ZHAW-CAI at CheckThat! 2023: Ensembling using Kernel Averaging

Notebook for the CheckThat! Lab at CLEF 2023

Pius von Däniken[1], Jan Deriu[1] and Mark Cieliebak[1]

[1]Zurich University of Applied Sciences, Centre for Artificial Intelligence, Winterthur, Switzerland

Abstract

We describe our approaches to sub-task 1A on multi-modal check-worthiness classification of the *CheckThat! Lab 2023* in English. The goal was to determine whether a tweet is worth fact-checking based on its text and image content. Our submission was based on a kernel ensemble of different uni-modal and multi-modal classifiers. It achieved second place out of 7 teams with an F1 score of 0.708.

Keywords

multi-modal, claim check-worthiness, multiple kernel learning, CheckThat!

1. Introduction

The *CheckThat! Lab 2023* [1] included five tasks targeting various aspects of misinformation. We describe our approach to *Task 1 Check-Worthiness in Multimodal and Unimodal Contents*, which contained two sub-tasks. Of the two sub-tasks, we participated specifically in sub-task 1A targeting multi-modal content. The goal was to classify a tweet consisting of both text and an image as *check-worthy* or not. The sub-task was offered both in Arabic and English. We only developed methods for the English data.

Check-worthiness classification represents an important first triage step in a fact-checking pipeline. Successfully removing claims that are not worth checking reduces the work-load of human fact-checkers. It has been part of all *CheckThat! Lab* iterations so far [2, 3, 4, 5, 6]. Where previously the focus was on text content, this year's sub-task 1A is a multi-modal task, involving both text and image data. This represents an important next step since much of the current content on social media is multi-modal in nature.

In this work, we will describe the approaches of our team, ZHAW-CAI, for sub-task 1A in English. We developed several different classifiers based on text only, ranging from traditional word frequency, to deep learning, and LLM solutions. We also developed a multi-modal classification model, as well as a kernel-method based ensemble model. When discussing our results, we will in particular highlight the importance of threshold selection.

CLEF 2023: Conference and Labs of the Evaluation Forum, September 18–21, 2023, Thessaloniki, Greece

vode@zhaw.ch (P. v. Däniken); deri@zhaw.ch (J. Deriu); ciel@zhaw.ch (M. Cieliebak)
0000-0001-8339-5543 (P. v. Däniken); 0000-0002-8405-1344 (J. Deriu); 0009-0007-3059-8516 (M. Cieliebak)

© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
CEUR Workshop Proceedings (CEUR-WS.org)

CEUR-WS.org/Vol-3497/paper-048.pdf

2. Related Work

The general problem of misinformation in social media has recieved a lot of interest from the community in recent years. Apart from the *CheckThat! Lab* tasks there have been tasks focusing on identifying the veracity of a claim or rumour, such as *RumourEval* [7, 8] and *FEVEROUS* [9].

The first modern systems for check-worthiness detection include *ClaimBuster* [10] and *ClaimRank* [11]. Their main focus is on identifying check-worthy claims in political debates. The various *CheckThat! Lab* check-worthiness tasks have targeted different text genres, including social media and tweets in particular. While TF-IDF features are a staple of any text classification task and have been included in systems such as *ClaimBuster*, many successful previous participants [12, 13] used fine-tuned masked language models such as *BERT* [14] and *RoBERTa* [15] in their solutions. We include both approaches in our solution. In terms of analysis of multi-modal social media content, the *Hateful Memes* challenge [16] has sparked a lot of interest in the community. For the challenge of multi-modality for disinformation in particular, we refer the reader to a recent survey [17]. The *MM-Claims* dataset [18] is a recent multi-modal claim detection dataset, on which this shared task is based. Our multi-modal sub-component is most similar to systems such as [19] that use cross-attention between modalities. However, we use a full transformer [20] encoder to fuse the modalities. Of course, an important recent development involves the use of large language models such as the *GPT* family [21] and *LLaMa* [22] that exhibit astonishing zero-shot classification capabilities. We include this approach in our solutions as well. Finally, we use a multiple kernel learning [23] approach to combine these disparate classifiers into a unified ensemble model.

