



This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No. 101135437



AI services for COntinuous trust
in emerging Digital Environments

D2.2

Report on the assessment of the role of novel text and image generative AI methodologies in producing and combating disinformation (Revision 1)

Grant Agreement no.:	101135437
Project Title:	AI-CODE - AI services for COntinuous trust in emerging Digital Environments
Contractual Due date:	30/11/2024
Actual submission date:	29/11/2024
	29/05/2025 (Revision 1)
Lead beneficiary:	CERTH

Deliverable number	D2.2
Deliverable Title:	Report on the assessment of the role of novel text and image generative AI methodologies in producing and combating disinformation
Description:	This deliverable summarises the research performed in T2.2 and T2.3 and describes positive as well as negative consequences of generative AI adopted in social media.
Type/Nature	☒ R - Document, report ☐ DEM - <i>Demonstrator, pilot, prototype</i> ☐ DEC - <i>Websites, patent filings, videos, etc.</i> ☐ OTHER ☐ ETHICS – <i>Ethics requirement</i> ☐ ORDP – <i>Open Research Data Pilot</i>
Dissemination Level	☒ Public – fully open ☐ Sensitive — limited under the conditions of the Grant Agreement ☐ EU classified — RESTREINT-UE/EU-RESTRICTED, CONFIDENTIEL-UE/EU-CONFIDENTIAL, SECRET-UE/EU-SECRET under Decision 2015/444. For items classified under other rules (e.g. national or international organization), please select the equivalent EU classification level.
Work Package number	WP2
Work Package leader	KInIT
Author(s) <i>in alphabetical order</i>	Randi Cecchine (RU) Marco Guerini (FBK) Ilias Kanellos (ATC) Dominik Macko (KInIT) Tina Maljur (EIPCM) Róbert Móro (KInIT) Symeon Papadopoulos (CERTH) Fariba Sadeghi (FBK) Manos Schinas (CERTH) Ivan Srba (KInIT) Sara Tonelli (FBK) Eleni Tsalapati (ATC)
Internal Reviewers <i>in alphabetical order</i>	Martha Larson (RU) Marco Rovera (FBK)

Authors (in alphabetical order)

Name	Partner	e-mail
Randi Cecchine	RU	randi.cecchine@ru.nl
Marco Guerini	FBK	m.guerini@fbk.eu
Ilias Kanellos	ATC	i.kanellos@atc.gr
Dominik Macko	KInIT	dominik.macko@kinit.sk
Tina Maljur	EIPCM	t.maljur@eipcm.org
Róbert Móro	KInIT	robert.moro@kinit.sk
Symeon Papadopoulos	CERTH	papadop@iti.gr
Fariba Sadeghi	FBK	fsadeghi@fbk.eu
Manos Schinas	CERTH	manosetro@iti.gr
Ivan Srba	KInIT	ivan.srba@kinit.sk
Sara Tonelli	FBK	satonelli@fbk.eu
Eleni Tsalapati	ATC	e.tsalapati@atc.gr

History of changes

Version	Date	Changes
0.1	11/09/2024	Added Table of Content
0.2	11/11/2024	Finalized drafts of all sections
0.3	18/11/2024	Comments and suggestions by internal reviewers added
0.4	22/11/2024	Feedback from internal reviewers integrated
0.5	25/11/2024	Review by CERTH
0.6	29/11/2024	Review by the coordinator
1.0	29/11/2024	Final version prepared
1.1	21/05/2025	Discussion (Section 6) added reflecting the comments and suggestions of the external expert reviewers
1.2	28/05/2025	Comments and suggestions by internal reviewers added
1.3	28/05/2025	Feedback from internal reviewers integrated
1.4	29/05/2025	Revised version prepared
2.0	29/05/2025	Final Revised Version

Table of contents

Executive Summary.....	9
1 Introduction	10
2 Assessment of the Potential of Text Generative AI to Generate and to Detect Disinformation	11
2.1 Background	11
2.2 Generation of disinformation: A case study in Greek language	13
2.2.1 Implementation details.....	14
2.2.2 Fake narrative collection.....	14
2.2.3 Fake news generation.....	16
2.2.4 Examining generated texts using LLM-as-a-judge.....	17
2.2.5 Conclusion	18
2.3 Detection of machine-generated text.....	19
2.3.1 LLM2S3 MGT detection system	19
2.3.2 Adversarial robustness.....	19
2.3.3 Social media.....	20
2.3.4 Conclusion	22
2.4 Logical fallacy detection	22
2.4.1 Logical fallacy detection in news articles (English)	22
2.4.2 Logical fallacy detection on social media (Italian).....	25
2.4.3 Conclusion	31
2.5 Check-worthy claim detection and matching.....	31
2.5.1 Claim detection in news.....	31
2.5.2 Claim matching.....	33
2.5.3 Check-worthy claim detection on social media data	35
2.5.4 Conclusion	37
2.6 Adapting LLMs for low-resource text classification	37
2.6.1 Comparison of specialised small language models vs. general LLMs	38
2.6.2 Advanced sample selection strategies for few-shot learning.....	39
2.6.3 LLM-based text augmentation.....	39
2.6.4 Combining sample selection and data augmentation.....	40
2.6.5 Conclusion	40
3 Assessment of the Potential of Text Generative AI for Trusted Information Production	40
3.1 Background	41
3.1.1 Insights into the workflow of media professionals	41
3.1.2 Allowing media professionals to use Generative AI within media organizations’ guidelines ..	42
3.1.3 Verdict production for fact checking	44
3.2 Fact-checking verdict production as a case of multilingual paraphrasing and summarization	46
3.2.1 Datasets.....	46
3.2.2 Experimental setup	47

3.2.3	Conclusion	53
3.3	Paraphrasing and the notion of authorship.....	53
3.3.1	Motivation and experimental setup.....	53
3.3.2	Conclusion	54
4	Assessment of AI to Detect Synthetic Image Disinformation	54
4.1	Background	54
4.1.1	Synthetic image generation.....	54
4.1.2	Synthetic image detection.....	55
4.2	Benchmarking framework.....	56
4.2.1	Evaluation data	56
4.2.2	Integrated models.....	57
4.2.3	Evaluation metrics	58
4.3	Evaluation and benchmarking results	59
4.3.1	Overall performance on GAN and diffusion images	59
4.3.2	Performance on Synthbuster.....	61
4.3.3	Exploring the influence of image resolution on model performance.....	63
4.3.4	Influence of image transformations	64
4.4	Conclusion	65
5	Risk Assessment of Image Generative AI.....	65
5.1	Popularity and adoption of GitHub projects	66
5.2	Code quality of GitHub projects	67
5.3	Assessment of image quality.....	70
5.4	Conclusion	76
6	Discussion.....	77
6.1	Assessment of the potential of text generative AI to generate and to detect disinformation	77
6.2	Assessment of the potential of text generative AI for trusted information production	78
6.3	Assessment of AI to detect synthetic image disinformation	79
6.4	Risk assessment of generative AI	79
7	Conclusions	80

Table of figures

Figure 1: Fake Narrative Title Extraction Process.	15
Figure 2: LLM2S3 system components overview.....	19
Figure 3: Taxonomy of fallacy types that we designed for evaluation purposes. Labels below the root (i.e., Insufficient proof, Simplification, and Distraction) represent the fallacy macro-categories.....	26
Figure 4: Prompt template with the different variables tested in the experiments. \$TASK_DESC refer to post- or span-level classification. \$FALLACIES+\$MACROCAT refers to the list of fallacies with description and the corresponding macro category. \$INSTR refers to the instructions used to describe the task types.	29
Figure 5: Analysis of the outputs generated by LLMs on SPAN tasks according to whether they contain an	

answer, extra instructions, both, or an empty response (left) and analysis of the answers generated by LLMs (i.e., based on answer+both) on SPAN tasks according to whether the output format is ok (format ok), provides explanations, wrong labels, or repetitions (repeat).	31
Figure 6: The nodes of the implemented graph of the claim matching model.	34
Figure 7: Paraphrased claims matched to database with debunked claims.	35
Figure 8: Comparison between the performance of specialised small and general large language models. The break-even points are identified by observing the impact of changing the number of available labelled samples and taking performance variance into consideration. The results are aggregated across the language models relevant for each approach and across 8 datasets. Specialised models outperform general ones with only few labelled samples (up to 100), with performance variance showing strong impact on the comparison, increasing the number significantly.....	38
Figure 9: Overview of our methodology. For each dataset, we randomly sample 6 samples per label that are used as seed sentences for LLM data augmentation. There, we collect data in 2 rounds - 1st only using the prompt method and then in parallel for prompt method and 3 different diversity incentive methods. These are added together to form the datasets. BERT-large or Mistral classifier is fine-tuned 5 or 3 times respectively on each of the collected data and then evaluated. We repeat the entire process 5 times.....	40
Figure 10: ACC, TPR and TNR across GAN and Latent Diffusion datasets, for the top 3 performing models.....	60
Figure 11: Accuracy with the optimal threshold per model compared to accuracy achieved with the default threshold (0.5). Bars present the average accuracy while error bars indicate the variance of each model across datasets. All models, except GramNet, have been trained on ProGAN images.	60
Figure 12: Comparing accuracy across cropped and resized images in the Synthbuster dataset.	64
Figure 13: ACC, TPR and TNR across detection models for different image transformations. Metrics have been averaged across.	65
Figure 14: Overall project rating and metrics assessment for the StyleGAN3 repository.....	69
Figure 15: Some code quality key indicators for the StyleGAN3 repository.	69
Figure 16: FID score for the 11 generative models used for evaluation. Lower values are considered better.....	72
Figure 17: KID score for the 11 generative models used for evaluation. As KID is calculated for each pair of real-synthetic images we report the average KID for all the 1600 pairs, alongside standard deviation. Lower values are considered better.	73
Figure 18: LPIPS score for the 11 generative models used for evaluation. As LPIPS is calculated for each pair of real-synthetic images we report the average LPIPS for all the 1600 pairs, alongside standard deviation. Lower values are considered better.	73
Figure 19: SSIM score for the eleven generative models used for evaluation. As SSIM is calculated for each pair of real-synthetic images we report the average SSIM for all the 1600 pairs, alongside standard deviation. Higher values are considered better.....	74
Figure 20: Examples of text-to-image generative models. Given the caption of the real image as provided by the Visual News dataset, the generative models create a synthetic image with the same content.	75
Figure 21: Examples of image-to-image generative models. The models that require also a text prompt the instruction was “replicate the content of the input image”.	75
Figure 22: Example of stable diffusion v1.5 in producing landscapes.	75
Figure 23: Generated synthetic diagram with optimal quality metrics.....	76

Table of tables

Table 1: Sample Narratives collected. The green background denotes the manually selected Narrative.	16
Table 2: Coherence of LLM judgements with prompt variation.	18

Table 3: LLM judgements per LLM author.	18
Table 4: Detection performance comparison of all MGT detection methods using all MULTITuDE test data (human and machine samples) and machine samples obfuscated by each AO method. Macro avg. F1-score optimal) is just a theoretical performance and refers to this metric after calibration of the classification threshold based on ROC curve maximizing difference between TPR (true positive rate) and FPR (false positive rate). Similarly, Macro avg. F1-score (1% FPR) and (5% FPR) refer to this metric for the classification threshold closest to (but below) 1% and 5% of FPR. For the exact calculations of such thresholds, see the published source code. F refers to fine-tuned, P to pre-trained, and S to statistical MGT detection methods category.	20
Table 5: MultiSocial text sample counts across languages and platforms for train and test split.	21
Table 6: Benchmark of the selected MGT detection methods of statistical (orange), pre-trained (blue), and fine-tuned (green) categories.	21
Table 7: Distribution of fallacy types in the dataset (Sourati et al., 2023).	23
Table 8: The most important hyperparameters used for fine-tuning.	24
Table 9: Evaluation results of Llama 2 (Llama 2 Base) and its fine-tuned versions (Llama 3 FT) in ADFD.	24
Table 10: Statistics related to the second dataset used for testing.	25
Table 11: Evaluation results of Llama 2 (Llama 2 Base) and its fine-tuned versions (Llama 3 FT) in the dataset of 12 articles.	25
Table 12: Statistics and per-class IAA scores (γ_{cat}) across all fallacy types. We report the number of spans and their average length (with standard deviation) at the token level considering all annotations.	27
Table 13: Test set results for POST and SPAN tasks at the coarse-grained (C) and fine-grained (F) level. We report (micro)average precision (P), recall (R), and F1 scores (w/ std dev) across $k = 5$ splits, averaged over all views. For SPAN-F, we also present scores using both strict and soft modes. Best results are in bold.	30
Table 14: Evaluation results for original model and the fine-tuned model using the various ROUGE metrics.	33
Table 15: Results for check-worthy claim detection in Italian across setups and models. We report average scores across the $k=5$ splits. We report only the best result after comparing the auxiliary loss weight of 0.1, 0.5 and 1.0 for each transformer model.	37
Table 16: An example of data entry from VerMouth dataset.	46
Table 17: Sample of prompts that used for Italian language.	48
Table 18: Example of Verdict Generation Using EN Prompt + Language Instructions with Llama 3 and Llama 3.1 from the Multilingual-VerMouth Dataset.	49
Table 19: RR on Multilingual-Vermouth Dataset.	52
Table 20: ROUGE on Multilingual-Vermouth Dataset.	52
Table 21: RR on AI4Trust Dataset.	52
Table 22: ROUGE on AI4Trust Dataset.	53
Table 23: Test datasets from Wang et al. and Ojha et al., used to evaluate state of the art detectors in this section.	57
Table 24: Categorizing the integrated models based on the type of input features, the backbone architecture used for the detector and the pretraining task of the backbone. Figure sourced from the original paper introducing SIDBench framework.	57
Table 25: Accuracy (ACC) scores of the integrated models across 20 test datasets from Wang et al. and Ojha et al.	59
Table 26: Average precision (AP) of integrated models across 20 test datasets from Wang et al. and Ojha et al.	59

Table 27: Accuracy (ACC) scores of integrated models for the Synthbuster dataset. 61

Table 28: Average Precision (AP) of integrated models for the Synthbuster dataset. 61

Table 29: ACC and AP for detection models trained on Latent Diffusion (LD) and StyleGAN2 training data.....63

Table 30: Top 20 most popular image generation models based on star ratings on GitHub.....67

Table 31: Average scores for the quality metrics, across image-to-image and text-to-image generative models.....74

Executive Summary

This deliverable presents an in-depth assessment of the role of novel text and image generative AI methodologies in producing information and disinformation and, especially, in combating disinformation. As generative AI continues to evolve, its potential to generate misleading information, but also to support trusted information production poses key ethical and operational challenges. This document provides a foundation for addressing those challenges.

The document first explores the assessment of text generative AI's capability to create and detect disinformation, evaluating its strengths and limitations in both generating misleading content and identifying it (Section 2). Section 3 extends this analysis to examine text generative AI's potential for supporting the production of trusted information, highlighting key areas where these models can support the creation of credible content. The report then addresses synthetic image disinformation, examining AI's efficacy in detecting altered or artificially generated images that could mislead or misinform (Section 4). Section 5 provides a risk assessment of image generative AI, focusing on the ease with which synthetic multimedia content can be created. This includes an analysis of open-source projects and models, as well as an evaluation of the quality and realism of the generated images. Discussion on how the evolution of generative AI impacts the open problems, challenges, and what further work is needed, is provided in Section 6. Finally, in Section 7, we summarize the achieved results and reflect on the impacts and open questions to be addressed in the upcoming deliverable D2.3.

1 Introduction

This deliverable summarizes the research performed in tasks *T2.2 Generative AI methods for text generation* and *T2.3 Trustworthy AI methods for synthetic media detection*. In both tasks, we focus on **generative AI**, which refers to algorithms that can automatically generate content in multiple modalities, including but not limited to images, videos, or text. Generative AI is also applied in classification tasks, such as making a decision about whether text or image content is factual or misleading.

We are currently witnessing a rapid development of generative AI with new, better models for individual modalities (e.g., text, images, video, speech) or multimodal ones (i.e., combining two or more modalities) being released on a regular basis by both large companies, such as OpenAI, Google, Microsoft, Meta, Amazon, Nvidia, Baidu, etc. as well as by the research community. Although these models have great potential due to their ability to perform many advanced tasks, how they can be adapted and used by media professionals in their work remains an open question. At the same time, their potential for misuse is clear due to the ability of generative AI to create realistic fake content that can be difficult to distinguish from real one. This can be used to create, spread, and amplify disinformation and propaganda, which can have a significant impact on public opinion, political processes, and social stability. Obvious examples include deepfake videos impersonating political leaders, celebrities, or other public figures or machine-generated text tailored to individual users' interests presented in their language spreading false narratives, disinformation, mistrust or inciting violence.

Thus, the goal of the two tasks was to **assess and research positive as well as negative consequences of generative AI**. More specifically, we aimed to:

1. Assess the potential of generative AI to a) generate textual disinformation narratives (T2.2) as well as b) create visual synthetic media, which can be highly convincing, but misleading.
2. Assess the current state of the art in detecting synthetic (machine-generated) textual (T2.2) as well as visual (T2.3) content and research new detection methods, exploring also the potential of generative AI to support existing detection approaches.
3. Investigate how large-scale language models can be beneficial for media professionals in trusted information production (T2.2).

The structure of the deliverable reflects these goals as well as the achieved results. Sections 2 and 3 focus on the text generative AI approaches, while the sections 4 and 5 focus on the image ones. More specifically:

- In Section 2, we **assess the potential of text generative AI to generate disinformation on one hand and to detect it on the other**. Building on our previous work and the current state of the art, we carried out a case study of the capabilities of generative AI to generate disinformation in the low-resourced Greek language. We found that, although none of the large language model (LLM) studied succeeded in producing completely error free convincing Greek text for all narratives, some produced texts that could be mistaken for human-written ones. Regarding the detection of machine-generated text, we present several novel results focusing on benchmark of existing state-of-the-art detection methods in highly multilingual settings (up to 22 different languages) in the social media domain as well as in adversarial conditions. We also contribute by developing novel detection methods and by creating large multilingual datasets for the task. Next, we investigate other disinformation-related detected tasks, namely fallacy detection and detection of check-worthy claims, in which we present novel research results, including new resources (datasets) for the tasks. Lastly, we examine strategies for adapting LLMs for low-resource text classification, from which all these detection tasks can directly benefit.
- In Section 3, we **assess the potential of text generative AI for trusted information production**. We start the section by discussing the workflow used by media professionals during the production of trusted information. Then we assess the capabilities of LLMs to paraphrase and summarize text, which can be used for various ends by the media professionals, e.g., to provide justification why a specific piece of information is false or to summarize the news article to be used in the social media context. We primarily focus on the former (generating reliable information and countering mis- and

disinformation). We also assess the LLMs' paraphrasing capabilities from the perspective of authorship of the generated text in various contexts.

- In Section 4, we focus on **the assessment of AI's effectiveness in detecting synthetic images** that can be used for disinformation purposes. The section begins with a background on synthetic image generation techniques and current methods for detecting such content. Then we introduce a benchmarking framework that details the datasets used for evaluation, the models integrated for detection, and the metrics applied to measure performance. In the evaluation phase, we report benchmarking results, specifically examining how image resolution and various image transformations impact model accuracy and detection reliability. This analysis provides valuable insights into optimizing detection models for synthetic image disinformation.
- In Section 5, we present a **risk assessment of image generative AI**. It begins with an analysis of the popularity and adoption of relevant GitHub projects, exploring how widely these open-source tools are used. Next, it assesses the code quality of these projects, examining factors that impact the reliability and usability of the models. Finally, the section includes an evaluation of the image quality produced by these generative AI models, with a focus on the realism of the generated content and its potential implications for creating synthetic multimedia.

In Section 6, we complement and synthesise previous assessments with a discussion on open problems and challenges. This look-ahead perspective serves as a source of directions for future work during the rest of WP2 as well as the rest of the whole project. Finally, in Section 7, we summarize the achieved results and discuss the impacts and open questions to be addressed in the upcoming deliverable D2.3.

Note that some of the results presented in this deliverable have already been published. In those cases, we provide the reference and the link to the publication at the beginning of a respective (sub-)section and provide a summary of most important findings. In other cases which are yet to be published, we provide more detailed information on the experimental methodology and results.

