done on an NVIDIA TESLA V100 GPU using the Pytorch [15] framework with a vocabulary size of 50,265 and an input size of 512.

Model Architecture – CODE. We also experiment with an alternative setup for the token classification problem of Subtask B, using a simplified tag set. In this setup, the RoBERTa model is fine-tuned to predict a binary label (0 or 1) for each token, describing whether the token is part of a claim or not. Unlike the IOB tag set, the first token of a claim is not distinguished and is assigned the same label 1 as the subsequent tokens that are part of the claim. For this experiment, we stop fine-tuning the RobertaForSequenceClassification model after 4 epochs on the training set to avoid overfitting, as we empirically observe a degradation of the performance on the validation dataset after this point. We observed that regularly short sequences of only one or two tokens were incorrectly annotated as claims with this setup. We therefore decide to filter out those predicted claims that are shorter than three words in a second step, in order to reduce noise and obtain more realistic annotations.

4. Results

We participated in both Subtasks A and B. Because of the similar approach, these working notes describe the results of two teams, NLytics and CODE. The official evaluation results for the test set are shown in Tables 1 and 6. In the following, the results are presented for each subtask. In the discussion of the results, we address the reasons for the differences in the performance of the two teams. The submissions were optimized for the minimum validation loss to avoid overfitting the resulting model. During the training phase, we focused on finding the best combinations of deep learning methods and optimizing the corresponding hyperparameter settings. Fine-tuning pre-trained language models like RoBERTa on downstream tasks has become ubiquitous in NLP research and applied NLP. Even without extensive preprocessing of the training data, we already achieved competitive results. The resulting models serve as strong baselines, that, when fine-tuned, significantly outperform models trained from scratch.

4.1. Subtask A

The model checkpoint with the minimum validation error was selected for submission. For NLytics, this minimum was reached after four epochs of training. The class-related differences in model performance shown in Table 2 clearly reflect the class imbalance in the initial distribution (cf. Figure 1). Different data cleaning strategies to mitigate the impact of technical structures such as URLs or account names in the linguistic evaluation, had a negative impact on the performance of the resulting models on the development set. For example, URLs were replaced with a unique sequence to clean up the data. The same happened with the account names.

As shown in Table 1, the Macro-F1 value for CODE differs from NLytics by 0.0476. This discrepancy is due to the choice of the model, as the model with the lower Macro-F1 used an uncased BERT model, despite following the same approach.

Table 1
Leaderboard of Subtask A.

Rank	Name	Macro-F1
1	**NLytics**	**0.7002**
2	bhoomeendra	0.6900
3	amr8ta	0.6678
4	**CODE**	**0.6526**
5	michaelibrahim	0.6324
6	pakapro	0.4321

Table 2
Confusion matrix for Subtask A – development set.

		predicted	
		0	1
true	0	63	133
	1	28	1273

4.2. Subtask B

For NLytics, the model checkpoint with the minimum validation error was reached after three epochs of training. Table 3 shows the corresponding evaluation metrics. The best result could only be achieved by extending the model with the CRF. Similar to the results of Subtask A, the data cleaning strategies had a negative impact on the performance of the resulting models on the development set.

Table 3
Evaluation results on the development set of Subtask B.

Metric	Value
Validation loss	0.37
Accuracy	0.84
Precision	0.50
Recall	0.62
F1	0.55

The evaluation results obtained using the model for the CODE architecture setup on the development dataset are presented in Table 4. These numbers were obtained using the RoBERTa BPE token representation and before the postprocessing step that filters out the small claims.

Table 4
CODE subtask B: Evaluation results on Byte-Pair Encoding tokens on development set.

Metric	Value
Accuracy	0.85
Precision	0.85
Recall	0.85
F1	0.85

Table 5 shows the evaluation of the same model using the original token representation and after filtering out the small claims made of only one or two words. We observe that this additional cleaning step slightly degrades the performance of the system, with the F1 score dropping from 0.85 to 0.83. In addition, after the results were published, we discovered that due to a bug in the CSV formatting of the claim span indices, all predicted claims located at the end of the input texts were missing from the submission file. As a result, 254 claims were missing out of a total of 904 claims predicted on the test data. This should explain the poor score of 0.57 we got on the leaderboard. Since the gold standard has not yet been published, we cannot provide the actual F1 score that we would have normally obtained on the test data.