3. Method

3.1. Data

The multi-modal check-worthiness sub-task is a binary classification task where a tweet consisting of a short text and an image has to be classified as check-worthy or not. During the development phase of the shared task, the organizers released training data (D_{train}), validation data (D_{dev}) and a *dev-test* set to be used for evaluation during development ($D_{dev-test}$). The test data D_{test} was released shortly before the submission deadline and its labels were only released after the submission deadline. For all our experiments, we combine the D_{dev} and $D_{dev-test}$ sets into a single validation set D_{valid}. The individual systems are trained on D_{train} and evaluated on D_{test}. The sizes of these sets and their label distributions are shown in Table 1. We note that each sample contained both text and image data. The training and development data came from the *MM-Claims* dataset [18] and for the full description of the task data, we refer the reader to the task overview [24].

3.2. Systems

We will now describe the different uni-modal and multi-modal systems we trained and our method to combine them using a kernel-based ensemble.

Table 1
Information for the English Data

	Number of Samples	Number of Check-worthy	Number non-check-worthy
D_{train}	2356	820 (34.8%)	1536 (65.2%)
D_{dev}	271	87 (32.1%)	184 (67.9%)
$D_{dev-test}$	548	174 (32.8%)	374 (68.2%)
D_{valid}	819 (= 271 + 548)	261 (31.9%)	558 (68.1%)
D_{test}	736	277 (37.6%)	459 (62.4%)

3.2.1. Text N-gram Classifier

Our first uni-modal system is based on the tweet text only. We first pre-process the texts by replacing URLs [1], user handles, and sequences of emoji [2] by placeholder tokens. The text was then lower-cased and tokenized by splitting on white-space. Tokens shorter than 2 characters were discarded. Based on this we computed TF-IDF [25] vectors for each text. This means counting the uni-grams and bi-grams of tokens for each sample. We count only one occurrence for each n-gram, meaning we ignore repetitions. We also ignore n-grams that appear in fewer than 3 samples in D_{train}. Based on these counts one can compute the inverse document frequency (IDF) for each token. The resulting feature vectors are normalized to have unit euclidean length. We used the *TfidfVectorizer* implementation provided by *scikit-learn* [26]. We call the resulting feature vectors $x_{text-ngram}$.

We then use these feature vectors to train a linear Support Vector Machine (SVM) [27] with regularization strength of 1. We again rely on the implementation provided by *scikit-learn*. In particular we also employ their implementation of reweighing the classes based on their frequency in the training data which was inspired by [28]. We will call this model *text-ngram*.

3.2.2. MLM Classifier

Next, we trained another text-only system. For this we fine-tuned an *electra-base-discriminator* [29] model on the training data. *Electra* models have the same architecture as *BERT* [14] but follow a different pre-training setup. During masked language modelling (MLM) pre-training there is both a generator network G and a discriminator network D. During pre-training a certain number of input tokens are masked and G has to predict the original token. The masked tokens are then replaced by those predicted by G and D has to determine whether a token was the original or has been replaced.

For our experiments we use the provided discriminator model checkpoint from *Huggingface* [3] [30]. We show the training hyper-parameters in Table 2. We will call the resulting model *electra-clf*.

In section 3.2.5 we will need access to a feature vector extracted from *electra-clf*. For this we remove the final dense layer of *electra-clf* and use the model activations as feature vectors and

[1]For this we use the *urlextract* package: https://github.com/lipoja/URLExtract.
[2]For this we use the *emoji* package: https://github.com/carpedm20/emoji/.
[3]https://huggingface.co/google/electra-base-discriminator

Table 2
Training Hyper-parameters for *electra-clf*

Parameter	Value
Epochs	10
Batch Size	16
Optimizer	AdamW [31]
Learning Rate	5e − 5
Weight Decay	0.01

scale them to unit length. We will refer to these feature vectors as $x_{electra}$.

3.2.3. Multi-Modal Classifier

Our multi-modal model relies on pre-trained encoder models for each modality. For text, we use the *twitter-roberta-base* [4] checkpoint from *Huggingface*. This is a *RoBERTa* [15] model that has been pre-trained on 58M tweets [32]. The output of this text encoder has dimensions $L_{text} \times d_{text}$ where L_{text} is the number of tokens and d_{text} the dimension of the token embedding.

For images, we use a Vision Transformer (*ViT*) [33] that has been pre-trained on *ImageNet21k* [34]. We again use a checkpoint provided by *Huggingface*[5]. The model takes images at a 224×224 pixel resolution as input and processes them as a sequence of 16×16 pixel patches. This results in an output representation of size $L_{img} \times d_{img}$ where L_{img} is the number of patches and d_{img} the patch embedding dimension.