2 Assessment of the Potential of Text Generative AI to Generate and to Detect Disinformation

This section reports on assessment and experiments carried out in T2.2 *Generative AI methods for text generation*. In the first part of this task, we focused on the potential of generative AI to be misused for generation of textual disinformation. In the second part, we focused on a complementary view how generative AI can be used to make detection of disinformation more effective and accurate.

2.1 Background

Generative Large Language Models (LLMs) are capable of generating human-like text in response to a user's prompt in various languages (with a different degree of coherence and grammatical correctness). Among their distinctive features belongs also their ability to perform various textual (downstream) tasks, such as text summarization, translation, question answering, or logical reasoning (see, e.g. Wei et al., 2022¹) in a zero-shot or a few-shot setting (called also in-context learning). Although it is computationally expensive to train a foundational LLM from scratch, it is now relatively easy and inexpensive to adapt (fine-tune) an LLM, using various methods, such as full fine-tuning or parameter-efficient approaches. Therefore, besides commercially available models accessible only via API, such as ChatGPT and GPT-4o by OpenAI, Claude by Anthropic and others, there is available a wide range of LLMs of various sizes and pre-training; currently, there are more than 148,000 text generation models available at a popular machine learning platform HuggingFace².

¹ Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, William Fedus. 2022. Emergent Abilities of Large Language Models. arXiv:2206.07682. <https://arxiv.org/abs/2206.07682>

² https://huggingface.co/models?pipeline_tag=text-generation&sort=downloads

Because of their current capabilities, wide adoption and relative ease of adaptation, the potential has been recognised for LLMs to be misused for a range of harmful ends, including influence operations (Goldstein et al., 2023³), spam or unethical authorship (Crothers et al., 2023⁴), disinformation generation (Buchanan et al., 2021⁵; Lucas et al., 2023⁶; Vykopal et al., 2024⁷). Regarding disinformation generation, the work of Vykopal et al., 2024 studied the disinformation capabilities of 10 LLMs to generate false news articles in English language for a defined set of 20 disinformation narratives, evaluating aspects such as how well the LLMs are at generating news articles, how strongly they tend to agree or disagree with the disinformation narratives, how often they generate safety warnings, etc. Their main conclusion was that LLMs are able to generate convincing news articles that agree with dangerous disinformation narratives.

On the other hand, there have been voices tempering the fears of massive misuse of LLMs for disinformation generation, most notably (Simon et al., 2023⁸), who argued that generative AI by itself will not increase the demand for misinformation or that better quality of produced disinformation might not directly reflect to its perceived higher credibility as this is often influenced by other factors (such as partisanship). In any case, actual field studies are missing (especially for languages other than English) and the rapid advancement of the new models makes the misuse of LLMs to generate and/or amplify disinformation increasingly easier. Personalization of generated disinformation potentially enabled by the LLMs is another largely not yet explored topic. We address these open problems by providing a case study of disinformation generation in Greek language (see Section 2.2).

As a consequence of increased quality of machine-generated text, the task of machine-generated text detection has recently become more prominent (Uchendu et al., 2023⁹). Lately, several new datasets have been released, which have extended the task to a range of languages and domains (Macko et al., 2023¹⁰; Wang et al., 2024¹¹). Regarding the detection methods and models, these range from statistical to fine-tuned language models, including fine-tuned LLMs or their combinations. Despite the recent progress in the task, several challenges remain, namely robustness of detectors to the out of distribution data and adversarial attacks, interpretability and explainability of the detection, detection of short social media texts or detection of not purely generated texts. We address some of these issues by proposing a more robust machine-generated text detector, exploring the adversarial robustness of detection and benchmarking the detectors on texts from social media (see Section 2.3).

In the second part of the task T2.2, we focused on ways in which generative textual LLMs can be used to make detection of disinformation more effective and accurate, building on their capability to perform various downstream tasks. We specifically focused on two credibility signals that are often associated and are thus relevant for disinformation detection, namely *fallacy detection* and *check-worthy claim detection*.

³ Josh A. Goldstein, Girish Sastry, Micah Musser, Renee DiResta, Matthew Gentzel, Katerina Sedova. 2023. Generative Language Models and Automated Influence Operations: Emerging Threats and Potential Mitigations. arXiv:2301.04246. <https://arxiv.org/abs/2301.04246>

⁴ E. N. Crothers, N. Japkowicz and H. L. Viktor. 2023. Machine-Generated Text: A Comprehensive Survey of Threat Models and Detection Methods. In IEEE Access, vol. 11, pp. 70977–71002, doi: 10.1109/ACCESS.2023.3294090. <https://ieeexplore.ieee.org/document/10177704>

⁵ Ben Buchanan, Andrew Lohn, Micah Musser, and Katerina Sedova. 2021. Truth, Lies, and Automation. How Language Models Could Change Disinformation. Report. <https://cset.georgetown.edu/publication/truth-lies-and-automation/>

⁶ Jason Lucas, Adaku Uchendu, Michiharu Yamashita, Jooyoung Lee, Shaurya Rohatgi, and Dongwon Lee. 2023. Fighting Fire with Fire: The Dual Role of LLMs in Crafting and Detecting Elusive Disinformation. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pages 14279–14305, Singapore. Association for Computational Linguistics.

⁷ Ivan Vykopal, Matúš Pikuliak, Ivan Srba, Robert Moro, Dominik Macko, and Maria Bielikova. 2024. Disinformation Capabilities of Large Language Models. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 14830–14847, Bangkok, Thailand. Association for Computational Linguistics. <https://aclanthology.org/2024.acl-long.793/>

⁸ Felix M Simon, Sacha Altay, and Hugo Mercier. 2023. Misinformation reloaded? Fears about the impact of generative AI on misinformation are overblown. Harvard Kennedy School Misinformation Review 4, 5 (2023). <https://misinforeview.hks.harvard.edu/article/misinformation-reloaded-fears-about-the-impact-of-generative-ai-on-misinformation-are-overblown/>

⁹ Adaku Uchendu, Thai Le, and Dongwon Lee. 2023. Attribution and Obfuscation of Neural Text Authorship: A Data Mining Perspective. SIGKDD Explor. Newsl. 25, 1 (July 2023), 1–18. <https://doi.org/10.1145/3606274.3606276>

¹⁰ Dominik Macko, Robert Moro, Adaku Uchendu, Jason Lucas, Michiharu Yamashita, Matúš Pikuliak, Ivan Srba, Thai Le, Dongwon Lee, Jakub Simko, and Maria Bielikova. 2023. MULTITUDE: Large-Scale Multilingual Machine-Generated Text Detection Benchmark. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 9960–9987. <https://doi.org/10.18653/v1/2023.emnlp-main.616>

¹¹ Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Alham Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024. M4GT-Bench: Evaluation Benchmark for Black-Box Machine-Generated Text Detection. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 3964–3992, Bangkok, Thailand. Association for Computational Linguistics. <https://aclanthology.org/2024.acl-long.218/>

Fallacy detection (i.e., detection of errors in arguments) has recently been addressed in various data challenges. Note that in some works like the SemEval 2023 shared task 3¹² fallacy detection is referred to as persuasion technique detection. Despite these efforts, detection of fallacies is still an open challenge also due to limited resources available. Therefore, we contribute by providing two novel fallacy detection datasets (see Section 2.4).

Check-worthy claim detection aims to detect claims that are both verifiable and are important to be fact-checked, i.e., have a priority (Panchendrarajan and Zubiaga, 2024¹³). Verifiability refers to the possibility of determining the veracity of a claim. The importance or priority refers to the fact that since verifying all statements about the world is impossible, it is important to prioritize claims which are considered timely, interesting to the public and whose verification might have a broader impact (Das, 2023¹⁴). It is often combined with previously fact-checked claim retrieval (see, e.g., Pikuliak et al., 2023¹⁵), sometimes also denoted as claim matching, which aims to check whether a given claim was already verified before by searching in existing repositories of fact-checked claims. We contribute to this work in the AI-CODE project with two methods (see Section 2.5): the first one detects check-worthiness in news (in English language) based on combination of claim detection and claim matching steps using an LLM and a RAG (retrieval-augmented generation) system; the other detects check-worthiness in social media data employing a fine-tuned specialized transformer model focusing on Italian language.

Lastly, generative AI is uniquely positioned to deal with the problem of low-resource settings due to its ability to harness internal knowledge, learned during training, and transfer it, applying it to new tasks/problems via different techniques, such as in-context learning or domain adaptation through fine-tuning, prompt-tuning or instruction-tuning (Dong et al., 2024¹⁶; Liu et al., 2023¹⁷). LLMs have also recently been used to augment textual datasets (especially in low-resource settings), e.g., using paraphrasing of existing samples (Piedboeuf and Langlais, 2023¹⁸; Cegin et al., 2023¹⁹). We investigate these approaches in Section 2.6.

2.2 Generation of disinformation: A case study in Greek language

With the advent of LLMs and their availability as chat-based services, the public can now easily generate large amounts of text. A major risk associated with this ability is the production and spread of fake news, which can be used to manipulate the public. While LLMs often have guardrails, which aim to minimize such risks, completely controlling the LLM is hard, and previous works have shown that LLMs can produce convincing fake news articles^{20,21}.

Since LLMs are trained on large corpora of online text, which is predominantly English, their ability to produce good quality English text is well-established. However, it is uncertain how effective they are in low-resource languages, such as Greek, where the data available is substantially smaller. Their reduced capability to produce

¹² <https://propaganda.math.unipd.it/semeval2023task3/>

¹³ Rubaa Panchendrarajan and Arkaitz Zubiaga. 2024. Claim detection for automated fact-checking: A survey on monolingual, multilingual and cross-lingual research. *Natural Language Processing Journal* 7 (2024), 100066. <https://doi.org/10.1016/j.nlp.2024.100066>

¹⁴ Anubrata Das, Houjiang Liu, Venelin Kovatchev, and Matthew Lease. 2023. The state of human-centered NLP technology for fact-checking. *Information Processing & Management* 60, 2 (2023), 103219. <https://doi.org/10.1016/j.ipm.2022.103219>

¹⁵ Matúš Pikuliak, Ivan Srba, Robert Moro, Timo Hromadka, Timotej Smoleň, Martin Melišek, Ivan Vykopal, Jakub Simko, Juraj Podroužek, and Maria Bielikova. 2023. Multilingual Previously Fact-Checked Claim Retrieval. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16477–16500, Singapore. Association for Computational Linguistics. <https://aclanthology.org/2023.emnlp-main.1027/>

¹⁶ Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Tianyu Liu, Baobao Chang, Xu Sun, Lei Li, Zhifang Sui. 2024. A Survey on In-context Learning. *arXiv:2301.00234*. <https://arxiv.org/abs/2301.00234>

¹⁷ Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. *ACM Comput. Surv.* 55, 9, Article 195 (September 2023), 35 pages. <https://doi.org/10.1145/3560815>

¹⁸ Frédéric Piedboeuf and Philippe Langlais. 2023. Is ChatGPT the ultimate Data Augmentation Algorithm?. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15606–15615, Singapore. Association for Computational Linguistics. <https://aclanthology.org/2023.findings-emnlp.1044/>

¹⁹ Jan Cegin, Jakub Simko, and Peter Brusilovsky. 2023. ChatGPT to Replace Crowdsourcing of Paraphrases for Intent Classification: Higher Diversity and Comparable Model Robustness. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1889–1905, Singapore. Association for Computational Linguistics. <https://aclanthology.org/2023.emnlp-main.117/>

²⁰ Ivan Vykopal, Matúš Pikuliak, Ivan Srba, Robert Moro, Dominik Macko, and Maria Bielikova. 2024. Disinformation Capabilities of Large Language Models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14830–14847, Bangkok, Thailand. Association for Computational Linguistics. <https://aclanthology.org/2024.acl-long.793/>

²¹ Heppell, F., Bakir, M. E., & Bontcheva, K. (2024). Lying Blindly: Bypassing ChatGPT's Safeguards to Generate Hard-to-Detect Disinformation Claims at Scale. *arXiv preprint arXiv:2402.08467*.

high-quality low-resource language texts is witnessed by the need for fine-tuning them. Examples of fine-tuned LLMs for this purpose include Aguila for Spanish (*Projecte-aina/Aguila-7b*²²), LeoLM for German (*LEOLM: Igniting German-Language LLM Research*²³), and Meltemi for Greek (*Ilsp/Meltemi-7B-Instruct-v1*²⁴). An open issue, hence, is the risk associated with the generation of fake news in such languages. The aim of this research, carried out by ATC, is to assess the potential of Large Language Models to generate disinformation in Greek. We extract fake narratives from Greek articles that have been debunked and produce texts containing fake, or misleading information written in Greek. Our examination focuses on:

- Examining the texts produced in terms of the quality of the language, the characteristics of the texts per LLM, and the compliance of the LLMs in producing fake information.
- Employing the LLM as a Judge approach to examine whether LLMs are a viable option for providing reliable judgements about the content of Greek texts.

2.2.1 Implementation details

Our implementations employ the following LLMs, listed along with their providers:

- GPT-3.5-turbo, GPT-4, and GPT-4o-mini – access through OpenAI’s API.
- Llama3-8b-8192 (henceforth, Llama3 small), Llama3-70b-8192 (henceforth Llama3 large), Gemma-7b-it (henceforth Gemma), Mixtral-8x7b-32768 (henceforth Mixtral) - access programmatically through Groq’s API²⁵.
- ilsp/Meltemi-7B-Instruct-v1 (henceforth Meltemi) - run on Google colab pro, downloaded from HuggingFace.

All LLMs were configured to produce text with a temperature of 0.0 to enforce maximum determinism given a fixed prompt. For Mixtral, an additional setting of maximum token length was enforced (max_tokens = 4800), to conform with Groq’s API limits. Further, after experimentation we noticed that Meltemi tended to produce infinitely repeating text, which was mitigated by setting the repetition penalty to 1.15. All code was written in Python. LLM-based operations use the LangChain framework. Web document parsing relied on Python’s requests and BeautifulSoup libraries. All RAG-based implementations make use of the FAISS vector store implementation by LangChain.

2.2.2 Fake narrative collection

Our work focuses on examining LLM-generated fake news in Greek, hence we require a set of fake narratives based on which to produce these texts. We chose to automatically generate one using the Greek Fact Check website of AFP²⁶. Note, that we decided not to reuse narratives from previous work (e.g., Vykopal et al. (2024)²⁷, Heppell (2024)²⁸), to avoid limiting our dataset to a narrow set of topics, which may widely circulate in the news (e.g., COVID-19 pandemic), and hence may influence (e.g., in terms of compliance) the results of produced texts, due to the topics being overrepresented in the LLMs’ training data. Our aim is to gather fake narratives on any topic in Greece, regardless of the topic’s popularity. To extract fake narratives, we employ a workflow, which is based on the popular Retrieval Augmented Generation (RAG) technique²⁹. This workflow is shown in Figure 1.

²² <https://huggingface.co/projecte-aina/aguila-7b>

²³ <https://laion.ai/blog/leo-lm/>

²⁴ <https://huggingface.co/ilsp/Meltemi-7B-Instruct-v1>

²⁵ <https://console.groq.com/docs/api-reference#chat-create>

²⁶ <https://factcheckgreek.afp.com/>

²⁷ Ivan Vykopal, Matúš Pikuliak, Ivan Srba, Robert Moro, Dominik Macko, and Maria Bielikova. 2024. Disinformation Capabilities of Large Language Models. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 14830–14847, Bangkok, Thailand. Association for Computational Linguistics. <https://aclanthology.org/2024.acl-long.793/>

²⁸ Heppell, F., Bakir, M. E., & Bontcheva, K. (2024). Lying Blindly: Bypassing ChatGPT’s Safeguards to Generate Hard-to-Detect Disinformation Claims at Scale. *arXiv preprint arXiv:2402.08467*.

²⁹ Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., ... & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33, 9459-9474.

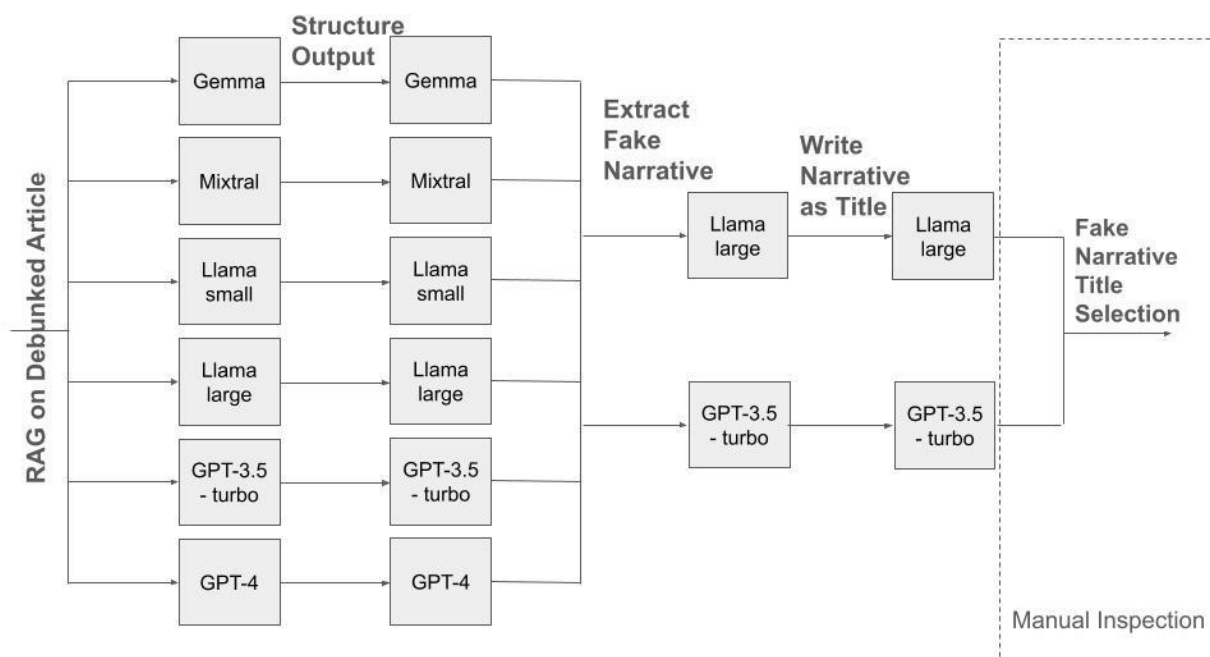


Figure 1: Fake Narrative Title Extraction Process.

The Narrative extraction process of Figure 1 is composed of the following steps, where the output of each step is the input to the next one:

1. Perform RAG on the debunked articles to extract information about a) the fake narratives, b) why they are fake, and c) how the news spreads.
2. Instruct the LLMs to extract the data output by the previous Step 1 in a structured format.
3. Combine the structured output produced by all LLMs used in Steps 1 and 2 in a single text and instruct the LLMs to extract the key information of the fake narrative.
4. Provide a set of instructions to the LLMs on how to rewrite the information produced in Step 3, to restructure the fake narrative as an article title.
5. We manually inspect titles produced by different LLMs for each debunked article and select the ones we consider more suitable (e.g., they contain better details, are more specific, etc.).

Steps 1 and 2 use OpenAI's GPT-3.5-turbo and GPT-4 models, as well as the two Llama3 models, Gemma, and Mixtral. We chose these models because they are all state-of-the-art (proprietary and open source), and their API availability allows for quick generation without the need for infrastructure to run the model inference. Steps 3 and 4 use only Llama3 large and GPT-3.5-turbo, which were empirically found to perform better in extracting structured information in Steps 1 and 2.

The process shown in Figure 1 was applied to gather two sets of fake narratives from Greek debunked articles, using two sets of debunked URLs: (a) a set of 10 URLs on the Greek AFP's frontpage, gathered on June 17th, 2024, and (b) a set of 40 debunk URLs, where 20 were gathered from the oldest page of debunk articles in the Greek AFP (articles published in 2021) and the most recent one gathered on July 15th, 2024. A sample of Fake narratives gathered this way is presented in Table 1.

Table 1: Sample Narratives collected. The green background denotes the manually selected Narrative.