5. Conclusion

The use of neural architectures in the field of pragmatics remains largely unexplored. The limitations are clearly demonstrated by the results of the given task. In the future, we would like to extend the current approach by adding features that represent the extended communicative context. Our research aims at the specification of a consistent goal function that is adapted

Table 5
CODE subtask B: Evaluation results on tokens after post-processing

Metric	Value
Accuracy	0.83
Precision	0.83
Recall	0.83
F1	0.83

Table 6
Leaderboard of Subtask B.

Rank	Name	Token-F1
1	mjs227	0.8344
2	bhoomeendra	0.8030
3	**NLytics**	**0.7821**
4	**CODE**	**0.5714**

to the discursive context of manipulative communication. We hypothesize that the target variables of this function in the form of different discourse elements will respond to different features of the given communicative context. If the required features cannot be derived from the linguistic structure of the utterances, they have to be obtained from the extended context of the communication. We are investigating ways to make external features available to the training process. Thus, in order to identify pragmatic features and to know how to take advantage of them, the application of XAI methods seems to be promising.

References

[1] R. T. Lakoff, Persuasive discourse and ordinary conversation, with examples from advertising, Analyzing discourse: Text and talk (1982) 25–42. Publisher: Georgetown, Georgetown University Press.

[2] M. Sundriyal, M. S. Akhtar, T. Chakraborty, Overview of the claimscan-2023: Uncovering truth in social media through claim detection and identification of claim spans, in: Working Notes of FIRE 2023 - Forum for Information Retrieval Evaluation, CEUR, 2023.

[3] I. Tenney, P. Xia, B. Chen, A. Wang, A. Poliak, R. T. McCoy, N. Kim, B. V. Durme, S. R. Bowman, D. Das, E. Pavlick, What do you learn from context? Probing for sentence structure in contextualized word representations, CoRR (2019). URL: http://arxiv.org/abs/1905.06316.

[4] G. Jawahar, B. Sagot, D. Seddah, What does BERT learn about the structure of language?, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 3651–3657. URL: https://aclanthology.org/P19-1356. doi:10.18653/v1/P19-1356.

[5] S. Gupta, P. Singh, M. Sundriyal, M. S. Akhtar, T. Chakraborty, LESA: Linguistic Encapsulation and Semantic Amalgamation Based Generalised Claim Detection from Online Content, in: Proceedings of the 16th Conference of the European Chapter of the As-

sociation for Computational Linguistics: Main Volume, Association for Computational Linguistics, Online, 2021, pp. 3178–3188. doi:10.18653/v1/2021.eacl-main.277.

[6] M. Sundriyal, A. Kulkarni, V. Pulastya, M. S. Akhtar, T. Chakraborty, Empowering the Fact-checkers! Automatic Identification of Claim Spans on Twitter, in: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 2022, pp. 7701–7715. doi:10.18653/v1/2022.emnlp-main.525.

[7] J. L. Austin, How to do things with words, Oxford University Press, 1975.

[8] J. R. Searle, Sprechakte: ein sprachphilosophischer Essay, Suhrkamp, 1977.

[9] H. P. Grice, Logic and conversation, in: Speech acts, Brill, 1975, pp. 41–58.

[10] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, CoRR (2018). ISBN: 1810.04805v2.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, volume 2017-Decem, 2017, pp. 5999–6009. ISSN: 10495258.

[12] 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 e-prints (2019) arXiv–1907.

[13] 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. v. Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, A. Rush, Transformers: State-of-the-Art Natural Language Processing, in: arxiv.org, 2020, pp. 38–45. doi:10.18653/v1/2020.emnlp-demos.6.

[14] I. Loshchilov, Ilya, F. Hutter, Decoupled Weight Decay Regularization, in: 7th International Conference on Learning Representations, ICLR, 2019, pp. 1–18.

[15] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, PyTorch: An imperative style, high-performance deep learning library, CoRR 32 (2019). URL: http://arxiv.org/abs/1912.01703, iSSN: 10495258.

[16] J. D. Lafferty, A. McCallum, F. C. N. Pereira, Conditional random fields: Probabilistic models for segmenting and labeling sequence data, in: Proceedings of the Eighteenth International Conference on Machine Learning, ICML '01, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2001, p. 282–289.

[17] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, C. Dyer, Neural architectures for named entity recognition, in: 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2016 - Proceedings of the Conference, Association for Computational Linguistics (ACL), 2016, pp. 260–270. doi:10.18653/v1/n16-1030.