We first project both representations into a shared space of dimension d_{shared} using a dense layer and *relu* activation for each representation. This results in representations of sizes $L_{text} \times d_{shared}$ and $L_{img} \times d_{shared}$. We then concatenate them to get a new representation of size $L \times d_{shared}$ where $L = L_{text} + L_{img}$. We then feed this representation through a transformer encoder [20] and a *relu* activation. The transformer encoder preserves the size of the representation and we use mean pooling across the sequence length to get an embedding $x_{multi-modal}$ of size d_{shared}. Finally, we normalize $x_{multi-modal}$ to unit length and feed it through a final dense layer for classification.

We fine-tune this model on D_{train} but keep the weights of both the *RoBERTa* and the *ViT* encoders frozen. We call the resulting model *multi-modal-clf* and show its hyper-parameters in Table 3.

3.2.4. LLM Classifier

Recent Large Language Models (LLMs) such as the GPT family [21] have shown impressive few-shot and even zero-shot classification capabilities. In particular, chain-of-thought prompting [35], where the model is asked to generate a step-by-step explanation how it arrives at a certain prediction, has shown much promise.

[4] https://huggingface.co/cardiffnlp/twitter-roberta-base
[5] https://huggingface.co/google/vit-base-patch16-224-in21k

Table 3
Training Hyper-parameters for *multi-modal-clf*

Parameter	Value
d_{text}	768
d_{img}	768
d_{shared}	256
Transformer Encoder Layers	1
Attention Heads	4
Transformer Feedforward Dimension	1024
Epochs	10
Batch Size	16
Optimizer	AdamW [31]
Learning Rate	$5e-5$
Weight Decay	0.01

Based on these observations, we constructed a very simple zero-shot classification prompt. We use the *Language Model Query Language (LMQL)* [36] to formulate the prompt and constrain the answers. We show the prompt written in *LMQL* in Listing 1.

Listing 1: LMQL Prompt

```
argmax
    Consider the following Tweet:
    {claim}
    Do you think this Tweet contains a claim that is worth
        fact-checking?
    Answer:[ANSWER]
    Reasoning:[REASON]
from
    openai/text-davinci-003
where
    STOPS_AT(REASON, ".")
    and ANSWER in ['Yes', 'No']
```

The placeholder *claim* is where we insert the tweet text. The placeholders *ANSWER* and *REASON* are filled in by the model. In our case we use *OpenAI*'s *text-davinci-003* [6] model. The answer is constrained to the words *Yes* and *No* which we can directly use as predictions, which we will call *gpt-answer*. The reasoning is constrained to be one sentence, since it should stop generating when it produces the first full stop. We apply a similar feature extraction procedure as for *text-ngram* in Section 3.2.1 to these reasoning sentences. We forgo any special token replacements and use n-grams up to length 3 but keep the other parameters the same. The resulting feature vectors will be called $x_{gpt-ngram}$. We then train a linear SVM on $x_{gpt-ngram}$ and call it *gpt-ngram*.

[6] https://platform.openai.com/docs/models/gpt-3-5

3.2.5. Kernel Ensemble

We have seen that all our base models have an associated feature vector: $x_{text-ngram}$, $x_{electra}$, $x_{multi-modal}$, and $x_{gpt-ngram}$. For each of these we can define a linear kernel. The kernel value for two samples i and j for a given system s is then defined as $k_s(i,j) = x_s^T(i)x_s(j)$, where $x_s(i)$ is the feature vector of system s for sample i. Given such a kernel k_s, we can then train an SVM. For $x_{text-ngram}$ and $x_{gpt-ngram}$ this is equivalent to their associated classifiers *text-ngram* and *gpt-ngram*. On the other hand, for $x_{electra}$ and $x_{multi-modal}$ we will call the resulting SVM classifiers *electra-kernel* and *multi-modal-kernel* respectively.

We will include an additional *ViT* encoder based feature vector $x_{img-untrained}$. It is based on the same *ViT* encoder as *multi-modal-clf*, which also provides a pooled representation for classification, which we will use as $x_{img-untrained}$. We will call the resulting kernel-based SVM classifier *img-untrained-kernel*.