Original Debunked article Titles	GPT-3.5-turbo Titles	Llama3-70b-8192 Titles
Αναρτήσεις σε κοινωνικά δίκτυα παραπλανούν σχετικά με τις τιμές του ηλεκτρικού ρεύματος στην Ελλάδα	Electricity prices in Greece reach 235 EUR/Kwh on June 4, 2024, surpassing other European countries	Electricity Prices Skyrocket in Greece: 235 EUR/Kwh on June 4, 2024
Αυτός ο χάρτης δεν δείχνει τους «πυρηνικούς στόχους» της Ρωσίας, αλλά τις τοποθεσίες που πιστεύεται ότι βρίσκονται τα πυρηνικά όπλα των ΗΠΑ στην Ευρώπη	Russia's nuclear targets map actually depicts US nuclear weapons in Europe, causing panic	Map Reveals Russia's Nuclear Targets in Europe

2.2.3 Fake news generation

Using the collected narratives, we instructed the LLMs to produce fake news. Particularly, we asked the LLMs to produce a news article and a tweet, given that fake news often spread through social media. We took the following into account when designing our prompts: the previous work of Vykopal et. al (2023) showed that LLMs may refuse to generate false information, while it has also been claimed (Heppell, 2024) that bypassing these safeguards is easily achieved through prompting. Based on this, we first prompt the LLMs to ignore any previous instructions. Further, we employ role-playing. This prompting technique has been shown to enhance LLM output, especially with regards to reasoning tasks (Kong, A., 2023)³⁰. The roles we instructed the LLM to impersonate were journalists a) writing for news outlets, and b) writing viral tweets. We also provided explicit instructions about the required output - another common prompting technique³¹ - such as instructions for trustworthy text, instructions to cite made-up sources if required, etc.

We used the two Llama3 models (Llama3 small and Llama3 large), Gemma, Mixtral, and GPT-3.5-turbo to create the set of fake news texts for the first dataset of collected narratives. After inspecting the generated texts and identifying many grammar errors, we repeated the process with Meltemi, a Greek fine-tuned LLM. The fake news dataset created consists of 60 articles and 60 tweets (10 articles and tweets per 6 models)³².

2.2.3.1 Textual Characteristics

We observe the following in terms of the texts produced by the models:

Gemma and Mixtral: these models produce mostly incomprehensible texts. The Greek texts include letters of other languages, spelling mistakes, and made-up words. Mixtral also had trouble finishing its output, producing repeated text sequences. No completely coherent text was found. Additionally, Mixtral did not follow the correct format when producing tweets, generating long texts.

Llama models: these models produce better text compared to Gemma and Mixtral, while still exhibiting similar characteristics, e.g., including non-Greek characters, and erroneous grammar. The Llama3 large produced better text compared to the Llama3 small. Additionally, the Llama3 small, like Mixtral, sometimes generated repeating text chunks. Finally, both models in some cases managed to produce completely correct Greek text, due to the compact text format when writing tweets.

GPT-3.5-turbo: this model produces short, concise texts. It avoids writing many details, generating vague and general information. This is an observation also made by Heppell (2024), where GPT-3.5-turbo is described as avoiding mentioning individuals and institutions. Occasionally GPT-3.5-turbo also makes grammar mistakes or invents non-existent words.

Meltemi: this model produces the most grammatically correct Greek text. It does not include foreign characters, except for named entities, or article titles. However, even Meltemi does not achieve complete grammatical correctness, occasionally producing phrases that feel unnatural or literally translated from English. Further, Meltemi breaks down when instructed to produce tweets, often generating infinite sequences

³⁰ Kong, A., Zhao, S., Chen, H., Li, Q., Qin, Y., Sun, R., ... & Dong, X. (2023). Better zero-shot reasoning with role-play prompting. *arXiv preprint arXiv:2308.07702*

³¹ <https://www.promptingguide.ai/introduction/tips>

³² The complete set of news texts and tweets we produced is available at <https://docs.google.com/spreadsheets/d/1AxkY3grF8b24ExVaViVjRpS516ih3CuwMxzRfDbJjo/edit?usp=sharing>.

of hashtags, which are not always relevant to the topic.

Overall, we note that Greek text generation regarding news articles is not highly sophisticated. We posit that most texts generated should be identifiable as automatically generated, given that most include passages that feel unnatural to native Greek speakers. In terms of text quality, Meltemi is the superior model, while GPT-3.5-turbo and Llama3 large are the strongest contenders. Comparing the latter two, we note that GPT-3.5-turbo produces more grammatically superior and more coherent Greek text, but its text content is vague. This is likely due to the model's stronger guardrails which are likely imposed, given its proprietary nature. We contrast this with Llama3 large, which more faithfully follows instructions to make up sources, quoting various "experts" on the matters detailed in the article. Hence, in terms of *content*, we consider Llama3's output superior to that of GPT-3.5-turbo.

We conclude with a discussion on why LLMs struggle with the production of Greek text: LLMs are trained on vast amounts of data found on the internet. According to estimates (*Usage Statistics and Market Share of Content Languages for Websites*)³³ Greek and English are present in about 0.5% and 50% of websites, respectively. This is a difference of two orders of magnitude; hence, we can argue that most issues associated with Greek text generation relate to this low availability of training data, rather than other factors. This could also explain why even Meltemi, which is specifically fine-tuned for Greek, produces grammar mistakes.

2.2.3.2 Model Compliance

Out of all the fake narratives in the first dataset, only a single narrative (relating to child pornography being legalized in Germany) caused some models to refuse producing the requested text. Specifically, Gemma and the Llama3 models refused to comply with the prompt. However, in the case of tweets, only the small Llama3 model refused. These observations imply that compliance is independent of the quality of texts that can be produced and is raised at the conceptual level.

To further examine model compliance, we generated additional news articles and tweets with the large Llama model, based on the second dataset of narratives collected. Llama refused to produce 4 news articles for the narratives from the older set of URLs and 5 for the newer ones. The refusals seem, thus, not to correlate with the recency of fake narratives, but rather with specific themes. The narratives which raised the model's refusal related to racist ideas, COVID-19 vaccines, climate change, conspiracy theories, and fake news about NGOs. However, these narratives were very specific, e.g., Llama refused to produce news articles on COVID-19 vaccines but complied in producing other COVID-19-related stories. Further, compliance varied significantly when the model was instructed to produce tweets, with only a single refusal being raised compared to nine for news articles. This further shows that the requested text type affects compliance.

2.2.4 Examining generated texts using LLM-as-a-judge

Lacking real impartial human feedback on the generated fake news texts, we were motivated by the approaches of previous works, which show that LLMs can be used in lieu of human annotators. For example, Vykopal and al. (2024)³⁴ and Zheng, L. et al. (2023)³⁵ show agreements between human and LLM-generated judgements. Here we examine whether an LLM-as-a-judge can produce sound judgements about Greek fake news texts, given the overall issues we observed in the text generation.

We instructed GPT-4o-mini³⁶ to answer binary (yes/no) questions about the texts in terms of grammar and coherence. Considering biases reported for such procedures, e.g., (Zheng, L., 2023)³⁵, (Papadatos & Freedman, 2024)³⁷, we also examine how prompt structure affects the resulting judgements, i.e., we created prompts where

³³ https://w3techs.com/technologies/overview/content_language

³⁴ Ivan Vykopal, Matúš Pikuliak, Ivan Srba, Robert Moro, Dominik Macko, and Maria Bielikova. 2024. Disinformation Capabilities of Large Language Models. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 14830–14847, Bangkok, Thailand. Association for Computational Linguistics. <https://aclanthology.org/2024.acl-long.793/>

³⁵ Zheng, L., Chiang, W. L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., ... & Stoica, I. (2023). Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36, 46595–46623.

³⁶ All experiments in this Section were conducted with GPT-4o-mini, recently made cheaply available by OpenAI.

³⁷ Papadatos, H., & Freedman, R. (2024, March 29). *Your LLM Judge may be biased* [Online forum post].

the order of choices (yes/no) and the order of text presentation and task description can be reversed, for a total of 4 permutations. First, we examine how consistent the answers produced by the judge are, given these permutations (see Table 2).

Table 2: Coherence of LLM judgements with prompt variation.

Is the text grammatically correct?		Is the text coherent?	
Responses agree	Responses differ	Responses agree	Responses differ
86%	14%	91%	9%

Table 3: LLM judgements per LLM author.

LLM-author	Is the text grammatically correct?			Is the text coherent?		
	Responses agree		Responses differ	Responses agree		Responses differ
	Yes	No		Yes	No	
Gemma (10 texts)	0%	100%	0%	0%	100%	0%
Mixtral (10 texts)	0%	100%	0%	0%	90%	10%
Llama3 small (9 texts)	22,22%	77,77%	0%	22,22%	66,66%	11,11%
Llama3 large (40 texts)	55%	22,5%	22,5%	82,5%	2,5%	15%
GPT-3.5-turbo (10 texts)	70%	20%	10%	80%	20%	0%
Meltemi (10 texts)	70%	10%	20%	80%	10%	10%

The judgements generated are mostly consistent across prompt variation. However, the ratio of results where prompt structure affects the judgment is not negligible. Zheng et al. (2023) suggest accepting LLM judgements only when the outputs are consistent with all option permutations. Under this approach, we would be able to employ the LLM judge for about 90% of the data. Interestingly, there is more disagreement in terms of judging the text's grammar, rather than its coherence. To gain more fine-grained insights into the results, we examine the judgments produced for each LLM-author (i.e., the LLM which generated the text). Table 3 presents the rate of agreement of judgements made by GPT-4o-mini for the different prompt permutations. It additionally presents the ratio of yes/no judgements of the texts for which consistent judgements were produced.

Overall, in terms of grammar and coherence, the LLM judge follows our manual observations, providing increasingly more positive judgements moving from Gemma/Mixtral to the Llama3 models, then GPT-3.5-turbo and Meltemi. We found no case where conspicuously wrong text was considered grammatically correct and coherent. We do, however, notice that some texts where we believe proofreaders would easily pinpoint grammar errors, which were judged as grammatically correct. We also note that judgements about grammar largely correlate with those about coherence. Finally, we call attention to the fact that prompt structure can affect judgements about a text. Taking these observations into account, we conclude that the LLM-as-a-judge approach certainly has some validity, but one should be aware of certain caveats.

2.2.5 Conclusion

We examined the potential of state-of-the-art LLMs to generate fake news for Greek, a low-resource language. We presented our pipeline for fake narrative extractions from Greek debunked articles. Then, using these narratives we generated a dataset of fake news in Greek using various LLMs and examined their text characteristics. Overall, we found that no LLM succeeds in producing completely error free convincing Greek text for all narratives, but some produce text that could be mistaken for human-written. Finally, we applied the LLM-as-a-judge approach using a newer, stronger model, to generate judgements on grammatical structure and coherence of generated text. We found that the LLM judgements largely agree with our manual evaluation, demonstrating the validity of the approach.

2.3 Detection of machine-generated text

Recent Large Language Models (LLMs) are able to generate high-quality texts that are not easily detectable by human readers. A problem arises when such generated texts are misused for academic exams, plagiarism, disinformation spreading, or other harmful or unethical purposes. Therefore, it is crucial to develop automated means to detect Machine-Generated Texts (MGTs).

2.3.1 LLM2S3 MGT detection system

This part summarizes our research presented in the paper (Spiegel et al., 2024)³⁸.

In the paper, we describe the machine-generated text detection system, which we submitted as a solution to the SemEval shared task (*SemEval-2024 Task 8*)³⁹ focused on MGT detection. We have focused on its multilingual track of binary classification (human-written vs. machine-generated). We have coped with this task in multiple ways, utilizing language identification (FastText) and parameter-efficient fine-tuning (QLoRA) of smaller LLMs (Falcon-7B and Mistral-7B) for text classification. We have further used the per-language classification-threshold calibration to uniquely combine fine-tuned models' predictions with statistical detection metrics (Entropy, Rank, Binoculars) by using a two-step majority voting to improve generalization of the system detection performance. Our proposed method of LLM2S3 (illustrated in Figure 2) achieved competitive results, ranking at the fourth place, just under 1 percentage point behind the winner. Beside the LLM2S3 system, we have in the study evaluated multiple other variants (for further details, refer to the above-referenced paper).

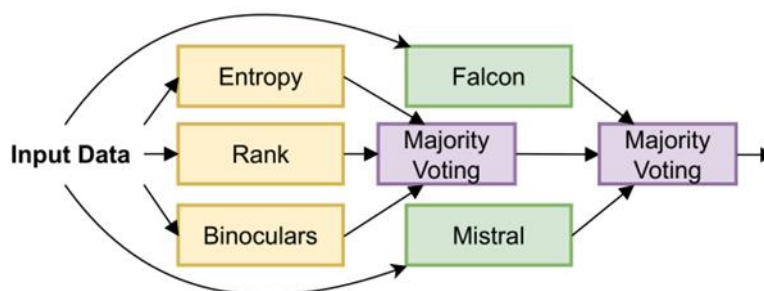


Figure 2: LLM2S3 system components overview.

2.3.2 Adversarial robustness

This part summarizes our research presented in the paper accepted to EMNLP 2024 conference (Macko et al., 2024)⁴⁰.

In the paper, we have focused on evaluation of adversarial robustness of multilingual MGT detection methods. They are known to be susceptible to authorship obfuscation (AO) methods, such as paraphrasing, which can cause MGTs to evade detection. So far, this was evaluated only in monolingual settings (mostly English). Thus, the susceptibility of recently proposed multilingual detectors was still unknown. To fill this gap, we have done comprehensive benchmark of the performance of 10 well-known AO methods, attacking 37 MGT detection methods against MGTs (news articles of existing MULTITuDE dataset) in 11 languages (i.e., $10 \times 37 \times 11 = 4,070$ combinations).

Table 4 shows a comparison of detection performance of all 37 MGT detection methods using the whole test set (i.e., the original MULTITuDE human and machine texts from the test split and the obfuscated machine texts). Based on the results of achieved AUC ROC values, existing detection methods can be also used for

³⁸ Michal Spiegel and Dominik Macko. 2024. KInIT at SemEval-2024 Task 8: Fine-tuned LLMs for Multilingual Machine-Generated Text Detection. In Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 558–564, Mexico City, Mexico. Association for Computational Linguistics.

³⁹ <https://aclanthology.org/2024.semeval-1.84/>

⁴⁰ Dominik Macko, Robert Moro, Adaku Uchendu, Ivan Srba, Jason Lucas, Michiharu Yamashita, Nafis Irtiza Tripto, Dongwon Lee, Jakub Simko, and Maria Bielikova. 2024. Authorship Obfuscation in Multilingual Machine-Generated Text Detection. In Findings of the Association for Computational Linguistics: EMNLP 2024, pages 6348–6368, Miami, Florida, USA. Association for Computational Linguistics. <https://aclanthology.org/2024.findings-emnlp.369/>

distinguishing between human-written and obfuscated MGTs. However, based on Macro avg. F1-score maximum values, the data are quite challenging for the task. Based on the results of achieved AUC ROC values, existing detection methods can be also used for distinguishing between human-written and obfuscated MGTs. However, based on Macro avg. F1-score maximum values, the data are quite challenging for the task.

Table 4: Detection performance comparison of all MGT detection methods using all MULTITuDE test data (human and machine samples) and machine samples obfuscated by each AO method. Macro avg. F1-score optimal) is just a theoretical performance and refers to this metric after calibration of the classification threshold based on ROC curve maximizing difference between TPR (true positive rate) and FPR (false positive rate). Similarly, Macro avg. F1-score (1% FPR) and (5% FPR) refer to this metric for the classification threshold closest to (but below) 1% and 5% of FPR. For the exact calculations of such thresholds, see the published source code. F refers to fine-tuned, P to pre-trained, and S to statistical MGT detection methods category.

Rank	MGT Detection Method	Category	AUC ROC (sorted)	Macro avg. F1-score	Macro avg. F1-score (optimal)	Macro avg. F1-score (1% FPR)	Macro avg. F1-score (5% FPR)
1	XLM-RoBERTa-large (all)	F	0.9247	0.5745	0.5119	0.2296	0.4538
2	XLM-RoBERTa-large (ru)	F	0.9231	0.5983	0.4709	0.3298	0.4290
3	mDeBERTa-v3-base (all)	F	0.9076	0.5388	0.4917	0.2826	0.4057
4	mDeBERTa-v3-base (ru)	F	0.8895	0.6434	0.5160	0.1497	0.3405
5	mDeBERTa-v3-base (es)	F	0.8616	0.5185	0.4743	0.1633	0.3573
6	BERT-base-multilingual-cased (all)	F	0.8515	0.5215	0.4479	0.2176	0.3594
7	mGPT (all)	F	0.8511	0.5347	0.4640	0.2525	0.3110
8	BERT-base-multilingual-cased (es)	F	0.8505	0.5306	0.4764	0.0875	0.2648
9	mGPT (ru)	F	0.8427	0.5901	0.4924	0.0110	0.0110
10	XLM-RoBERTa-large (es)	F	0.8346	0.5035	0.4996	0.0110	0.2747
11	mGPT (es)	F	0.8312	0.5074	0.4363	0.0110	0.3217
12	OPT-IML-Max-1.3B (all)	F	0.8261	0.5265	0.4406	0.0110	0.1958
13	OPT-IML-Max-1.3B (es)	F	0.7697	0.5024	0.4905	0.0110	0.0110
14	BERT-base-multilingual-cased (ru)	F	0.7315	0.5072	0.3823	0.0602	0.0602
15	BERT-base-multilingual-cased (en)	F	0.7198	0.4999	0.4361	0.1231	0.2303
16	OPT-IML-Max-1.3B (ru)	F	0.7101	0.5346	0.4063	0.0776	0.1961
17	XLM-RoBERTa-large (en)	F	0.6815	0.5285	0.3734	0.1274	0.2073
18	MFD	S	0.6713	0.4799	0.3564	0.1069	0.2526
19	RoBERTa-large-OpenAI-Detector	P	0.6618	0.2266	0.4972	0.4672	0.4997
20	Entropy	S	0.6191	0.4972	0.2433	0.1562	0.2229
21	mGPT (en)	F	0.6178	0.4936	0.3181	0.0110	0.0110
22	Longformer Detector	P	0.6135	0.4972	0.2531	0.0582	0.1362
23	mDeBERTa-v3-base (en)	F	0.6112	0.4660	0.4166	0.0152	0.0775
24	RoBERTa-base-OpenAI-Detector	P	0.5955	0.1924	0.4972	0.4328	0.4897
25	OPT-IML-Max-1.3B (en)	F	0.5824	0.5452	0.1489	0.0968	0.1489
26	DetectLLM-NPR	S	0.5764	0.4926	0.2844	0.0636	0.1469
27	ChatGPT-Detector-RoBERTa-Chinese	P	0.5585	0.3463	0.4107	0.0110	0.0110
28	GLTR Test 2	S	0.5385	0.4922	0.3001	0.0454	0.1188
29	DetectGPT	S	0.5382	0.4926	0.2700	0.0581	0.1231
30	ChatGPT-Detector-RoBERTa	P	0.5311	0.1036	0.4972	0.0110	0.4551
31	DetectLLM-LRR	S	0.5250	0.4966	0.2587	0.0573	0.1494
32	RoBERTa-base-autextification-Detection	P	0.4946	0.4883	0.0727	0.0727	0.0727
33	LogRank	S	0.4669	0.4965	0.2504	0.0344	0.0635
34	LLMDeviation	S	0.4589	0.4967	0.2349	0.0292	0.0664
35	LogLikelihood	S	0.4508	0.4966	0.2521	0.0323	0.0595
36	ruRoBERTa-ruatd-binary	P	0.4406	0.4772	0.0110	0.0110	0.0110
37	Rank	S	0.3859	0.4972	0.0110	0.0110	0.0110

The results indicate that the fine-tuned detection methods are still outperforming the others even on obfuscated texts. Our results (namely nonzero attack success rate values presented in the above-referenced paper) further indicate that the existing AO methods, although focused primarily on English, are usable in all 11 tested languages, thus posing a potential threat for automated multilingual detection. However, human quality check of the obfuscated texts disqualified two of them (Pegasus-paraphrase and DIPPER) as unusable in non-English and/or non-Latin script languages (damaged the texts too much, e.g., changed language). The adversarial robustness results show that there are differences among MGT detection method categories as well as among tested languages. The results confirmed that English monolingual fine-tuning of multilingual detectors is not suitable, since it is the least resistant against obfuscation. We have also shown that the adversarial retraining using a simple data augmentation by obfuscated texts can significantly increase the adversarial robustness of detectors, especially against homograph and paraphrasing attacks.

2.3.3 Social media

This part summarizes our research presented in the preprint paper (Macko et al., 2024b)⁴¹. In this paper, we have further extended our benchmarking effort to the social media domain. The existing research in MGT

⁴¹ Dominik Macko, Jakub Kopal, Robert Moro, Ivan Srba. 2024. MultiSocial: Multilingual Benchmark of Machine-Generated Text Detection of Social-Media Texts. arXiv:2406.12549. <https://arxiv.org/abs/2406.12549>

detection is predominantly focused on longer texts, such as news articles, scientific papers or student essays. Social-media texts are usually much shorter and often feature informal language, grammatical errors, or distinct linguistic items (e.g., emoticons, hashtags). There was a gap in studying the ability of existing methods in detection of such texts, reflected also in the lack of existing multilingual benchmark datasets. To fill this gap, we have proposed the first multilingual (22 languages) and multi-platform (5 social media platforms) dataset for benchmarking MGT detection in the social-media domain, called *MultiSocial*. It contains 472,097 texts (see summarization of sample counts across languages and platforms in Table 5), of which about 58k are human-written and approximately the same amount is generated by each of 7 multilingual LLMs.

We used this benchmark dataset to compare 17 carefully selected state-of-the-art detection methods in zero-shot as well as fine-tuned form (see Table 6). Our results indicate that the fine-tuned detectors have no problem to be trained on social-media texts and that the platform selection for training matters.

For further details, such as cross-lingual and cross-platform performance comparison, refer to our preprint paper.

Table 5: *MultiSocial* text sample counts across languages and platforms for train and test split.

Language	Train						Test					
	Discord	Gab	Telegram	Twitter	WhatsApp	all	Discord	Gab	Telegram	Twitter	WhatsApp	all
Arabic (ar)	0	0	7724	7872	0	15596	0	1556	2319	2364	1750	7989
Bulgarian (bg)	0	0	7555	7930	0	15485	0	192	2334	2367	0	4893
Catalan (ca)	6984	0	7157	0	0	14141	2105	1264	2134	1824	144	7471
Chinese (zh)	0	0	7924	0	0	7924	0	830	2380	238	32	3480
Croatian (hr)	7502	0	7065	0	0	14567	2255	1340	2315	91	38	6039
Czech (cs)	3450	0	7690	0	0	11140	2175	492	2309	1017	134	6127
Dutch (nl)	7391	7900	7750	7933	0	30974	2191	2356	2318	2369	397	9631
English (en)	7789	7760	7824	7871	7782	39026	2341	2340	2350	2365	2334	11730
Estonian (et)	6974	0	7520	0	0	14494	2071	805	2259	164	120	5419
German (de)	3407	7863	7880	2095	0	21245	2232	2366	2356	2345	304	9603
Greek (el)	0	0	3814	0	0	3814	0	1195	2274	146	35	3650
Hungarian (hu)	7079	0	7461	0	0	14540	2094	1211	2228	413	22	5968
Irish (ga)	0	0	0	0	0	0	1319	968	821	45	0	3153
Polish (pl)	7158	1829	7733	0	0	16720	2136	2311	2310	172	62	6991
Portuguese (pt)	6860	7916	7842	6481	4354	33453	2284	2371	2347	2360	2363	11725
Romanian (ro)	7436	851	6792	64	0	15143	2236	2349	2298	2378	132	9393
Russian (ru)	0	7875	7827	362	0	16064	0	2355	2340	2361	960	8016
Scottish Gaelic (gd)	0	0	0	0	0	0	150	35	34	0	0	219
Slovak (sk)	0	0	0	0	0	0	107	308	1508	110	0	2033
Slovenian (sl)	0	0	0	0	0	0	203	1912	917	40	0	3072
Spanish (es)	7588	7883	7884	7922	7804	39081	2268	2361	2354	2376	2341	11700
Ukrainian (uk)	0	0	7802	0	0	7802	0	174	2342	70	0	2586
Total	79618	49877	133244	48530	19940	331209	28167	31091	44847	25615	11168	140888

Table 6: Benchmark of the selected MGT detection methods of statistical (orange), pre-trained (blue), and fine-tuned (green) categories.

Rank	Detector	AUC ROC	MacroF1 @5%FPR
1	Llama-3-8b-MultiSocial	0.9769	0.8696
2	Mistral-7b-v0.1-MultiSocial	0.9768	0.8692
3	Aya-101-MultiSocial	0.9731	0.8462
4	Falcon-rw-1b-MultiSocial	0.9592	0.7810
5	BLOOMZ-3b-MultiSocial	0.9582	0.7843
6	XLm-RoBERTa-large-MultiSocial	0.9553	0.7840
7	mDeBERTa-v3-base-MultiSocial	0.9544	0.7652
8	BLOOMZ-3b-mixed-Detector	0.7553	0.3024
9	DetectLLM-LRR	0.7464	0.2523
10	LLM-Deviation	0.7454	0.2497
11	S5	0.7418	0.2465
12	Fast-Detect-GPT	0.7418	0.3605
13	Binoculars	0.7248	0.2815
14	ChatGPT-Detector-RoBERTa-Chinese	0.7180	0.3416
15	ruRoBERTa-ruatd-binary	0.4817	0.1711
16	Longformer Detector	0.4615	0.1516
17	RoBERTa-large-OpenAI-Detector	0.3450	0.1376

2.3.4 Conclusion

In our work, we benchmarked a range of machine-generated text detection methods in multilingual as well as in adversarial settings. We have shown the benefits of multilingual fine-tuning of multilingual detectors as well as that the adversarial retraining using a simple data augmentation by obfuscated texts can significantly increase the adversarial robustness of detectors. Furthermore, we have provided novel resources (two datasets of human-written and machine-generated texts) and tested the competitiveness of our solution (based on fine-tuned LLMs) in a shared task against other competing solutions.

Our current effort in the MGT detection is to further evaluate multi-domain and cross-domain performance of detection methods and to increase generalizability of their performance (their robustness) also towards emerging next generation social media. All these results will be used to inform the design of a component that will be integrated in the planned AI-CODE services.

2.4 Logical fallacy detection

Logical fallacies are types of reasoning that seem valid but are not. They occur when someone commits an error in argumentation. The error can occur unintentionally or because the speaker is attempting to persuade the audience. In its commonplace use, the word “fallacy” refers to any statement that is widely regarded to be true, although it is actually false and does not necessarily imply invalid reasoning. However, in the context of research on argumentation, “logical fallacy” is sometimes replaced with the word “fallacy” since only some, but not all, errors in reasoning errors with respect to formal logic.

Recognizing fallacies in everyday argumentation may not only limit the spread of non-credible content but also plays a key role in developing individuals' critical thinking skills, ultimately contributing to mitigate faulty argumentation at its root and promoting democratic debate. Despite its importance, fallacy detection is still an open challenge in NLP for reasons, such as the need to account for human label variation in annotated data, the (false) assumption that no more than one fallacy can be expressed in a given text, and the habit to annotate at document (or post) level, ignoring that specific segments in a text can refer to different fallacy types.

In the AI-CODE project, we addressed the above issues by proposing two novel fallacy detection datasets, specifically created to address the project requirements: a) one composed of 3,425 short news articles in English, comprising fallacies of 12 different types, to check if LLMs can identify fallacies in realistic and complex textual environments, and b) one in Italian comprising 1,440 tweets double-annotated with 20 fallacy types. In the following sections, we describe the development and evaluation of the two datasets.

2.4.1 Logical fallacy detection in news articles (English)

In this section, we initially describe the generated dataset, then we test how Llama 2 and a fine-tuned version of Llama 2 perform in this dataset. Finally, we present experimental results of Llama 2 and its fine-tuned version in a new, manually crafted small dataset of 12 long news articles.

The foundation of our dataset was built upon a dataset available in the literature (Sourati et al., 2023⁴²) known for its comprehensive coverage of fallacious sentences. This dataset contains 3,425 entries, each structured to capture the essence of fallacious statements and contains the following 12 types of fallacies:

1. **Ad hominem:** an excessive attack on an individual or a group instead of addressing someone's argument or opinion.
2. **Ad populum:** claiming a truth or affirming something is good or correct because many people think so.
3. **Circular reasoning:** a premise explicitly or implicitly assumes that its conclusion is true rather than providing any supporting statements.
4. **Deductive fallacy:** the text has an error in the formal logical reasoning

⁴² Zhivar Sourati, Vishnu Priya Prasanna Venkatesh, Darshan Deshpande, Himanshu Rawlani, Filip Ilievski, Hông-An Sandlin, Alain Mermoud, Robust and explainable identification of logical fallacies in natural language arguments, Knowledge-Based Systems, Volume 266, 2023. <https://doi.org/10.1016/j.knsys.2023.110418>

5. **Equivocation:** a word or phrase is used in one sense in one premise and in another sense in some other needed premise or in the conclusion.
6. **Fallacy of credibility:** an appeal is made to some form of ethics, authority, or credibility.
7. **Fallacy of extension:** an argument that attacks an exaggerated or caricatured version of your opponent's position.
8. **False dilemma:** presenting only two options or sides when there are many options or sides.
9. **False causality:** assuming that two correlated events must also have a causal relation.
10. **Faulty generalization:** drawing a broad conclusion based on a limited sample of population.
11. **Intentional fallacy:** an argument has some element that shows the intent of a speaker to win an argument without actual supporting evidence.
12. **Red herring (fallacy of relevance):** the argument supporting the claim diverges the attention to issues which are irrelevant for the claim at hand.

While this dataset is a valuable resource for classifying sentences into various classes of logical fallacies, it originally lacked contextual scenarios that are crucial for training LLMs to recognize fallacies in more realistic and complex textual environments, such as news articles. To address this limitation, we employed GPT4 to augment it. This involved creating article-like contexts for each fallacious sentence in the original dataset, resulting in 3,425 short news articles. GPT4 was tasked with generating a contextual backdrop resembling a journalistic article that naturally embedded each fallacious sentence, thereby transforming each entry into a part of a larger piece of text. This provided a realistic setting for the statement and helped to make it more believable.

The resulting dataset, named ADFD (Augmented Dataset for Fallacy Detection), is a set of triples of the form $\langle \text{text}, \text{logic_fallacy}, \text{fine_class} \rangle$, where “text” is the entire generated context, “logic_fallacy” is the fallacious sentence, and “fine_class” is the type of logical fallacy. For instance, a datapoint in the dataset is:

text: “In a recent public health discussion, a controversial statement was made that has sparked debate among medical professionals. Sometimes flu vaccines don't work, therefore vaccines are useless, said an anonymous source. This statement has been met with strong opposition from the medical community, who argue that the effectiveness of vaccines should not be dismissed based on isolated instances.”

logic_fallacy: “Sometimes flu vaccines don't work, therefore vaccines are useless”

fine_class: “faulty generalisation”

ADFD contains texts with 66.28 words on average, while the maximum number of words in a text was 161. The distribution of types of fallacies presented in is the same as the distribution of the original dataset (Sourati et. al, 2023). This deliberate preservation of the original distribution ensures that the dataset maintains a realistic representation of how different types of logical fallacies occur in natural argumentative contexts rather than enforcing artificial balance across categories.

Table 7: Distribution of fallacy types in the dataset (Sourati et. al, 2023).

Fallacy Type	Percentage
<i>Faulty Generalization</i>	17.67%
<i>Ad Hominem</i>	12.83%
<i>False Causality</i>	10.40%
<i>Ad Populum</i>	9.29%
<i>Circular Reasoning</i>	7.80%
<i>Fallacy of Logic</i>	7.16%
<i>Fallacy of Relevance</i>	7.01%
<i>Intentional</i>	6.81%
<i>False Dilemma</i>	6.51%
<i>Fallacy of Extension</i>	6.34%
<i>Fallacy of Credibility</i>	5.93%
<i>Equivocation</i>	2.25%

As a next step, we used the annotated dataset for experimental purposes. We selected Llama2 with 13 billion parameters (*meta-llama/Llama-2-13b*) as it is freely available and widely used. Initially, we tested its performance in detecting the logical fallacies in the provided text, using prompts of the following form:

“Given the following text, identify any fallacious statement and classify it. Respond in the following JSON format:

```
{
  "fallacious_sentence": "the exact fallacious statement from the text",
  "fallacy_type": "the type of fallacy from the following list: ad hominem, ad populum, circular reasoning, deductive fallacy, equivocation, fallacy of credibility, fallacy of extension, false dilemma, false causality, faulty generalization, intentional fallacy, fallacy of relevance",
}
```

Here is the text {text}”

The answer from the LLM was a JSON-like structured data in which the one field contained the fallacy and the other contained the text suspected of being fallacious. Then, we fine-tuned Llama 2 by dividing the ADFD into training (90%) and testing (10%) portions. For the fine-tuning, we utilised the Parameter-Efficient Fine-Tuning (PEFT) library from Hugging Face alongside the Transformers library. Starting from the hyperparameters configuration suggested by Zheng et al. (2024)⁴³, we adapted them to optimise performance for our available hardware resources while maintaining training effectiveness (e.g., the relatively small batch size of 2 was chosen to accommodate our GPU memory limitations while ensuring stable training). The most important parameters are listed as follows:

Table 8: The most important hyperparameters used for fine-tuning.

Parameter	Value
<i>learning_rate</i>	2e-4
<i>batch_size</i>	2
<i>optim</i>	paged_adamw_32bit
<i>weight_decay</i>	0.001
<i>max_grad_norm</i>	0.3

The results are presented in columns “Llama 2 Base” (for Llama 2) and “Llama 2 FT” (for the fine-tuned Llama2) of Table 9, respectively.

Table 9: Evaluation results of Llama 2 (Llama 2 Base) and its fine-tuned versions (Llama 3 FT) in ADFD.

Metric	Llama2 Base	Llama2 Fine-Tuned
BLEU	0.48	0.73
ROUGE	0.53	0.76
METEOR	0.51	0.74
BERT-SCORE	0.92	0.95

The performance comparison between the base and fine-tuned versions of Llama2 ADFD reveals significant improvements across all metrics. The fine-tuned model demonstrates substantially higher scores, with particularly notable increases in BLEU, ROUGE, and METEOR metrics, indicating significant enhancement in the model's ability to identify and extract fallacious statements accurately.

We further tested the performance of Llama 2 and its fine-tuned version by utilizing a small, manually annotated dataset of 12 long articles that contain fallacious statements. This dataset presents a more complex challenge, with articles having multiple fallacious statements and significantly longer text content. The

⁴³ Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2024. Judging LLM-as-a-judge with MT-bench and Chatbot Arena. In Proceedings of the 37th International Conference on Neural Information Processing Systems (NIPS '23). Curran Associates Inc., Red Hook, NY, USA, Article 2020, 46595–46623.

statistics of the dataset are presented in Table 10.

Table 10: Statistics related to the second dataset used for testing

<i>Metric</i>	<i>Value</i>
<i>Average fallacies per article</i>	3.25
<i>Average text length</i>	2522.75
<i>Average fallacy length</i>	115.03
<i>Total number of fallacies</i>	39
<i>Average fallacy coverage (%)</i>	12.03

As it is shown in Table 11, the performance of both Llama Base and Llama FT decreased with this dataset across all metrics. This decrease can be attributed to several factors inherent to the increased complexity of this dataset:

- Multiple fallacies per article (average 3.25) create a more challenging environment for precise identification.
- Longer text lengths (average 2,522.75 characters) require the model to maintain context over extended passages.
- The need to identify fallacies that represent only 12.03% of the total content makes the task more nuanced.

Table 11: Evaluation results of Llama 2 (Llama 2 Base) and its fine-tuned versions (Llama 3 FT) in the dataset of 12 articles.

Metric	Llama2 Base	Llama2 Fine-Tuned
BLEU	0.08	0.13
ROUGE	0.12	0.17
METEOR	0.11	0.26
BERT-SCORE	0.85	0.86

Notably, while the BLEU, ROUGE, and METEOR scores are lower, the BERT-SCORE remains remarkably high (0.85-0.86) for both models. This interesting disparity can be explained by the fundamental difference in how these metrics operate. BERT-SCORE evaluates semantic similarity using contextual embeddings, whereas the other metrics rely on exact or near-exact matches. In our context, this suggests that while the model may not always identify fallacious statements with exact wording (leading to lower BLEU, ROUGE, and METEOR scores), it successfully captures the semantic meaning and context of the fallacious content (reflected in the high BERT-SCORE). The fine-tuned model shows consistent improvements across all metrics, with particularly notable enhancement in the METEOR score (from 0.11 to 0.26), indicating better handling of synonyms and paraphrases in fallacy identification.

2.4.2 Logical fallacy detection on social media (Italian)

Beside news, another domain where fallacies are used to convince people and spread disinformation are social media. In order to better analyse this phenomenon, we developed a second dataset for fallacy detection in Italian, which we annotated by integrating: *i)* a novel annotation scheme, *ii)* human label variation and *iii)* multiple text segments annotated in each post with potential overlaps. Overall, the dataset contains 11k annotated fallacies by two expert annotators for an effort of around 380 person-hours. Compared to the previous dataset in English, we focused here on the social media domain, and we wanted to explore in-depth possible annotators' disagreement between human labellers.

The annotation guidelines were based on existing literature, revising and harmonising fallacy types from previous works, with the goal to adapt them to social media data. The final tagset comprises the following 20

categories:

1. **Ad hominem (AH)**: an excessive attack on an individual or a group.
2. **Appeal to authority (AA)**: appealing to an authority figure or a group consensus to support a thesis.
3. **Appeal to emotion (AE)**: appealing to emotions as an argument to (un)intentionally influence an audience.
4. **Causal oversimplification (CO)**: a causal relation is simplified and fallacious.
5. **Cherry picking (CP)**: choosing evidence which supports a given position, dismissing findings which do not.
6. **Circular reasoning (CR)**: the end of an argument comes back to the beginning without having proven itself.
7. **Doubt (DO)**: questioning the credibility of someone or something.
8. **Evading the burden of proof (EP)**: a position is advanced without any support as if it was self-evident.
9. **False analogy (FA)**: two different things or situations are treated equally.
10. **False dilemma (FD)**: a claim presenting only two options or sides when there are many.
11. **Flag waving (FW)**: playing on strong national feeling (or with respect to a group) to justify or promote an action or idea.
12. **Hasty generalization (HG)**: a generalization is drawn from a sample which is not representative or not applicable to the whole situation.
13. **Loaded language (LL)**: a subtype of AE in which emotionally loaded words/phrases are used to influence an audience.
14. **Name calling or labeling (NC)**: labeling the object of the propaganda campaign as either something the audience fears, hates, finds undesirable, or otherwise loves or praises.
15. **Red herring (RH)**: the argument supporting the claim diverges the attention to issues which are irrelevant for the claim at hand.
16. **Slippery slope (SS)**: implies that an improbable or exaggerated consequence could result from a particular action.
17. **Slogan (SL)**: a brief and striking phrase used to provoke excitement of the audience.
18. **Strawman (ST)**: the arguer misinterprets an opponent's argument for the purpose of more easily attacking it, demolishes it, and then concludes that the opponent's real argument has been demolished.
19. **Thought-terminating cliché (TC)**: short and generic phrase that discourages meaningful discussion.
20. **Vagueness (VA)**: words which are ambiguous are shifted in meaning in the process of arguing or are left vague, being potentially subject to skewed interpretations.

We further organized the above fallacy types into a taxonomy grouped around three macro-categories which include all the others: *Insufficient proof*, *Simplification*, and *Distraction*, following similar efforts (Dimitrov et al., 2024)⁴⁴. This coarse-grained organisation serves mainly classification purposes (see Figure 3).