Next, we show how we combine these kernels into an ensemble. Given a set of systems S, we can define their *average kernel* as:

$$k_{avg}(i,j) = \sum_{s \in S} \frac{1}{|S|} k_s(i,j)$$

This is known as a fixed rule multiple kernel learning method [23]. We can then use k_{avg} to train an SVM. Our main *submission* was based on this method and used an average kernel using *text-ngram*, *gpt-ngram*, *electra-kernel*, and *multi-modal-kernel* as components. We will also show results for *all-kernels* which additionally includes *img-untrained-kernel* in the average.[7] All kernel-based SVMs were trained using a regularization strength of 1 and frequency based class weights.

4. Results

In Table 4 we show our main results. Our *submission* achieved an F1 score of 0.708 on the test set. We note that if we use the default classification threshold [8] *electra-kernel* and *all-kernels* achieve that exact same score. This could indicate that our ensemble method is redundant. In practice, F1 scores can be sensitive to the decision threshold. In Figure 1 we show the Precision and Recall Curves for each system. They show the Precision and Recall of a system for all potential thresholds. In the plot we include lines of constant F1 in light gray. We can see that the default thresholds (black cross marks) tend to select sub-optimal operating points.

We could therefore try to find a better classification threshold. For this we can use the validation set D_{valid} and use the threshold which maximizes the F1 score on D_{valid}. The results are shown as red cross marks in Figure 1 and in the column called *Tuned Threshold* in Table 4. Since *gpt-answer* provides only binary outputs we can not change its threshold. The values for *electra-clf* and *multi-modal-clf* are missing since we did not compute their output on D_{valid}[9]. We can see that for most systems this method selects an even worse threshold. We had already

[7]The difference between *submission* and *all-kernels* was due to time constraints.
[8]For SVM-based systems the default threshold is 0, for classifiers trained using cross-entropy to produce class probabilities, the default threshold is 0.5.
[9]This was due to time constraints.

Table 4
Performance of our Systems on D_{test} based on different decision thresholds. Best values in each column in bold. See text for more details.

System	Default Threshold			Tuned Threshold			Optimal Threshold		
	P	R	F1	P	R	F1	P	R	F1
gpt-answer	0.536	**0.903**	0.673	-	-	-	-	-	-
img-untrained-kernel	0.498	0.556	0.526	0.476	**0.657**	0.552	0.440	0.852	0.581
text-ngram	0.676	0.458	0.546	0.728	0.397	0.514	0.517	0.809	0.631
gpt-ngram	0.630	0.664	0.647	0.695	0.567	0.624	0.566	**0.863**	0.684
electra-kernel	0.768	0.657	**0.708**	0.775	0.635	**0.698**	0.724	0.740	0.732
multi-modal-kernel	0.812	0.545	0.652	0.812	0.516	0.631	**0.756**	0.704	0.729
submission	0.768	0.657	**0.708**	0.832	0.556	0.667	0.733	0.722	0.727
all-kernels	0.768	0.657	**0.708**	**0.834**	0.509	0.632	0.724	0.747	0.735
electra-clf	0.765	0.657	0.707	-	-	-	0.708	0.780	0.742
multi-modal-clf	**0.830**	0.527	0.645	-	-	-	0.730	0.780	**0.754**

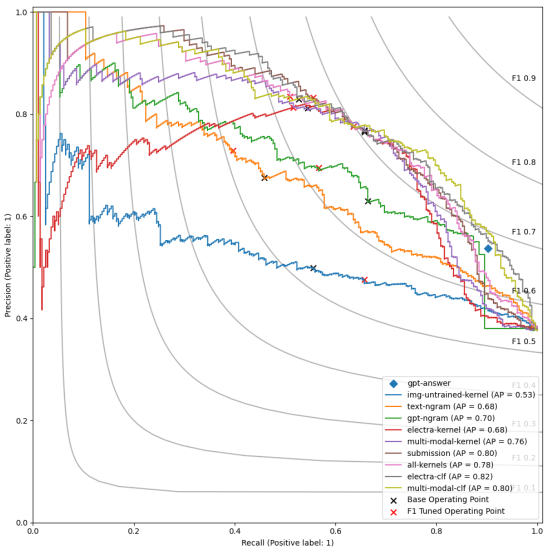

Figure 1: Precision Recall Curves for all Systems computed on D_{test}

noticed this during development, where system performance varied greatly between D_{dev} and $D_{dev-test}$, and therefore we chose the default classification threshold.

Finally, in Table 4 we also include the scores that could be achieved if we had access to the ideal threshold. We computed it by selecting the threshold which maximizes the F1 score on D_{test}. Of course, in reality one never has access to this knowledge, but we include it here to show how much influence threshold selection can have on the system comparison.