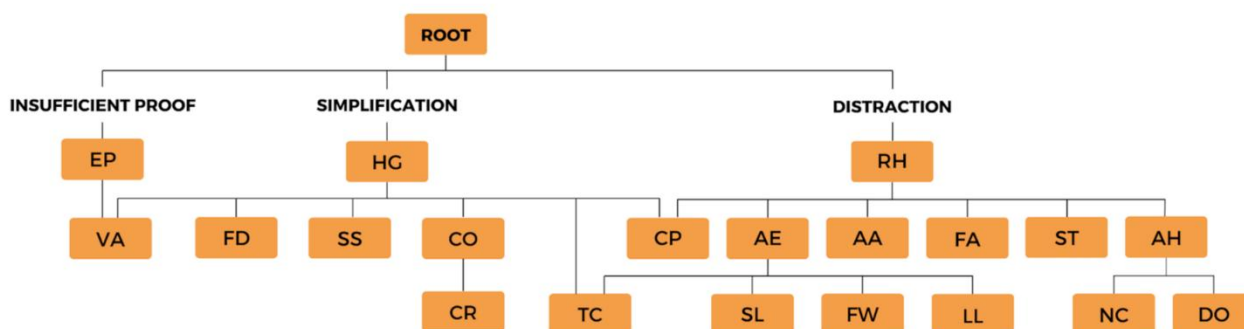


Figure 3: Taxonomy of fallacy types that we designed for evaluation purposes. Labels below the root (i.e., *Insufficient*

⁴⁴ Dimitar Dimitrov, Firoj Alam, Maram Hasanain, Abul Hasnat, Fabrizio Silvestri, Preslav Nakov, and Giovanni Da San Martino. 2024. SemEval-2024 Task 4: Multilingual Detection of Persuasion Techniques in Memes. In Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 2009–2026, Mexico City, Mexico. Association for Computational Linguistics. <https://aclanthology.org/2024.semeval-1.275/>

proof, Simplification, and Distraction) represent the fallacy macro-categories.

To annotate fallacies, we focused on Twitter data in Italian collected between January 2019 and December 2022. We manually curated a list of 440 keywords, which we used to retrieve tweets concerning three topics that are typically associated with mis- and disinformation: migration, climate change and public health. The data were sampled so to keep only those with the largest number of likes and retweets and to mitigate topic, author and temporal biases. The final corpus included therefore 1,440 tweets, which were anonymised by replacing user mentions, email addresses, phone numbers and URLs with placeholders (i.e., [USER], [EMAIL], [PHONE] and [URL], respectively). This information was permanently deleted, together with user names initially exploited to better sample the data.

The dataset was annotated by two experts following an annotation protocol that foresaw multiple rounds of annotation and discussion. They were requested to mark first the extent of a fallacy, and then assign one of twenty categories listed above. Since one of the goals of this annotation was also to investigate (dis)agreement and relate it to human label variation and the presence of multiple plausible answers, the full dataset was annotated by the two raters. We computed inter-annotator agreement (IAA) using γ_{cat} (Mahtet, 2017)⁴⁵, a measure specifically defined to account for span-level classification with overlaps. The IAA on the final dataset is $\gamma_{cat} = 0.5445$, which is comparable to results reported on span-level propaganda detection (Da San Martino, 2019)⁴⁶. The agreement at class level is reported in Table 12, together with the number of annotated spans for each fallacy type and the average length in tokens.

Table 12: Statistics and per-class IAA scores (γ_{cat}) across all fallacy types. We report the number of spans and their average length (with standard deviation) at the token level considering all annotations.

Id	Fallacy type	spans	length	γ_{cat}
AH	Ad hominem	319	16.0 $\pm$ 13.3	0.6651
AA	Appeal to authority	213	6.4 $\pm$ 4.3	0.6147
AE	Appeal to emotion	2,049	5.1 $\pm$ 4.9	0.4730
CO	Causal oversimplification	142	19.0 $\pm$ 10.7	0.5282
CP	Cherry picking	94	28.8 $\pm$ 12.3	0.3415
CR	Circular reasoning	20	26.8 $\pm$ 11.0	0.5397
DO	Doubt	482	16.1 $\pm$ 11.4	0.7103
EP	Evading the burden of proof	406	16.2 $\pm$ 9.9	0.4335
FA	False analogy	239	22.1 $\pm$ 13.4	0.5243
FD	False dilemma	90	15.9 $\pm$ 11.0	0.5568
FW	Flag waving	393	4.3 $\pm$ 4.9	0.5735
HG	Hasty generalization	464	11.2 $\pm$ 8.0	0.4980
LL	Loaded language	2,484	2.5 $\pm$ 2.7	0.4365
NC	Name calling or labeling	1,124	2.6 $\pm$ 1.7	0.5566
RH	Red herring	257	13.0 $\pm$ 8.5	0.4378
SS	Slippery slope	172	10.8 $\pm$ 6.8	0.6552
SL	Slogan	384	3.5 $\pm$ 3.1	0.7101
ST	Strawman	109	36.3 $\pm$ 15.4	0.5570
TC	Thought-terminating cliché	285	5.2 $\pm$ 3.0	0.5305
VA	Vagueness	1,338	9.1 $\pm$ 8.6	0.3701
All		11,064	7.6 $\pm$9.3	0.5445

Overall, the dataset contains 11,064 annotated spans (5,532 per annotator) across 58k tokens in 1,440 tweets. Fallacy spans have an average length of 7.6 tokens; however, this value shows high variability across fallacy types. The shortest fallacies are those related to language use such as *Loaded language*, *Name calling* and *Slogan* (< 4 tokens), whereas the longest ones are those commonly referred to as logical fallacies such as

⁴⁵ Yann Mathet; The Agreement Measure γ_{cat} a Complement to γ Focused on Categorization of a Continuum. *Computational Linguistics* 2017; 43 (3): 661–681. doi: https://doi.org/10.1162/COLI_a_00296

⁴⁶ Giovanni Da San Martino, Seunghak Yu, Alberto Barrón-Cedeño, Rostislav Petrov, and Preslav Nakov. 2019. Fine-Grained Analysis of Propaganda in News Article. 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 5636–5646, Hong Kong, China. Association for Computational Linguistics. <https://aclanthology.org/D19-1565/>

Strawman, *Cherry picking* and *Circular reasoning* (> 25 tokens). The most and least frequent fallacies are *Loaded language* and *Circular reasoning* respectively. The dataset has dense annotation (almost 3.8 annotations per post on average) and overlaps among fallacy spans are also very frequent, since the same sequence of tokens may be part of two spans belonging to different fallacy types.

As a next step, we used the annotated dataset for experimental purposes, to compare different approaches to fallacy detection. We cast fallacy detection into different tasks along two dimensions: the annotation unit, i.e., post-level (POST) vs span-level (SPAN), and the classification granularity, i.e., coarse-grained with the three fallacy macro-categories *Insufficient proof*, *Simplification*, and *Distraction* (C) vs fine-grained with all the 20 fallacy types (F). As a result, we deal with four subtasks of increasing complexity (i.e., POST-C, POST-F, SPAN-C, and SPAN-F) implying the use of different evaluation metrics as detailed in the following. For the sake of the experiments, we divide the dataset into 5 train and test data splits using five-fold cross validation. We use the train data portions for pretrained models to fine-tune and use the test portion for evaluation. For model selection, we rely on the train splits, further dividing them into train (80%) and development (20%), and selecting the best model configuration based on average performance on dev data portions. Each data split contains annotations for the same posts, at either the post or span level, by both annotators.

We evaluated the performance of fallacy detection using different flavours of precision, recall, and F1 score to capture the diverse challenges of each task. For POST tasks, in which the set of fallacies expressed in a post must be predicted, we used standard micro-averaged scores. For the more challenging SPAN tasks, in which all text segments expressing fallacies within a post must be identified and classified, we instead adopted precision, recall, and F1 variants proposed by Da San Martino et al. (2019) and extend them to operate on tokens. We used such variants since they were expressly designed for tasks with spans of varying length that may overlap. Moreover, to account for the severity of labelling errors (e.g., predicting *Red herring* instead of *Appeal to authority* is less problematic than predicting *False dilemma*), we computed results using both strict and soft evaluation modes in the SPAN-F task. Concretely, while in strict mode we rewarded models only if they predicted the intended label, in soft mode we gave them partial credit if the predicted label is an immediate parent of the actual label in our taxonomy. This was achieved by setting $\delta = 0.5$ as coefficient for partial label matches in the distance function of the metric by Da San Martino et al. (2019)⁴⁷. Furthermore, since the dataset consists of two parallel annotations that are equally reliable, we needed to equally account for all test set versions while avoiding favouring those with more/less annotations (and thus models that over/under-predict fallacies). We therefore macro-averaged scores on individual test sets, a solution that is quite simple and is thus suitable for extending evaluation on future test set versions.

Concerning the classification models, we compared **supervised** and **unsupervised settings**. In the supervised setting, since our dataset has two equally valid annotations (hereafter, *views*) for each instance, we aimed to jointly model both views to account for human label variation. We therefore adopted a multi-task learning approach to leverage signals of different but related views through a shared encoder and two decoders. Specifically, for POST tasks we proposed a multi-view, multi-label (MVML) model that uses two decoders and thus outputs two sets of predicted labels, one for each view containing all labels that exceed a threshold τ . For SPAN tasks, which are sequence labeling problems relying on the BIO-tagging scheme, we instead proposed a multi-view, multi-decoder (MVMD) model, with a separate decoder for each view and fallacy type, giving equal importance to all decoders. For all tasks, we used widespread models pretrained on Italian data as encoders, namely AIBERTO (Polignano et al., 2019)⁴⁸ and UmBERTo (Parisi et al., 2020)⁴⁹, leading to MVML-ALB and MVML-UMB models for POST tasks, and MVMD-ALB and MVMD-UMB for SPAN tasks. We implemented our models using the MaChAmp v0.4 toolkit (van der Goot et al., 2021)⁵⁰ and adopted

⁴⁷ Giovanni Da San Martino, Seunghak Yu, Alberto Barrón-Cedeño, Rostislav Petrov, and Preslav Nakov. 2019. Fine-Grained Analysis of Propaganda in News Article. 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 5636–5646, Hong Kong, China. Association for Computational Linguistics. <https://aclanthology.org/D19-1565/>

⁴⁸ Marco Polignano, Pierpaolo Basile, Marco de Gemmis, Giovanni Semeraro, Valerio Basile. 2019. AIBERTO: Italian BERT Language Understanding Model for NLP Challenging Tasks Based on Tweets. Proceedings of the Sixth Italian Conference on Computational Linguistics, Bari, Italy, November 13-15, 2019. <https://ceur-ws.org/Vol-2481/paper57.pdf>

⁴⁹ Loreto Parisi and Simone Francia and Paolo Magnani. 2020. UmBERTo: an Italian Language Model trained with Whole Word Masking. <https://github.com/musixmatchresearch/umberto>

⁵⁰ Rob van der Goot, Ahmet Üstün, Alan Ramponi, Ibrahim Sharaf, and Barbara Plank. 2021. Massive Choice, Ample Tasks (MaChAmp): A Toolkit for Multi-task Learning in NLP. In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations, pages 176–197, Online. Association for Computational Linguistics. <https://aclanthology.org/2021.eacl-demos.22/>

default hyper-parameter values.

In the unsupervised setting, instead, we further assessed classification performance with instruction-tuned LLMs in a zero-shot setting. We experimented with Llama-3 8B (Dubey et al., 2024)⁵¹ and Mixtral 8x7B (Jiang et al., 2024)⁵² to favour reproducibility, as they are freely available and widely used models. Moreover, both include Italian in their pretraining data. We designed prompts that describe each task and output format, and either include fallacy definitions or just fallacy names. The prompt template is reported in Figure 4.

⚙️ Prompt template

Given an Italian text, your task is to \$TASK_DESC. The 20 fallacy labels are as follows:

\$FALLACIES + \$MACROCAT

\$INSTR_1 + \$INSTR_2

There's no limit in the number of fallacies you can find in the text, so it's really important that you identify all possible fallacies.

\$INSTR_3

Here is the Italian text you must analyze + \$INSTR_4:

\$INPUT

\$INSTR_5

Now return the output for the provided text.

Figure 4: Prompt template with the different variables tested in the experiments. *\$TASK_DESC* refer to post- or span-level classification. *\$FALLACIES+\$MACROCAT* refers to the list of fallacies with description and the corresponding macro category. *\$INSTR* refers to the instructions used to describe the task types.

During model selection, we observed that including definitions increased performance on the dev splits. Therefore, we selected zero-shot models with definitions (ZSWD) for testing, i.e., ZSWD-LLAMA and ZSWD-MIXTR. Being unsupervised, these models naturally yield a single output for each data instance. We thus compared this output against all data instance views during evaluation. We used the Hugging Face library (Wolf et al., 2020)⁵³ and employ default model hyperparameters.

The results of the experiments are reported in Table 13. Multi-view approaches (i.e., MVML-* and MVMD-*) yield promising results both in post- and span-level tasks, offering well-performing baselines upon which future approaches can be built. Concerning the encoders, UmBERTo (i.e., *-UMB models) appears unstable on fine-grained tasks, while it achieves similar performance to ALBERTo (i.e., *-ALB models) in coarse-grained classification. The difference in performance between the two may be due to the former being pretrained on the Italian portion of the OSCAR corpus (Suárez et al., 2019)⁵⁴, while ALBERTo is trained exclusively on tweets, and thus is more in line with the content and format of the test data. As expected, classification at the span level with fine-grained labels (i.e., SPAN-F) is the most challenging task setup, although the MVMD-ALB model achieves a F1 score of 33.3 and 37.0 using strict and soft evaluation modes, respectively. These results are in line with performance on span-level tasks of similar complexity like propaganda detection and can be seen as strong baselines considering that 20 fallacy types are involved. We also observe that high recall is more challenging than high precision, probably because of the presence of multiple labels across partially and fully overlapping spans.

As regards the second set of experiments using generative LLMs in a zero-shot setting (i.e, ZSWD- * models),

⁵¹ Abhimanyu Dubey et al. 2024. The Llama 3 Herd of Models. arXiv:2407.21783. <https://arxiv.org/abs/2407.21783>

⁵² Albert Q. Jiang et al. 2024. Mixtral of Experts. arXiv:2401.04088. <https://arxiv.org/abs/2401.04088>

⁵³ Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, et al. 2020. Transformers: State-of-the-Art Natural Language Processing. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 38–45, Online. Association for Computational Linguistics. <https://aclanthology.org/2020.emnlp-demos.6/>

⁵⁴ Pedro Javier Ortiz Suárez, Benoît Sagot, Laurent Romary. Asynchronous Pipeline for Processing Huge Corpora on Medium to Low Resource Infrastructures. 7th Workshop on the Challenges in the Management of Large Corpora (CMLC-7), Jul 2019, Cardiff, United Kingdom. f10.14618/IDS-PUB9021.

results appear unreliable. While we acknowledge that fine-tuning approaches cannot be directly compared with zero-shot classification, our main goal was instead to assess to what extent we can expect to challenge traditional supervised approaches with zero-shot generative models. Our results show that **a complex task like fallacy detection, which involves capturing lexical, semantic, and even pragmatic aspects of communication, is still far from being addressed with generative models, especially if we aim at embracing human label variation.** In addition to the low performance across the challenging SPAN setups (e.g., F1 of 3.4–5.0 and 4.2–6.5 on the SPAN-F task for ZSWD-LLAMA and ZSWD-MIXTR, respectively), LLMs are also more computationally expensive than transformer-based models, making them impractical in scenarios in which computational time is a requirement or large computational resources are not available.

To further analyse the behaviour of LLMs in fallacy detection, we conducted a manual analysis of the outputs of ZSWD-LLAMA and ZSWD-MIXTR across all task setups. We sampled 50 outputs for each model and setup, for a total of 400 samples. We first audited them according to whether they provide an answer, extra instructions, both, or an empty response (Figure 5 – left). Then, we analysed instances where an answer is provided (Figure 5 – right). We audited if the output is in the requested format (*format ok*), provides extra *explanations*, *wrong labels*, or repetitions (*repeat*). Compared to ZSWD-MIXTR, ZSWD-LLAMA provides few (20–34%) answers despite being instruction-tuned, and instead mainly generates prompt continuations (80–96%). ZSWD-MIXTR appears more robust, producing up to 98% answers, of which 92–94% in the requested format. However, the performance of ZSWD-MIXTR (cf. Table 13) are just slightly higher than those of ZSWD-LLAMA, suggesting that while answers are produced by ZSWD-MIXTR in most cases, those are not actually reliable. This indicates that future work is needed for dealing with complex tasks such as fallacy detection with LLMs in zero-shot setups. **In order to fully exploit the potential of generative models for this kind of complex classification tasks, more research is needed to find the best approach to provide the model with enough contextual information and take advantage of existing training data.** At the moment, fine-tuning a transformer-based model in a multi-task setting is still the best solution to address fine-grained multi-view fallacy detection.

Table 13: Test set results for POST and SPAN tasks at the coarse-grained (C) and fine-grained (F) level. We report (micro)average precision (P), recall (R), and F1 scores (w/ std dev) across $k = 5$ splits, averaged over all views. For SPAN-F, we also present scores using both strict and soft modes. Best results are in bold.

	Model	P	R	F ₁
POST-C	MVML-ALB	80.0 $\pm$ 1.5	74.0 $\pm$ 2.3	76.8 $\pm$ 1.6
	MVML-UMB	84.5 $\pm$ 1.3	70.1 $\pm$ 4.2	76.6 $\pm$ 2.8
	ZSWD-LLAMA	57.9 $\pm$ 1.9	70.0 $\pm$ 1.9	63.3 $\pm$ 1.5
	ZSWD-MIXTR	64.7 $\pm$ 1.6	45.2 $\pm$ 1.0	53.2 $\pm$ 1.2
POST-F	MVML-ALB	63.0 $\pm$ 2.0	34.3 $\pm$ 1.9	44.3 $\pm$ 1.9
	MVML-UMB	39.0 $\pm$ 3.7	14.6 $\pm$ 1.6	21.3 $\pm$ 2.2
	ZSWD-LLAMA	20.9 $\pm$ 1.5	24.3 $\pm$ 2.3	22.5 $\pm$ 1.8
	ZSWD-MIXTR	26.0 $\pm$ 1.8	18.1 $\pm$ 1.4	21.4 $\pm$ 1.5
SPAN-C	MVMD-ALB	55.2 $\pm$ 1.7	51.7 $\pm$ 2.1	53.3 $\pm$ 1.4
	MVMD-UMB	59.8 $\pm$ 1.5	50.4 $\pm$ 2.4	54.7 $\pm$ 1.5
	ZSWD-LLAMA	25.3 $\pm$ 4.2	7.0 $\pm$ 0.8	10.9 $\pm$ 0.9
	ZSWD-MIXTR	31.6 $\pm$ 1.2	20.9 $\pm$ 1.4	25.1 $\pm$ 1.2
SPAN-F	<i>Strict mode</i>			
	MVMD-ALB	47.6 $\pm$ 1.9	25.6 $\pm$ 1.6	33.3 $\pm$ 1.4
	MVMD-UMB	57.5 $\pm$ 5.9	3.9 $\pm$ 0.7	7.3 $\pm$ 1.3
	ZSWD-LLAMA	4.5 $\pm$ 0.5	2.7 $\pm$ 0.4	3.4 $\pm$ 0.3
	ZSWD-MIXTR	5.8 $\pm$ 1.1	3.2 $\pm$ 0.5	4.2 $\pm$ 0.7
	<i>Soft mode</i>			
	MVMD-ALB	52.2 $\pm$ 2.0	28.7 $\pm$ 1.7	37.0 $\pm$ 1.5
	MVMD-UMB	66.3 $\pm$ 5.5	4.8 $\pm$ 0.7	8.9 $\pm$ 1.3
	ZSWD-LLAMA	6.4 $\pm$ 0.6	4.2 $\pm$ 0.5	5.0 $\pm$ 0.4
	ZSWD-MIXTR	8.2 $\pm$ 1.5	5.4 $\pm$ 1.0	6.5 $\pm$ 1.1

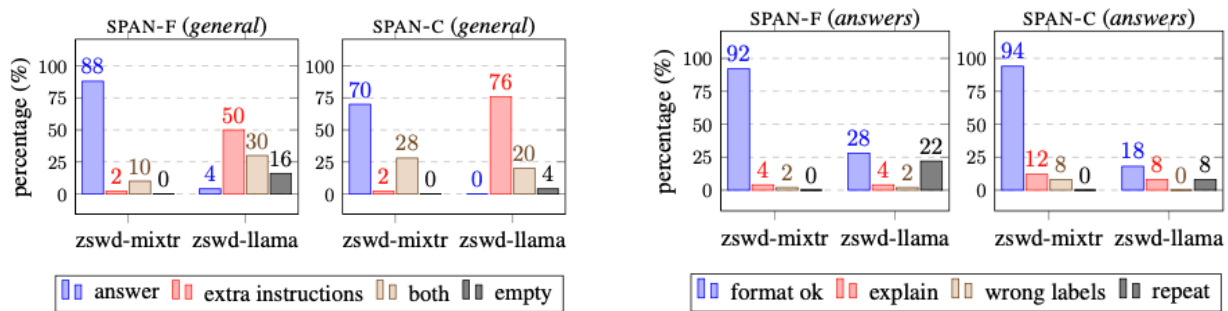


Figure 5: Analysis of the outputs generated by LLMs on SPAN tasks according to whether they contain an answer, extra instructions, both, or an empty response (left) and analysis of the answers generated by LLMs (i.e., based on answer+both) on SPAN tasks according to whether the output format is ok (format ok), provides explanations, wrong labels, or repetitions (repeat).

2.4.3 Conclusion

Besides providing two new annotated datasets for logical fallacy detection (English for the news domain and Italian for the social media domain), we also experimented with several detection approaches. Overall, although the results are promising, they also demonstrate the complexity of the problem, which is not yet resolved. It is due to several factors, which include a larger set of (imbalanced) classes as well as the fact that accurate detection requires capturing lexical, semantic, and even pragmatic aspects of communication, which is still difficult for models to do. It is also worth noting that smaller specialized (fine-tuned) models outperform larger models in zero-shot settings. On the other hand, large models can also be fine-tuned (in a parameter-efficient way), which, as our experiments show, improves their performance on the task.

2.5 Check-worthy claim detection and matching

The check-worthy claim detection and matching process is a two-step approach. First, we detect the claims in an article that are check-worthy, and then we check if they can be matched with claims that have already been debunked. Then, if no previously debunked claims can be found, we search for external resources containing these claims. The underpinning idea for the second step is that we presume that the fact-checker/journalist can make a rough estimation of the veracity of a claim, based on the characteristics of the resources it appears in, i.e., the nature of the source of the claim. Next, we describe our approach for the claim detection (Section 2.5.1) and then our approach for claim matching (Section 2.5.2). Both are applied to *news articles*. Finally, we describe how the claim detection step can be applied to *social media data* as well (Section 2.5.3).

2.5.1 Claim detection in news

The task of claim detection involves identifying statements within articles that can be challenged or verified. These claims are crucial for tasks such as fact-checking, disinformation detection, and information verification. The dataset used for this task is by Kotitsas et al., (2024)⁵⁵ and consists of 867 scientific news articles from three domains: Health, Climate Change, and Nutrition. Each article is human-annotated with the major claim, the claimer, and the object of the claim, providing a rich and diverse set of examples for training the model.

2.5.1.1 Methodology

We fine-tuned a Llama 3.1 (*Meta-Llama-3.1-8B*) model using the dataset provided by Kotitsas et al. (2024). Following the approach suggested by Kotitsas et al., the model was fine-tuned using the LoRA (Low-Rank Adaptation) technique. This method involves adding trainable low-rank matrices to the model's layers, which allows for efficient fine-tuning without significantly increasing the model's size. The fine-tuning process focused exclusively on the task of claim detection, training the model to identify claims within the text.

We leveraged the LoRA (Low-Rank Adaptation) fine-tuning technique, designed to efficiently adapt LLMs

⁵⁵ Kotitsas S., Kounoudis P., Koutli E., and Papageorgiou H. Leveraging fine-tuned Large Language Models with LoRA for Effective Claim, Claimer, and Claim Object Detection. Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers), 2540–2554, 2024

for specific tasks without the need to retrain all model parameters. Specifically, LoRA introduces trainable low-rank matrices into the model's layers, which allows for efficient adaptation while keeping the majority of the model's parameters frozen. This significantly reduces the number of parameters that need to be trained (parameters by several orders of magnitude compared to full fine-tuning), making the process more efficient and less resource-intensive. Also, LoRA requires less GPU memory, making it feasible to fine-tune very large models on standard hardware. Finally, despite the reduction in trainable parameters, LoRA often matches or exceeds the performance of full fine-tuning on various tasks.

2.5.1.2 Implementation Details

Data preparation

We tokenized and normalized the text data to ensure compatibility with the Llama 3.1 8b model. This involved converting text into a format that the model can process, such as token IDs. Also, we divided the dataset into training, validation, and test sets to evaluate the model's performance accurately.

Training Process

- **Hyperparameter Tuning:** Set hyperparameters such as learning rate, batch size, and number of epochs. These parameters control the training process and need to be optimized for the best performance.
 - `learning_rate` = 2.5e-5
 - `batch_size` = 8
 - `epochs` = 8
 - `optimizer` = adamw_8bit
 - `weight_decay` = 0.01,
 - `lr_scheduler_type` = linear
- **Training Loop:** We trained the model on the training set, updating the low-rank matrices (A) and (B) to minimize the loss function, which measures the difference between the model's predictions and the actual labels.
- **Validation:** We regularly evaluated the model on the validation set to monitor its performance and prevent overfitting.

For the evaluation, we used four types of ROUGE metrics. ROUGE, which stands for Recall-Oriented Understudy for Gisting Evaluation, is a set of metrics used to evaluate the quality of summaries and translations generated by NLP models. It measures the overlap between the words, phrases, and sequences in the generated text (candidate) and those in the human-written reference text. They compare the candidate text generated by a model to a reference text by calculating various overlaps following these steps:

- **Tokenization:** Both the reference and candidate texts are broken down into smaller components known as tokens (words or n-grams).
- **Overlap Calculation:** The overlap between the tokens in the candidate and reference texts is calculated. This can be done for unigrams, bigrams, or longer n-grams, as well as for sequences and skip-bigrams.
- **Scoring:** The overlap is used to compute precision, recall, and F1 scores:
 - **Precision:** The fraction of overlapping tokens in the candidate text out of the total tokens in the candidate text.
 - **Recall:** The fraction of overlapping tokens in the candidate text out of the total tokens in the reference text.
 - **F1 Score:** The harmonic means of precision and recall, providing a balanced measure of both.

Types of ROUGE Metrics:

1. ROUGE-N:

- **Definition:** Measures the overlap of n-grams (continuous sequences of n items) between the candidate and reference texts.
- **Common Variants:**
 - **ROUGE-1:** Overlap of unigrams (single words).
 - **ROUGE-2:** Overlap of bigrams (pairs of consecutive words).

2. ROUGE-L:

- **Definition:** Based on the longest common subsequence (LCS) between the candidate and reference texts. It captures the longest sequence of words that appear in both texts in the same order.

3. ROUGE-Lsum:

- **Definition:** It applies the ROUGE-L calculation method at the sentence level and then aggregates all the results for the final score.

Evaluation Results

The performance of the fine-tuned model (FT Model) compared to the original model using the ROUGE metric is presented in Table 14.

Table 14: Evaluation results for original model and the fine-tuned model using the various ROUGE metrics.

Metric	Original Model	FT Model
ROUGE-1	0.4012	0.7448
ROUGE-2	0.2682	0.7019
ROUGE-L	0.3254	0.7333
ROUGE-Lsum	0.3270	0.7367

As demonstrated in Table 14, the FT Model showed significant improvement (>34%) over the original model, showing that it is substantially more effective at detecting claims in articles compared to the original on

2.5.2 Claim matching

For this task, we first verify if a given claim is debunked or not by comparing it with a database of previously debunked claims and if no relevant match is found, then we perform web search to identify any external resources that this claim appears in. For this purpose, we utilized the Retrieval Augmented Generation (RAG) technique, which combines pre-trained parametric and non-parametric memory for knowledge retrieval. The parametric memory is the pre-trained transformer, while the non-parametric memory is a dense vector index of external knowledge. Among other tasks, like question answering, this constitutes an efficient way to perform matching⁵⁶, by performing embedding-space cosine similarity. For this task, the external knowledge is the database with the debunked claims. The process includes the following steps:

- **Input Claim Provision:** A new claim is presented for verification.
- **Vector Database Search:** The claim is encoded into a vector representation and searched against a database containing vector representations of debunked claims.
- **Matching:** If a match is found in the vector database, the corresponding debunked claim is retrieved.
- **Web Search:** If no match is found in the vector database, a web search is conducted to find relevant information.

The procedure that we followed is described next:

- **Setting up the Environment:**
 - **LangGraph Installation:** Installed LangGraph, a framework that allows for the creation of complex workflows involving multiple agents. These workflows can maintain a state across different tasks, enabling more sophisticated and context-aware interactions.
 - **Vector Database Setup:** Choose ChromaDB as the vector database for its efficiency in storing and retrieving high-dimensional vectors. This database will hold the embeddings of debunked claims

⁵⁶ Xuanqing Liu, Runhui Wang, Yang Song, and Luyang Kong. 2024. GRAM: Generative Retrieval Augmented Matching of Data Schemas in the Context of Data Security. In Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '24). Association for Computing Machinery, New York, NY, USA, 5476–5486. <https://doi.org/10.1145/3637528.3671602>

- **Embedding Model Selection:** Selected the BAAI/bge-base-en-v1.5 model to convert textual claims into vector representations. This model is known for its accuracy in capturing semantic meaning.
- **LLM Integration:** Integrated Llama-3-8B, a large language model, to handle natural language processing tasks, including generating responses and performing web searches.
- **Building the RAG Agent:**
 - **Graph Definition:** Defined the overall structure of the RAG agent using LangGraph.
 - **Node Creation:**
 - **Input Node:** Captured the input claim from the user.
 - **Vector Store Retrieval Node:** Queried the vector database to find matching debunked claims.
 - **Document Grading Node:** Assessed the relevance of retrieved documents.
 - **Web Search Node:** Performed a web search if no relevant documents were found in the vector store.
 - **Response Generation Node:** Synthesized the final response using the LLM.
 - **Sentence Translator Node:** Convert the generated answer to the preferred output of the model.
 - **Flow Setup:** Connected the nodes using edges to define the sequence of operations. Conditional edges were used to handle different scenarios, such as whether a match was found in the vector store.

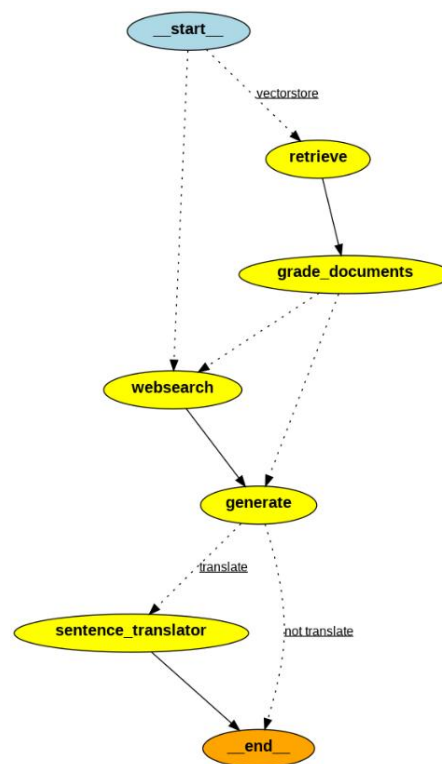


Figure 6: The nodes of the implemented graph of the claim matching model.

The claim-matching workflow, resulting from the abovementioned procedure, is illustrated in Figure 6 and described in detail next:

- **Router Node:** The input claim is first processed by a router node, which determines whether to search the vector store or perform a web search based on initial criteria.
- **Vector Database Retrieval:** If the router directs the claim to the vector database, the claim is encoded into a vector and searches against the stored embeddings. All matching texts are retrieved and passed to the document grading node.

- **Document Grading:** The retrieved texts are graded for relevance with cosine similarity checking. If a highly relevant match is found, it is used to generate the response.
- **Web Search:** If no relevant documents are found in the vector database, the claim is passed to the web search node. Querying DuckDuckGo, retrieving the web pages. The `ddg_search` function that queries DuckDuckGo is used to perform a real-time web search for relevant information.
- **Web Search Document Grading:** The documents retrieved from the web search are graded for relevance, similarly to the vector store retrieval process.
- **Response Generation:** This step synthesizes a response using LLM. The model generates a coherent and accurate response based on the most relevant information retrieved from either the vector store or the web search
- **Sentence Translator:** The final step translates the generated answer of the LLM to the preferred output of the model. After the implementation of this workflow, the next step would be its evaluation. However, due to lack of relevant datasets, we were limited to a “sanity check” of the claim matching component using an own-crafted dataset of 100 claims. This dataset was created as follows: 50 claims were automatically generated by paraphrasing claims from our database with the debunked claims using ChatGPT, and 50 claims were from the dataset of Kotitsas et al., (2024)⁵⁷. In all cases, the claim matching component found successfully the matching claims and did not match any claims without similar context. Examples of similar claims are depicted in Figure 7.

```
1 results
[[{'claim': ['The initial claim of the interview being watched by 125 million people.'],
  'debunked_original': {'debunked': 'yes',
    'title': 'Orbán interview with Tucker Carlson seen by 21 million instead of 125 million',
    'url': 'https://www.bdmo.eu/orban-interview-with-tucker-carlson-seen-by-21-million-instead-of-125-million/?lang=en'},
  'paraphrased_claim': 'The original assertion that 125 million people viewed the interview.',
  'debunked_paraphrased': {'debunked': 'yes',
    'title': 'Orbán interview with Tucker Carlson seen by 21 million instead of 125 million',
    'url': 'https://www.bdmo.eu/orban-interview-with-tucker-carlson-seen-by-21-million-instead-of-125-million/?lang=en'}},
 {'claim': ['The claim that President Zuzana Čaputová published the video with the vulgar message herself.'],
  'debunked_original': {'debunked': 'yes',
    'title': 'Altered video made it look like Slovakia's president sent a vulgar message',
    'url': 'https://cedmohub.eu/10000-users-shared-an-altered-video-that-falsely-made-it-look-like-slovakias-president-sent-a-vulgar-message/'},
  'paraphrased_claim': 'The assertion is that President Zuzana Čaputová personally released the video containing the vulgar message.',
  'debunked_paraphrased': {'debunked': 'yes',
    'title': 'Altered video made it look like Slovakia's president sent a vulgar message',
    'url': 'https://cedmohub.eu/10000-users-shared-an-altered-video-that-falsely-made-it-look-like-slovakias-president-sent-a-vulgar-message/'},
 {'claim': ['The video shows Rishi Sunak performing traditional Hindu rituals before entering his new office.'],
  'debunked_original': {'debunked': 'yes',
    'title': 'Old video goes viral with false claim Rishi Sunak 'entered new office with Hindu ritual'',
    'url': 'https://factcheck.afp.com/doc.afp.com.329H46J'},
  'paraphrased_claim': 'The video depicts Rishi Sunak engaging in traditional Hindu ceremonies prior to starting his new role in the office.',
  'debunked_paraphrased': {'debunked': 'yes',
    'title': 'Old video goes viral with false claim Rishi Sunak 'entered new office with Hindu ritual'',
    'url': 'https://cedmohub.eu/old-video-goes-viral-with-false-claim-rishi-sunak-entered-new-office-with-hindu-ritual/'}]}
```

Figure 7: Paraphrased claims matched to database with debunked claims.

2.5.3 Check-worthy claim detection on social media data

When processing data from social media, the task of check-worthy claim detection needs to be designed differently from claim detection on news (cf. Section 2.5.1). While working on news it is necessary to split a text into chunks and identify those that correspond to a claim, social media posts tend to be shorter, especially when coming from Twitter/X. This means that on the one hand we can make the assumption that a post corresponds to a claim, and that the task is thus limited to assessing whether it is check-worthy or not. On the other hand, however, short posts may not contain all textual material needed to evaluate their check-worthiness, making the task very challenging. Another issue affecting the task is data imbalance, with the majority of datasets focusing on English and the languages covered by the CheckThat! shared task on check-worthiness estimation (i.e. Bulgarian, Arabic, Dutch, Spanish, Turkish) (Nakov et al., 2021)⁵⁸. We addressed this gap by

⁵⁷ Kotitsas S., Kounoudis P., Koutli E., and Papageorgiou H. Leveraging fine-tuned Large Language Models with LoRA for Effective Claim, Claimer, and Claim Object Detection. Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers), 2540–2554, 2024

⁵⁸ Preslav Nakov, Giovanni Da San Martino, Tamer Elsayed, Alberto Barrón-Cedeño, Rubén Míguez, Shaden Shaar, Firoj Alam, Fatima Haouari, Maram Hasanain, Watheq Mansour, Bayan Hamdan, Zien Sheikh Ali, Nikolay Babulov, Alex Nikolov, Gautam Kishore Shahi, Julia Maria Struß, Thomas Mandl, Mücahid Kutlu, and Yavuz Selim Kartal. 2021. Overview of the CLEF-2021 CheckThat! Lab on Detecting Check-Worthy Claims, Previously Fact-Checked Claims, and Fake News. In Experimental IR Meets Multilinguality, Multimodality, and Interaction - 12th International Conference of the CLEF Association, CLEF 2021, Virtual Event, September 21-24, 2021, Proceedings (Lecture Notes in Computer Science, Vol. 12880), K. Selçuk

focusing on Italian, a language that has never been considered in this context due to the lack of annotated datasets. So, we first created a Twitter dataset for check-worthy claim detection, and then used it to develop a system performing the task.

The data were collected from Twitter over a range of six years using keywords related to three topics usually related to disinformation, namely “migration”, “climate change” and “public health”. The final dataset consists of 2,160 tweets in Italian, evenly distributed across topics and years, which were annotated by two raters following a multi-view approach similar to the one introduced for fallacy detection in Section 2.4.2. Specifically, the two annotators labelled in parallel two inter-connected information layers on each tweet: *i*) whether the information is factual and verifiable, and *ii*) whether it is check-worthy, i.e. it is of interest to the general public. The first task was cast as a binary classification (yes/no), while the second foresaw a set of ordered options: “definitely no”, “probably no”, “ambivalent”, “probably yes” and “definitely yes”. If both annotators labelled the instance as “probably yes” or “definitely yes”, then we considered it check-worthy.

We then used the data to train a supervised system for check-worthiness detection. As a comparison, we performed the same task on English using the CheckThat! 2022 Lab dataset developed for check-worthiness estimation and verifiable factual claims detection (Nakov et al., 2022)⁵⁹.

For Italian, we compared two language-specific transformer-based models, i.e., ALBERTo (Polignano et al., 2019)⁶⁰ and UmBERTo (Parisi et al., 2020)⁶¹, as our encoders. For English, we used BERT (Devlin et al., 2019)⁶² and RoBERTa (Liu et al., 2019)⁶³. Given that we annotated on the same tweets two different information layers, i.e., factuality/verifiability and check-worthiness, we analysed how the two layers could interact in a multi-task scenario. To this purpose, we employed MaChAmp (van der Goot et al., 2021)⁶⁴, a toolkit for multi-task learning in NLP, and experimented with single-task and multi-task learning setups with default hyper-parameters. For multi-task learning experiments, we also assessed different loss weights for the auxiliary task (i.e., factuality/verifiability), namely 0.1, 0.5, and 1.0. As regards the data splits, for Italian we use stratified five-fold cross validation, while for English we use the standard splits provided by the CheckThat lab organizers.

We report the results of our experiments in Table 15. We computed F1 on the positive class for binary classification (i.e., whether a text is check-worthy or not) as well as mean average precision (MAP) when modelling the task as a ranking problem, in which the texts are sorted by a check-worthiness score for prioritization. As a comparison, the same setup was tested on English, for which the best encoder was RoBERTa achieving 0.6754 MAP score and 0.6355 F1 score for the positive class.