In Figure 1 we can also see that the curve for *submission* lies above the individual kernel based systems over the most recall values. Meaning that for most fixed recalls it achieves higher precision. This indicates that our ensembling method indeed yields an improved classifier. On the other hand, we can also see that *electra-clf* and *multi-modal-clf* perform even better.

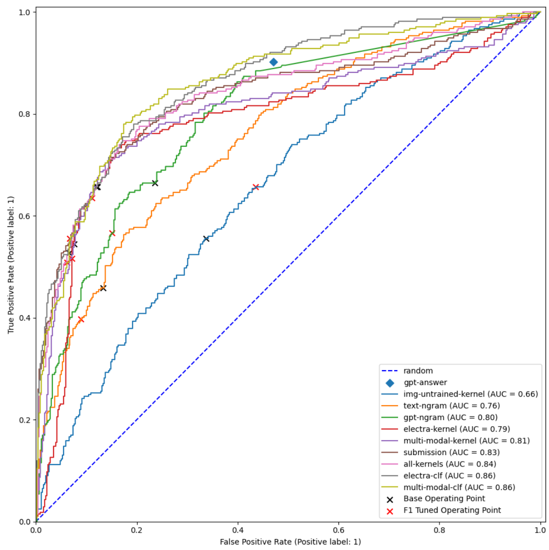

Figure 2: ROC Curves for all Systems computed on D_{test}

In Figure 2 we show the Receiver Operating Characteristic (ROC) curves for all our systems. We can again see that *electra-clf* and *multi-modal-clf* have the highest area under the curve (AUC), meaning that for most fixed false positive rates they have a higher true positive rate than other systems. We can also see that our ensembling method outperforms individual kernel methods.

5. Conclusion

We have laid out our solution to the *CheckThat! Lab 2023* sub-task 1A on multi-modal check-worthiness classification. Our solution includes diverse components that we combine using a multiple kernel learning approach. Our submission achieved second place out of 7 teams with an F1 score of 0.708. While analysing our results, we noted that the performance measure can vary drastically based on the selected decision threshold. When considering threshold-free methods such as ROC and PR curves, we find that our ensemble indeed seems to perform better than its individual components. Nevertheless, we note that the directly fine-tuned models outperform our submission under this lens. The performance gap between *electra-clf* and *electra-kernel* as well as *multi-modal-clf* and *multi-modal-kernel* is an open question requiring further study.

Acknowledgments

This work has been funded by the Hamison project supported by the EU ERA-Net CHIST-ERA; the Swiss National Science Foundation [20CH21_209672].

References

[1] A. Barrón-Cedeño, F. Alam, T. Caselli, G. Da San Martino, T. Elsayed, A. Galassi, F. Haouari, F. Ruggeri, J. M. Struß, R. N. Nandi, G. S. Cheema, D. Azizov, P. Nakov, The clef-2023 checkthat! lab: Checkworthiness, subjectivity, political bias, factuality, and authority, in: J. Kamps, L. Goeuriot, F. Crestani, M. Maistro, H. Joho, B. Davis, C. Gurrin, U. Kruschwitz, A. Caputo (Eds.), Advances in Information Retrieval, Springer Nature Switzerland, Cham, 2023, pp. 506–517.

[2] P. Nakov, A. Barrón-Cedeño, G. Da San Martino, F. Alam, R. Míguez, T. Caselli, M. Kutlu, W. Zaghouani, C. Li, S. Shaar, H. Mubarak, A. Nikolov, Y. S. Kartal, J. Beltrán, Overview of the CLEF-2022 CheckThat! lab task 1 on identifying relevant claims in tweets, in: Working Notes of CLEF 2022—Conference and Labs of the Evaluation Forum, CLEF '2022, Bologna, Italy, 2022.

[3] S. Shaar, M. Hasanain, B. Hamdan, Z. S. Ali, F. Haouari, A. Nikolov, M. Kutlu, Y. S. Kartal, F. Alam, G. Da San Martino, A. Barrón-Cedeño, R. Míguez, J. Beltrán, T. Elsayed, P. Nakov, Overview of the CLEF-2021 CheckThat! lab task 1 on check-worthiness estimation in tweets and political debates, 2021.