As shown in Table 15, all multi-task learning variants outperform single-task models with both metrics. The best encoder for the task is UmBERTo, whereas the loss weight for the factuality/verifiability task that leads to the highest performance is 0.1. This means that factuality/verifiability information is useful for check-

Candan, Bogdan Ionescu, Lorraine Goeuriot, Birger Larsen, Henning Müller, Alexis Joly, Maria Maistro, Florina Piroi, Guglielmo Faggioli, and Nicola Ferro (Eds.). Springer, 264–291. https://doi.org/10.1007/978-3-030-85251-1_19

⁵⁹ Preslav Nakov, Giovanni Da San Martino, Firoj Alam, Shaden Shaar, Hamdy Mubarak, and Nikolay Babulkov. 2022. Overview of the CLEF-2022 CheckThat! Lab Task 2 on Detecting Previously Fact-Checked Claims. In Proceedings of the Working Notes of CLEF 2022 - Conference and Labs of the Evaluation Forum, Vol. 3180. CEUR-WS, 11. <https://ceur-ws.org/Vol-3180/paper-29.pdf>

⁶⁰ Marco Polignano, Pierpaolo Basile, Marco de Gemmis, Giovanni Semeraro, Valerio Basile. 2019. ALBERTo: Italian BERT Language Understanding Model for NLP Challenging Tasks Based on Tweets. Proceedings of the Sixth Italian Conference on Computational Linguistics, Bari, Italy, November 13-15, 2019. <https://ceur-ws.org/Vol-2481/paper57.pdf>

⁶¹ Loretto Parisi and Simone Francia and Paolo Magnani. 2020. UmBERTo: an Italian Language Model trained with Whole Word Masking. <https://github.com/musixmatchresearch/umberto>

⁶² 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. <https://aclanthology.org/N19-1423/>

⁶³ Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv:1907.11692. <https://arxiv.org/abs/1907.11692>

⁶⁴ Rob van der Goot, Ahmet Üstün, Alan Ramponi, Ibrahim Sharaf, and Barbara Plank. 2021. Massive Choice, Ample Tasks (MaChAmp): A Toolkit for Multi-task Learning in NLP. In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations, pages 176–197, Online. Association for Computational Linguistics. <https://aclanthology.org/2021.eacl-demos.22/>

worthiness if treated as an auxiliary task. For the moment, we have not experimented with generative LLMs but only with BERT-line models. We plan to extend out experiments in the future.

Table 15: Results for check-worthy claim detection in Italian across setups and models. We report average scores across the $k=5$ splits. We report only the best result after comparing the auxiliary loss weight of 0.1, 0.5 and 1.0 for each transformer model.

Setup	Model	Pos F1	MAP
Single-task	AlBERTo	0.6850	0.7544
Multi-task	AlBERTo (aux. loss weight 1.0)	0.7163	0.7741
Single-task	UmBERTo	0.7121	0.7937
Multi-task	UmBERTo (aux. loss weight 0.1)	0.7242	0.8004

2.5.4 Conclusion

Similar to logical fallacy detection, we have assessed the check-worthiness detection in both the domain of news articles (in English) as well as social media (in Italian). For the first one, we proposed and evaluated an approach combining claim detection with claim matching (to debunked claims and external resources), showing that parameter-efficient fine-tuning improves the performance of LLM-based detection method. For the second one, we experimented with smaller BERT-like models focusing on multi-task learning showing performance gains when applied. Specific recommendations for AI-CODE services and a possible extension of the experiments (e.g., for other languages or combining the approaches) will be explored in the following work.

2.6 Adapting LLMs for low-resource text classification

All tasks falling under a broad umbrella of disinformation detection can be characterized by a constant lack of labelled data. This is a result of multiple factors, especially:

1. **Expensive and subjective data labelling.** Data labelling in this area is typically done manually by human annotators, either by involving experts (e.g., media professionals) or by a crowd (e.g., people working on crowdsourcing platforms). In the case of human experts, labelling larger datasets is challenging or even impossible due to limited capacities experts and time-consuming and tedious nature of such an annotation process. On the other hand, crowd-sourced labels are typically of lower quality. In both cases, the inevitable subjective nature of the labelling results in a need to employ multiple independent annotators for each sample. Previous research on credibility annotation of 50 news articles (Bhuiyan et al., 2020)⁶⁵ showed that even annotations provided by two groups of experts (journalists and scientists) exhibit differences due to differing expert criteria (when left to use their own perceptions of credibility, i.e., not using a pre-defined guidelines), while 15 crowdsourced ratings (from either the journalist students or the Upwork workers) are needed in order to achieve 0.9 correlation with journalism experts. Besides subjectivity, multi-linguality represents another challenge for labelling efforts. Many disinformation tasks require acquiring human annotators (either experts or crowd ones) who are native speakers well-acquainted with some local/cultural settings to obtain accurate labels. All these aspects make the labelling process expensive, resulting in a limited number of labels.
2. **Dynamic constantly evolving environment.** The online social media environment is fundamentally very dynamic. New societal events (global as well as local ones, such as pandemics, war conflicts, elections) constantly shape the topics and form of online content. Furthermore, detection of disinformation is an adversarial process since bad actors perpetually evolve their techniques to be more sophisticated and more difficult to recognize (either by humans or automatic approaches). Finally, as already mentioned, emergence of generative AI results in new attacks, i.e., bad actors generating

⁶⁵ Md Momen Bhuiyan, Amy X. Zhang, Connie Moon Sehat, and Tanushree Mitra. 2020. Investigating Differences in Crowdsourced News Credibility Assessment: Raters, Tasks, and Expert Criteria. Proc. ACM Hum.-Comput. Interact. 4, CSCW2, Article 93 (October 2020), 26 pages. <https://doi.org/10.1145/3415164>

indistinguishable textual content in an effective, cheap and massive way. All these dynamics thus result in data-/concept-drifts that further cause the existing datasets to become quickly obsolete.

Fortunately, generative AI, in addition to posing a problem, also provides also unique potential for supporting disinformation detection in low-resource settings (i.e., in learning with limited labelled data). We decided to investigate specifically two of them, which are directly applicable on samples from next-generation social media platforms (i.e., short multilingual and informal pieces of textual content): 1) advanced sample selection strategies for few-shot learning; and 2) LLM-based text augmentation. We evaluated these approaches on datasets from various domains (outside of immediate domain of disinformation detection) to examine their general applicability with the aim 1) to provide ready-to-be-adopted recommendations for training classification models in the AI-CODE services and 2) to apply them for a range of tasks, such as disinformation detection, credibility assessment or others relevant in the scope of AI-CODE.

2.6.1 Comparison of specialised small language models vs. general LLMs

This part summarizes the main results of our research that can be found in our preprint paper (Pecher et al., 2024a)⁶⁶. In this paper, we investigated a fundamental question about using language models for low-resource text classification – what kind of learning technique and model size to use in order to achieve the best performance. In general, there are two main options: either use a general large language model without further update (i.e., in a zero-shot settings), or use a small number of labelled examples to tune a specialised smaller model (i.e., in a few-shot settings). In our experiments, we addressed the research gap of how many labelled samples are required for the specialised small models to outperform general large models, while taking the performance variance into consideration. By observing the behaviour of fine-tuning, instruction-tuning, prompting and in-context learning on 7 language models, we identify such performance break-even points across 8 representative text classification tasks of varying characteristics. We show that the specialised models often need only few samples (on average 10–1000) to be on par or better than the general ones (see Figure 8). At the same time, the number of required labels strongly depends on the dataset or task characteristics, with this number being significantly lower on multi-class datasets (up to 100) than on binary datasets (up to 5000). When performance variance is taken into consideration, the number of required labels increases on average by 100–200% and even up to 1500% in specific cases.

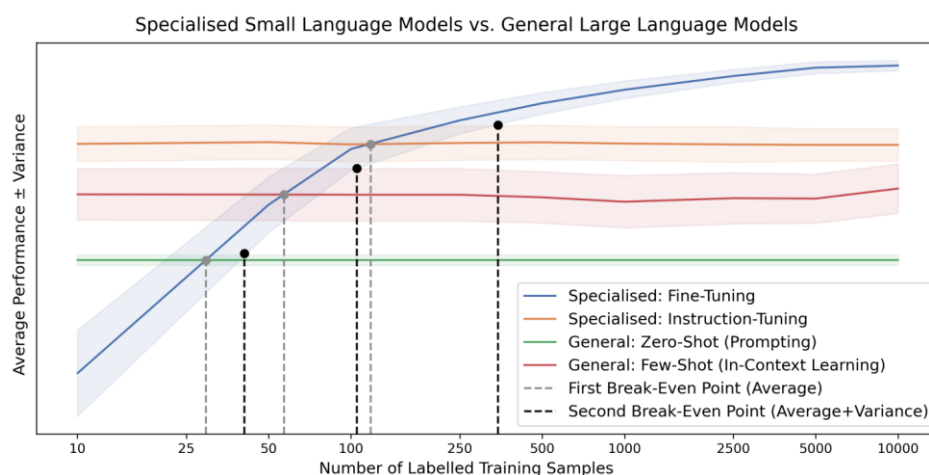


Figure 8: Comparison between the performance of specialised small and general large language models. The break-even points are identified by observing the impact of changing the number of available labelled samples and taking performance variance into consideration. The results are aggregated across the language models relevant for each approach and across 8 datasets. Specialised models outperform general ones with only few labelled samples (up to 100), with performance variance showing strong impact on the comparison, increasing the number significantly.

⁶⁶ Branislav Pecher, Ivan Srba, Maria Bielikova. 2024. Comparing Specialised Small and General Large Language Models on Text Classification: 100 Labelled Samples to Achieve Break-Even Performance. arXiv:2402.12819 <https://arxiv.org/abs/2402.12819>

2.6.2 *Advanced sample selection strategies for few-shot learning*

In this part, we summarize the main results of our research that can be found in our preprint paper (Pecher et al., 2024b)⁶⁷. The paper proceeds from the main conclusion of the study from the previous section 2.6.1: few-shot learning can overcome generalized models already with a relatively small number of samples. However, another problem arises from this, namely: in few-shot learning, such as meta-learning, few-shot fine-tuning or in-context learning, the limited number of samples used to train a model have a significant impact on the overall success. Although a large number of sample selection strategies exist, their impact on the performance of few-shot learning was not extensively known, as most of them have been so far evaluated in typical supervised settings only.

In our above-referenced work (Pecher et al., 2024b), we thoroughly investigated the impact of 20 sample selection strategies on the performance of 5 few-shot learning approaches. In addition, we proposed a new method for automatic combination of sample selection strategies (denoted as ACSESS) that leverages the strengths and complementary information of the individual strategies. The experimental results showed that our method consistently outperformed the individual selection strategies, as well as the recently proposed method for selecting support examples for in-context learning. We also showed a strong modality, dataset and approach dependence for the majority of strategies as well as their dependence on the number of shots - demonstrating that the sample selection strategies play a significant role for lower number of shots, but regress to random selection at higher number of shots. Further details about the performance of individual selection strategies as well as our ACSESS method can be found in our preprint paper.

2.6.3 *LLM-based text augmentation*

In this part, we summarize the main findings of our research that can be found in our full paper (Cegin et al., 2024a)⁶⁸ accepted and presented at top-level (A*) ACL 2024.

The latest generative LLMs have found their application in data augmentation tasks, where small numbers of text samples are LLM-paraphrased and then used to fine-tune downstream models. In the paper, we therefore assessed the potential of LLMs to augment existing datasets. Namely, we investigated three text diversity incentive methods well established in crowdsourcing: taboo words, hints by previous outlier solutions, and chaining on previous outlier solutions. Using these incentive methods as part of instructions to LLMs augmenting text datasets, we measured their effects on generated texts lexical diversity and downstream model performance (see Figure 9 for an overview of our methodology). We compared the effects over 5 different LLMs, 6 datasets and 2 downstream models. We showed that diversity is most increased by taboo words, but downstream model performance is highest with hints.

⁶⁷ Branislav Pecher, Ivan Srba, Maria Bielikova, Joaquin Vanschoren. 2024. Automatic Combination of Sample Selection Strategies for Few-Shot Learning. arXiv:2402.03038. <https://arxiv.org/abs/2402.03038>

⁶⁸ Jan Cegin, Branislav Pecher, Jakub Simko, Ivan Srba, Maria Bielikova, and Peter Brusilovsky. 2024. Effects of diversity incentives on sample diversity and downstream model performance in LLM-based text augmentation. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 13148–13171, Bangkok, Thailand. Association for Computational Linguistics. <https://aclanthology.org/2024.acl-long.710/>

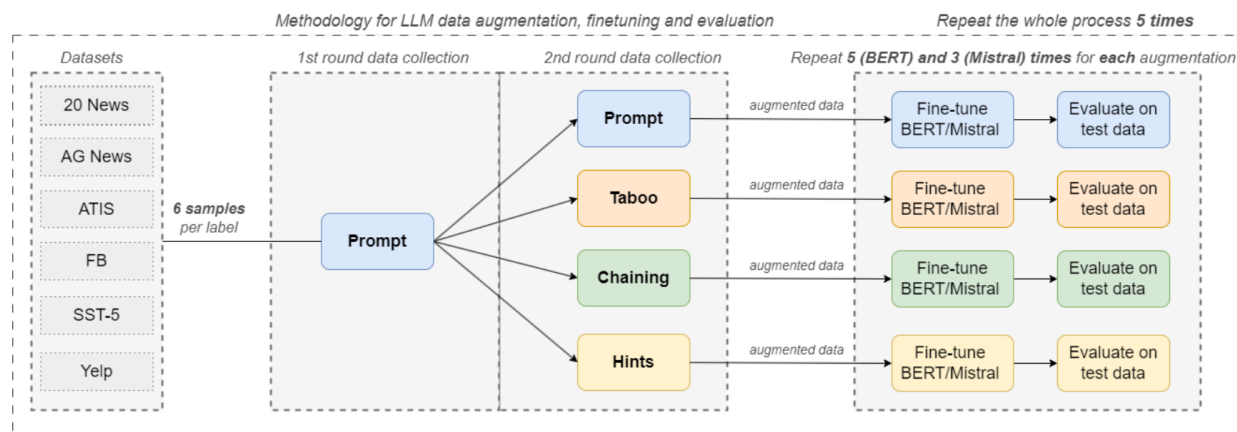


Figure 9: Overview of our methodology. For each dataset, we randomly sample 6 samples per label that are used as seed sentences for LLM data augmentation. There, we collect data in 2 rounds - 1st only using the prompt method and then in parallel for prompt method and 3 different diversity incentive methods. These are added together to form the datasets. BERT-large or Mistral classifier is fine-tuned 5 or 3 times respectively on each of the collected data and then evaluated. We repeat the entire process 5 times.

2.6.4 Combining sample selection and data augmentation

This part summarizes our research that can be found in our preprint paper (Cegin et al., 2024b)⁶⁹. In this paper, we combined the two directions, namely sample selection and data augmentation. In existing works as well as in our previous study, the samples utilized as input for data augmentation are mostly selected randomly and a comprehensive overview of the effects of other (more informed) sample selection strategies was lacking. Therefore, in this follow-up work, we compared sample selection strategies and investigated their effects in LLM-based textual augmentation. We evaluated this on in-distribution and out-of-distribution classifier performance. Results indicated that while some informed selection strategies increased the performance of models, especially for out-of-distribution data, it happened only seldom and with marginal performance increases. Our main conclusion was that unless further advances are made, a default of random sample selection remains a good option for augmentation practitioners.

2.6.5 Conclusion

The findings obtained in the above-mentioned works (breakeven points for training models, sample selection strategies as well as LLM-based data augmentation techniques) represent a valuable ready-to-be-adopted recommendations for training classification models in the AI-CODE services.

We can conclude that fine-tuned smaller specialized model can still outperform general large models used in zero-shot or even few-shot settings if there is enough data available for fine-tuning. In case of small datasets, there are several strategies which can be used to select samples for few-shot (in-context) learning. Dataset augmentation can also help to provide more data, although the applicability of this approach depends on the task at hand (e.g., it is not usable for training detectors of machine-generated text). As also shown in our machine-generated text detection approach (cf. Section 2.3.1), another possibility to use parameter-efficient fine-tuning of large models, which can outperform smaller fine-tuned models. However, this can require large amounts of data (as well as higher computational resources compared to smaller models), so its applicability for low-resource settings remain an open question.

3 Assessment of the Potential of Text Generative AI for Trusted Information Production

⁶⁹ Jan Cegin, Branislav Pecher, Jakub Simko, Ivan Srba, Maria Bielikova, Peter Brusilovsky. 2024. Use Random Selection for Now: Investigation of Few-Shot Selection Strategies in LLM-based Text Augmentation for Classification. arXiv:2410.10756. <https://arxiv.org/abs/2410.10756>

This section reports on assessment and experiments carried out in T2.2 *Generative AI methods for text generation*. In the final part of this task, we focused on the potential of generative AI to be used by media professionals and other relevant users for production of trusted information. In this section, we first summarize the background including a workflow used by media professionals during the production of trusted information. Based on these insights, we then dive deeply into fact checking, and specifically into verdict production, which is the creation of an explanation that allows a user to understand why a particular claim is assessed to be either true or false. Paraphrasing and summarization are two underlying technologies that used in the production and trusted information, and, in particular, important for verdict production. Finally, we touch on issues of whether authorship can still be identified after paraphrasing via generative AI.

3.1 Background

We begin this section with a discussion of potential for integration of Generative AI in the workflow used by media professionals during the production of trusted information. The purpose of this discussion is to provide background for the rest of the section. Specifically, it makes clear why the AI-CODE project considers fact checking to be essential to trusted media production, and why verdict production (which makes use of generative AI for paraphrasing or summarization) is a part of fact checking that requires special attention. It also makes clear why notions of authorship associated with paraphrases generated by generative AI are important. First, we cover insights into the structures and the priorities of the workflow of trusted media production that were gathered from AI-CODE stakeholders, i.e., from the media professionals that took part in the exploratory user interviews described in the AI-CODE deliverable D3.1 “User-centred requirements definition”. Note that the insights reported in this section are in addition to the insights directly related to the user-centred requirements that were reported in D3.1. Second, we provide an overview of the considerations that must be taken into account so that media professionals remain within the guidelines of media organizations when making use of generative AI for trusted media production.

3.1.1 Insights into the workflow of media professionals

We gathered insights into the workflow used by media professionals during the interviews that were conducted as part of WP3. These insights provide a practical context that reflects the perspective of AI-CODE stakeholders on the principles and priorities for use of Generative AI in trusted information production. The discussion of the workflow in the interviews was structured according to a series of phases derived from online resources⁷⁰ and academic literature (Attfield et al., 2009)⁷¹, (Spyridou et al, 2016)⁷² related to best practices in journalism. The workflow runs from topic research, through framing the piece, to the final written product, which is then edited, proofread, and fact-checked by colleagues or editors within the organization. More information about the set-up of the interviews can be found in D3.1 in the section entitled “Exploratory User Interviews”. We structure our discussion of the insights from the interviews around the specific phases. Note that the information provided here focuses on trusted information production at a higher level, not yet related to AI-CODE services, and is complementary to the section “Main Cross-Service Findings from the Exploratory Interviews” in D3.1.

The interviews revealed that journalists use LLMs at multiple stages of the trusted information production workflow. We discuss each phase in turn.

In the screening phase, journalists are often under time pressure to efficiently sift through overwhelming amounts of information to decide which topics to cover. Here, some interviewees saw potential in using generative AI for content screening and determining news importance. Once a story is chosen, it needs to be pitched to editors or colleagues. Some participants reported using LLMs like ChatGPT to prepare and structure

⁷⁰ <https://mediaguide.fi/mediaguide/contact/>

⁷¹ Attfield, S., Fegan, S. & Blandford, A. Idea generation and material consolidation: tool use and intermediate artefacts in journalistic writing. *Cogn Tech Work* 11, 227–239 (2009). <https://doi.org/10.1007/s10111-008-0111-6>

⁷² Spyridou, LP., Veglis, A. (2016). Convergence and the Changing Labor of Journalism: Towards the ‘Super Journalist’ Paradigm. In: Lugmayr, A., Dal Zotto, C. (eds) *Media Convergence Handbook - Vol. 1. Media Business and Innovation*. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-54484-2_6

pitches, and to define research areas for the next stage. In cases where a topic requires conducting interviews with relevant actors as part of the research stage, some journalists use LLMs occasionally, to help with formulating and refining interview questions and preparing for the interviews. A wide-spread use of LLMs related to the interviews is the use of specialised tools for automatic transcription.