[4] S. Shaar, A. Nikolov, N. Babulkov, F. Alam, A. Barrón-Cedeño, T. Elsayed, M. Hasanain, R. Suwaileh, F. Haouari, G. Da San Martino, P. Nakov, Overview of CheckThat! 2020 English: Automatic identification and verification of claims in social media, CEUR Workshop Proceedings, 2020.

[5] P. Atanasova, P. Nakov, G. Karadzhov, M. Mohtarami, G. Da San Martino, Overview of the CLEF-2019 CheckThat! lab on automatic identification and verification of claims. Task 1: Check-worthiness, CEUR Workshop Proceedings, 2019.

[6] P. Atanasova, L. Marquez, A. Barrón-Cedeño, T. Elsayed, R. Suwaileh, W. Zaghouani, S. Kyuchukov, G. Da San Martino, P. Nakov, Overview of the CLEF-2018 CheckThat! lab on automatic identification and verification of political claims. Task 1: Check-worthiness, CEUR Workshop Proceedings, 2018.

[7] L. Derczynski, K. Bontcheva, M. Liakata, R. Procter, G. Wong Sak Hoi, A. Zubiaga, SemEval-2017 task 8: RumourEval: Determining rumour veracity and support for rumours, in: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), Association for Computational Linguistics, Vancouver, Canada, 2017, pp. 69–76. URL: https://aclanthology.org/S17-2006. doi:10.18653/v1/S17-2006.

[8] G. Gorrell, E. Kochkina, M. Liakata, A. Aker, A. Zubiaga, K. Bontcheva, L. Derczynski, SemEval-2019 task 7: RumourEval, determining rumour veracity and support for rumours, in: Proceedings of the 13th International Workshop on Semantic Evaluation, Association for Computational Linguistics, Minneapolis, Minnesota, USA, 2019, pp. 845–854. URL: https://aclanthology.org/S19-2147. doi:10.18653/v1/S19-2147.

[9] R. Aly, Z. Guo, M. S. Schlichtkrull, J. Thorne, A. Vlachos, C. Christodoulopoulos, O. Cocarascu, A. Mittal, The fact extraction and VERification over unstructured and structured information (FEVEROUS) shared task, in: Proceedings of the Fourth Workshop on Fact Extraction and VERification (FEVER), Association for Computational Linguistics, Dominican Republic, 2021, pp. 1–13. URL: https://aclanthology.org/2021.fever-1.1. doi:10.18653/v1/2021.fever-1.1.

[10] N. Hassan, C. Li, M. Tremayne, Detecting check-worthy factual claims in presidential debates, in: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, CIKM '15, Association for Computing Machinery, New York, NY, USA, 2015, p. 1835–1838. URL: https://doi.org/10.1145/2806416.2806652. doi:10.1145/2806416.2806652.

[11] I. Jaradat, P. Gencheva, A. Barrón-Cedeño, L. Màrquez, P. Nakov, ClaimRank: Detecting check-worthy claims in Arabic and English, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations, Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 26–30. URL: https://aclanthology.org/N18-5006. doi:10.18653/v1/N18-5006.

[12] A. Savchev, AI Rational at CheckThat! 2022: using transformer models for tweet classification, in: N. Faggioli, Guglielmo andd Ferro, A. Hanbury, M. Potthast (Eds.), Working Notes of CLEF 2022 - Conference and Labs of the Evaluation Forum, CLEF '2022, Bologna, Italy, 2022.

[13] R. M. Buliga Nicu, Zorros at CheckThat! 2022: ensemble model for identifying relevant claims in tweets, in: N. Faggioli, Guglielmo andd Ferro, A. Hanbury, M. Potthast (Eds.), Working Notes of CLEF 2022 - Conference and Labs of the Evaluation Forum, CLEF '2022, Bologna, Italy, 2022.

[14] 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://aclanthology.org/N19-1423. doi:10.18653/v1/N19-1423.

[15] 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, 2019. arXiv:1907.11692.

[16] D. Kiela, H. Firooz, A. Mohan, V. Goswami, A. Singh, P. Ringshia, D. Testuggine, The hateful memes challenge: Detecting hate speech in multimodal memes, in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems, volume 33, Curran Associates, Inc., 2020, pp. 2611–2624. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/1b84c4cee2b8b3d823b30e2d604b1878-Paper.pdf.

[17] F. Alam, S. Cresci, T. Chakraborty, F. Silvestri, D. Dimitrov, G. D. S. Martino, S. Shaar, H. Firooz, P. Nakov, A survey on multimodal disinformation detection, in: Proceedings of the 29th International Conference on Computational Linguistics, International Committee on Computational Linguistics, Gyeongju, Republic of Korea, 2022, pp. 6625–6643. URL: https://aclanthology.org/2022.coling-1.576.

[18] G. S. Cheema, S. Hakimov, A. Sittar, E. Müller-Budack, C. Otto, R. Ewerth, MM-claims: A dataset for multimodal claim detection in social media, in: Findings of the Association for Computational Linguistics: NAACL 2022, Association for Computational Linguistics, Seattle, United States, 2022, pp. 962–979. URL: https://aclanthology.org/2022.findings-naacl.72. doi:10.18653/v1/2022.findings-naacl.72.

[19] K. D. N., A. Patil, Multimodal Emotion Recognition Using Cross-Modal Attention and 1D Convolutional Neural Networks, in: Proc. Interspeech 2020, 2020, pp. 4243–4247. doi:10.21437/Interspeech.2020-1190.

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, I. Polosukhin, Attention is all you need, in: I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems, volume 30, Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[21] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language models are few-shot learners, CoRR abs/2005.14165 (2020). URL: https://arxiv.org/abs/2005.14165. arXiv:2005.14165.

[22] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, G. Lample, Llama: Open and efficient foundation language models, 2023. arXiv:2302.13971.

[23] M. Gönen, E. Alpaydın, Multiple kernel learning algorithms, J. Mach. Learn. Res. 12 (2011) 2211–2268.

[24] F. Alam, A. Barrón-Cedeño, G. S. Cheema, S. Hakimov, M. Hasanain, C. Li, R. Míguez, H. Mubarak, G. K. Shahi, W. Zaghouani, P. Nakov, Overview of the CLEF-2023 CheckThat! lab task 1 on check-worthiness in multimodal and multigenre content, in: M. Aliannejadi, G. Faggioli, N. Ferro, Vlachos, Michalis (Eds.), Working Notes of CLEF 2023–Conference and Labs of the Evaluation Forum, CLEF 2023, Thessaloniki, Greece, 2023.

[25] C. D. Manning, P. Raghavan, H. Schütze, Introduction to Information Retrieval, Cambridge University Press, 2008. URL: https://nlp.stanford.edu/IR-book/.

[26] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.

[27] C. Cortes, V. N. Vapnik, Support-vector networks, Machine Learning 20 (1995) 273–297.

[28] G. King, L. Zeng, Logistic regression in rare events data, Political Analysis 9 (2001) 137–163.

[29] K. Clark, M. Luong, Q. V. Le, C. D. Manning, ELECTRA: pre-training text encoders as discriminators rather than generators, in: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020, OpenReview.net, 2020. URL: https://openreview.net/forum?id=r1xMH1BtvB.

[30] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Brew, Huggingface's transformers: State-of-the-art natural language processing, CoRR abs/1910.03771 (2019). URL: http://arxiv.org/abs/1910.03771. arXiv:1910.03771.

[31] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, in: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019, OpenReview.net, 2019. URL: https://openreview.net/forum?id=Bkg6RiCqY7.

[32] F. Barbieri, J. Camacho-Collados, L. Espinosa Anke, L. Neves, TweetEval: Unified benchmark and comparative evaluation for tweet classification, in: Findings of the Association for Computational Linguistics: EMNLP 2020, Association for Computational Linguis-

tics, Online, 2020, pp. 1644–1650. URL: https://aclanthology.org/2020.findings-emnlp.148. doi:10.18653/v1/2020.findings-emnlp.148.

[33] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, An image is worth 16x16 words: Transformers for image recognition at scale, CoRR abs/2010.11929 (2020). URL: https://arxiv.org/abs/2010.11929. arXiv:2010.11929.

[34] T. Ridnik, E. Ben-Baruch, A. Noy, L. Zelnik-Manor, Imagenet-21k pretraining for the masses, 2021. arXiv:2104.10972.

[35] J. Wei, X. Wang, D. Schuurmans, M. Bosma, E. H. Chi, Q. Le, D. Zhou, Chain of thought prompting elicits reasoning in large language models, CoRR abs/2201.11903 (2022). URL: https://arxiv.org/abs/2201.11903. arXiv:2201.11903.

[36] L. Beurer-Kellner, M. Fischer, M. Vechev, Prompting is programming: A query language for large language models, PLDI '23 (2022).