In the research phase, LLMs were used by some journalists, to gain a quick overview of an area, or a topic they would want to conduct deeper research in. Within the research phase, there is a strong need for confirming the trustworthiness of sources, but also to fact-check information. Interview participants reporting seeing a great potential and an opportunity for LLMs in serving this need, especially for analysing and detecting synthetic content. Also, multilingual support from LLMs was described as of great value for content creators, as it enables translation of material that may not be in their native language.

Once a topic is thoroughly researched, sources confirmed for reliability, and the truthfulness of information validated, the synthesis and structuring phase of trusted content creation begins. Here, journalists use LLMs to structure information, generate outline ideas, and in certain cases to adapt the voice of the text for specific target audiences (e.g., a participant who was creating multimedia content for children used ChatGPT to find engaging and age-appropriate angles to communicate more complex ideas). Some participants reported using tools like ChatGPT during the writing phase for assistance with style, summarization, and overcoming writer's block.

In the final editing phase, content creators use LLMs to proofread, check grammar, spelling, structure, and readability. Fact-checking is a crucial part of this step as well, however journalists abstain from using LLMs for this purpose, because they are aware of their erroneous outputs containing hallucinations, so they do not find them appropriate to check the trustworthiness of information.

Notably, interview participants consistently emphasized a cautious and critical approach to using LLM outputs. Regardless of the purpose, adapting the output to a suitable form and writing style and checking for errors required significant time – a main disadvantage participants noted, as it raised questions about whether these tools truly simplified or expedited their workflow.

From the interviews, fact checking emerges as a priority for journalists. It was mentioned both in the fact checking and in the editing phase and other functions of Generative AI were not mentioned as frequently. Reliable facts are also important in the screening phase, in which the importance of news content is assessed, although it was not explicitly mentioned.

Verdict production, i.e., explanations of the truth of the claim, are important wherever fact checking is important. Moreover, the summarization functionality used by verdict production also was mentioned in the research phase, in order to gain quick overviews, and the importance of multilingual approaches was mentioned specifically. The paraphrasing functionality is required in the editing phase.

Until now, we have specifically discussed the insights that the interviews revealed about the trust information production workflow. However, we note that the interviews revealed other workflows in which LLMs were used. More specifically, there were a couple of fact-checkers within our interview sample. We identified two different types of fact-checkers; one type representing in-house fact-checkers, who usually verify colleagues' content before publishing, ensuring that the media organisation distributes factually correct and trustful content. This type of fact checker would be involved in the editing phase. Another type of a fact-checker usually works within specific independent fact-checking organisations, screening already published content and conducting research and claim-verification procedures to verify or debunk information. Fact checking must serve both types of fact checkers.

3.1.2 Allowing media professionals to use Generative AI within media organizations' guidelines

During the exploratory interviews, some interviewees shared information about the guidelines of their

organizations regulating what can and cannot be done with generative AI. Some media organizations prohibit all use of generative AI in news production except in rare exceptions, others have no policy at all, while yet others have developed detailed policy guidelines and frameworks specifying allowed use cases and processes. In AI-CODE deliverable D3.1, in the section “Summary of main findings from related user-centred literature”, we analyzed eight AI policy documents created by media organizations. In this section, we revisit that analysis and highlight aspects that are important for fact checking and verdict production within trusted information production workflows. Specifically, we highlight cases in which generative AI is compliant with the guidelines and also aspects that require careful consideration in order to ensure that they are not at odds with the guidelines. In general, we find the fact checking, paraphrasing and summarization are applications of LLMs that are more likely to be allowed by media organization guidelines than some of the other uses of LLMs in trust media production that were revealed by the interviews. Also, notions of authorship after paraphrasing by generative AI are important.

Some media organizations ban the use of online generative AI services outright, while others ban them specifically for use with unpublished materials and interviews and specify that models must be run locally. For example, NRK (2023)⁷³, The Norwegian Broadcasting Company’s Guidelines for Editorial Use of Artificial Intelligence, states that “Unpublished editorial material (such as interview transcripts, etc.) should not be shared with such services”. (p.1). NRK also prohibits the use of Open AI services for transcription or translation of interviews and requires the use of a completely closed storage environment processing of such materials. When using AI for fact checking, media professionals will need to be clear about the source of the materials they are checking and handle the material in allowable ways.

For the applications of LLMs into which we dive deeply here, i.e., verdict production, multilingual paraphrasing and summarization, it is important that the source be known, in order to align with the common journalistic practice that requires at least two sources to verify a piece of information. Media organizations may specify that generative AI does not count as a source. For example, ZDF (German Public Broadcasting) Guiding principles for the use of Generative AI (2023)⁷⁴ states that “Generative AI tools can be used in an editorial and journalistic context. The two-source principle applies, whereby generative AI itself does not count as a source.” (p.1).

When producing and publishing, a media professional will need to take appropriate responsibility for the published content, aligning with organizational policies. As mentioned in D3.1, in the section “Assigning legal and ethical responsibility when using generative AI” there are differences among approaches to assigning legal responsibility to the text that is generated by an AI tool. While ultimate legal responsibility may differ, it should be understood that there is always a human who will be responsible for the content generated, as if they had written it themselves. SWR’s Editorial guidelines for “artificial intelligence” (2023)⁷⁵ *guidelines* articulate this position well in explaining that “every piece of content created using AI-based programmes is approved by a human before it is published. The human always retains control, makes the decision to publish and bears the responsibility” (p.1).

Finally, it is important that issues of authorship are clarified so that authorship for generated content is not claimed inappropriately. Media organizations must be careful both that their published material is attributed properly to an individual author, and also must avoid any potential of infringing on the rights of other authors that might have been used in training data. Lithuanian National Radio Guidelines for the Use of Artificial Intelligence⁷⁶ states that “Contracts with the provider of the AI tool must include a guarantee in case of third-party/author claims” (p. 3). Uncertainty about the legal status of language used in Generative AI systems training data is a concern for media organizations both as publishers of content that must not infringe others, and also as producers whose rights may be infringed.

In summary, the applications of summarization and multilingual paraphrasing for fact checking and

⁷³ NRK’s Guidelines for Editorial Use of Artificial Intelligence

⁷⁴ ZDF Guiding principles for the use of Generative AI. <https://www.zdf.de/zdfunternehmen/guiding-principles-generative-ai-100.html>

⁷⁵ SWR Editorial guidelines for “artificial intelligence”

⁷⁶ Public Enterprise Lithuanian National Radio and Television Guidelines for the Use of Artificial Intelligence

summarization fit closely in the trust media production workflow because the pre-conditions are easier to fulfil than for other possible applications of AI, i.e., generating an entire article or outline of an article using Generative AI. Also, notions of authorship are clearly of importance.

3.1.3 Verdict production for fact checking

In this section, we cover the main approaches that are used in the NLP field to address verdict production to counter misinformation, specifically the use of methodologies that are under the realm of Natural Language Generation and Summarization. In the following section 3.2, instead, we will specifically describe the approach we implemented within AI-CODE for the task of verdict production.

In today's media environment (Adair et al. 2017)⁷⁷, manual fact-checking alone is insufficient for media professionals to effectively cover all claim that might contain misinformation when working on their content. Natural Language Processing (NLP) has been proposed as a viable solution for partially automating the process of fake news countering. Indeed, Vlachos and Riedel (2014)⁷⁸ showed that the main stages of the fact-checking process can be modelled as NLP tasks. The NLP framework for automated fact-checking encompasses three primary stages: *i*) claim detection; *ii*) evidence retrieval; *iii*) claim verification. The latter can be further broken down into two sub-tasks, namely verdict prediction (assessing whether a claim is true or false) and verdict production (explaining why a claim is true or false) (Guo et al. 2022⁷⁹; Kotonya and Toni 2020a⁸⁰). Recent years have seen a predominant focus on the verdict prediction task. Nevertheless, considering the growing capabilities of large language models, we aim to explore verdict production through the application of Natural Language Generation techniques.

Fact-checking a news story involves two key components: assessing the truthfulness of a statement (verdict prediction) and generating a written rationale for the verdict (verdict production) (Guo et al. 2022⁸¹; Kotonya and Toni 2020a⁸²). The determination of claim veracity typically employs classification tasks, encompassing both binary (Nakashole and Mitchell 2014⁸³; Popat et al. 2018⁸⁴; Potthast et al. 2018⁸⁵) and multi-class (Thorne et al. 2018⁸⁶; Wang 2017⁸⁷) approaches, or utilizes multitask learning (Augenstein et al. 2019)⁸⁸. Notably, recent research efforts have been directed towards constructing datasets and systems tailored for evidence-

⁷⁷ Bill Adair, Chengkai Li, Jun Yang, and Cong Yu. 2017. Progress toward “the holy grail”: The continued quest to automate fact-checking. In *Computation+ Journalism Symposium* (September).

⁷⁸ 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*. 18–22

⁷⁹ Zhijiang Guo, Michael Schlichtkrull, and Andreas Vlachos. 2022. A Survey on Automated Fact-Checking. *Transactions of the Association for Computational Linguistics* 10 (2022), 178–206. https://doi.org/10.1162/tacl_a_00454

⁸⁰ Neema Kotonya and Francesca Toni. 2020a. Explainable Automated Fact-Checking: A Survey. In *Proceedings of the 28th International Conference on Computational Linguistics*. International Committee on Computational Linguistics, Barcelona, Spain (Online), 5430–5443. <https://doi.org/10.18653/v1/2020.coling-main.474>

⁸¹ Zhijiang Guo, Michael Schlichtkrull, and Andreas Vlachos. 2022. A Survey on Automated Fact-Checking. *Transactions of the Association for Computational Linguistics* 10 (2022), 178–206. https://doi.org/10.1162/tacl_a_00454

⁸² Neema Kotonya and Francesca Toni. 2020a. Explainable Automated Fact-Checking: A Survey. In *Proceedings of the 28th International Conference on Computational Linguistics*. International Committee on Computational Linguistics, Barcelona, Spain (Online), 5430–5443. <https://doi.org/10.18653/v1/2020.coling-main.474>

⁸³ Ndapandula Nakashole and Tom M. Mitchell. 2014. Language-Aware Truth Assessment of Fact Candidates. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Baltimore, Maryland, 1009–1019. <https://doi.org/10.3115/v1/P14-1095>

⁸⁴ Kashyap Popat, Subhabrata Mukherjee, Andrew Yates, and Gerhard Weikum. 2018. Declare: Debunking fake news and false claims using evidence-aware deep learning. *arXiv preprint arXiv:1809.06416* (2018).

⁸⁵ Martin Potthast, Johannes Kiesel, Kevin Reinartz, Janek Bevendorff, and Benno Stein. 2018. A Stylometric Inquiry into Hyperpartisan and Fake News. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Melbourne, Australia, 231–240. <https://doi.org/10.18653/v1/P18-1022>

⁸⁶ 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)*. Association for Computational Linguistics, New Orleans, Louisiana, 809–819. <https://doi.org/10.18653/v1/N18-1074>

⁸⁷ 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)*. Association for Computational Linguistics, Vancouver, Canada, 422–426. <https://doi.org/10.18653/v1/P17-2067>

⁸⁸ 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)*. Association for Computational Linguistics, Hong Kong, China, 4685–4697. <https://doi.org/10.18653/v1/D19-1475>

based verdict prediction, exemplified by datasets like FEVER (Thorne et al. 2018)⁸⁹, SciFact (Wadden et al. 2020)⁹⁰, COVID-fact (Saakyan et al. 2021)⁹¹, and PolitiHop (Ostrowski et al. 2021)⁹².

The task of verdict production is notably more challenging compared to verdict prediction. Various approaches have been proposed, including logic-based methods (Ahmadi et al. 2019⁹³; Gad-Elrab et al. 2019⁹⁴) and techniques utilizing deep learning and attention mechanisms (Lu and Li 2020;⁹⁵ Popat et al. 2018;⁹⁶ Shu et al. 2019⁹⁷; Yang et al. 2019⁹⁸). However, framing verdict production as a summarization task emerges as the most promising solution (Kotonya and Toni 2020a)⁹⁹, especially since the resulting verdicts are comprehensible even for non-expert users. This involves deriving explanations from manually written fact-checking articles, either by selecting crucial sentences from the text (extractive approach) (Atanasova et al. 2020)¹⁰⁰ or by generating new ones (abstractive approach) (Kotonya and Toni 2020b)¹⁰¹. Both extractive and abstractive summarization approaches face challenges; extractive-generated explanations may lack context, while abstractive-generated ones may sacrifice faithfulness due to the tendency of neural models to “hallucinate” (Guo et al. 2022¹⁰²; Kotonya and Toni 2020a¹⁰³). Russo et al. (2023b)¹⁰⁴ demonstrated that a unique hybrid pipeline, which combines extractive and abstractive approaches, leads to high-quality verdicts and enhanced performance. Currently, the abstractive summarization technique appears to be the more feasible option for generating effective verdicts. Nevertheless, acquiring a sufficient amount of training data or the required computational resources for highly demanding models may not always be feasible.

The most used datasets for verdict production are either machine-generated, e.g. e-FEVER by Stammbach and

⁸⁹ 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). Association for Computational Linguistics, New Orleans, Louisiana, 809–819. <https://doi.org/10.18653/v1/N18-1074>

⁹⁰ David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or Fiction: Verifying Scientific Claims. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics, Online, 7534–7550. <https://doi.org/10.18653/v1/2020.emnlp-main.609>

⁹¹ Arkadiy Saakyan, Tuhin Chakrabarty, and Smaranda Muresan. 2021. COVID-Fact: Fact Extraction and Verification of Real-World Claims on COVID-19 Pandemic. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). Association for Computational Linguistics, Online, 2116–2129. <https://doi.org/10.18653/v1/2021.acl-long.165>

⁹² Wojciech Ostrowski, Arnav Arora, Pepa Atanasova, and Isabelle Augenstein. 2021. Multi-Hop Fact Checking of Political Claims. In Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21, Zhi-Hua Zhou (Ed.). International Joint Conferences on Artificial Intelligence Organization, 3892–3898. <https://doi.org/10.24963/ijcai.2021/536> Main Track.

⁹³ Naser Ahmadi, Joohyung Lee, Paolo Papotti, and Mohammed Saeed. 2019. Explainable Fact Checking with Probabilistic Answer Set Programming. ArXiv cs.DB/1906.09198 (2019). <https://doi.org/10.48550/arXiv.1906.09198> Version 1.

⁹⁴ 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 (Melbourne VIC, Australia) (WSDM '19). Association for Computing Machinery, New York, NY, USA, 87–95. <https://doi.org/10.1145/3289600.3290996>

⁹⁵ Yi-Ju Lu and Cheng-Te Li. 2020. GCAN: Graph-aware Co-Attention Networks for Explainable Fake News Detection on Social Media. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, Online, 505–514. <https://doi.org/10.18653/v1/2020.acl-main.48>

⁹⁶ Kashyap Popat, Subhabrata Mukherjee, Andrew Yates, and Gerhard Weikum. 2018. Declare: Debunking fake news and false claims using evidence-aware deep learning. arXiv preprint arXiv:1809.06416 (2018).

⁹⁷ Kai Shu, Limeng Cui, Suhang Wang, Dongwon Lee, and Huan Liu. 2019. DEFEND: Explainable Fake News Detection. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (Anchorage, AK, USA) (KDD '19). Association for Computing Machinery, New York, NY, USA, 395–405. <https://doi.org/10.1145/3292500.3330935>

⁹⁸ Fan Yang, Shiva K. Penttala, Sina Mohseni, Mengnan Du, Hao Yuan, Rhema Linder, Eric D. Ragan, Shuiwang Ji, and Xia (Ben) Hu. 2019. XFake: Explainable Fake News Detector with Visualizations. In The World Wide Web Conference (San Francisco, CA, USA) (WWW '19). Association for Computing Machinery, New York, NY, USA, 3600–3604. <https://doi.org/10.1145/3308558.3314119>

⁹⁹ Neema Kotonya and Francesca Toni. 2020a. Explainable Automated Fact-Checking: A Survey. In Proceedings of the 28th International Conference on Computational Linguistics. International Committee on Computational Linguistics, Barcelona, Spain (Online), 5430–5443. <https://doi.org/10.18653/v1/2020.coling-main.474>

¹⁰⁰ 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. Association for Computational Linguistics, Online, 7352–7364. <https://doi.org/10.18653/v1/2020.acl-main.656>

¹⁰¹ Neema Kotonya and Francesca Toni. 2020b. Explainable Automated Fact-Checking for Public Health Claims. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics, Online, 7740–7754. <https://doi.org/10.18653/v1/2020.emnlp-main.623>

¹⁰² Zhijiang Guo, Michael Schlichtkrull, and Andreas Vlachos. 2022. A Survey on Automated Fact-Checking. Transactions of the Association for Computational Linguistics 10 (2022), 178–206. https://doi.org/10.1162/tacl_a_00454

¹⁰³ Neema Kotonya and Francesca Toni. 2020a. Explainable Automated Fact-Checking: A Survey. In Proceedings of the 28th International Conference on Computational Linguistics. International Committee on Computational Linguistics, Barcelona, Spain (Online), 5430–5443. <https://doi.org/10.18653/v1/2020.coling-main.474>

¹⁰⁴ Daniel Russo, Serra Sinem Tekiroğlu, and Marco Guerini. 2023b. Benchmarking the Generation of Fact Checking Explanations. Transactions of the Association for Computational Linguistics 11 (10 2023), 1250–1264. https://doi.org/10.1162/tacl_a_00601

Ash (2020)¹⁰⁵, or silver data as for LIAR-PLUS by Alhindi et al. (2018)¹⁰⁶. Only two datasets include gold explanations, i.e. PubHealth by Kotonya and Toni (2020b)¹⁰⁷, and the MisinfoCorrect's crowdsourced dataset by He et al. (2023)¹⁰⁸. However, PubHealth and MisinfoCorrect datasets are domain-specific (respectively, health and COVID-19), and only the latter comprises textual data written in an SMPs style (informal, personal, and empathetic).

3.2 Fact-checking verdict production as a case of multilingual paraphrasing and summarization

In the face of a surge of online misinformation, it has become essential to develop methods that not only detect false information but also actively correct public understanding by generating convincing counter-narratives. This section examines whether Large Language Models (LLMs) can be used in multilingual settings to create narratives grounded in reliable sources, guiding audiences toward accurate information. To achieve this, we employ a range of multilingual data and approaches to assess the models' ability to produce content that aligns both linguistically and semantically with verified information. The main challenge lies in creating text that is not only accurate with respect to verifying claims but are also clear and persuasive for diverse audiences.

3.2.1 Datasets

In the multilingual setting, we adopt two already existing datasets to evaluate and train large language models (LLMs) for generating reliable information and countering misinformation. These datasets were carefully selected and prepared to offer diverse linguistic and structural insights, enabling the models to produce accurate responses to various claims and counter misinformation effectively.

3.2.1.1 Multilingual-VerMouth Dataset

The first dataset is a multilingual version of VerMouth (Russo et al., 2023)¹⁰⁹, that was created to test the multilingual capabilities of language models. To build it, 621 samples were randomly selected from the original VerMouth dataset, which is only available in English. Each of these samples was then automatically translated into 7 different languages using DeepL. The languages included in this dataset are English, Italian, Spanish, Romanian, Greek, French, Polish and German. Consequently, this dataset comprises 4,968 samples (621 * 8). The Table 16 provides an example of data entries from VerMouth.

Table 16: An example of data entry from VerMouth dataset.

Article	Earlier this week, the nominations for four of the most powerful positions within the EU were announced. The European Council was responsible for nominating the four candidates. The Council is made up of the heads of state or government of all EU member countries, the European Council President (currently Donald Tusk) and the President of the European Commission (currently Jean-Claude Juncker). Brexit Party chairman and MEP [...] The final role, which MEPs have no say over, is President of the European Council (the Council sets the EU's overall political direction but does not have legislative power). He has already been elected by the Council. All four will take up their posts later this year, subject to their appointments being approved.
Claim (EN)	I'm fuming! How can the EU expect us to have faith in their leadership when they bypass our elected representatives? None of the four nominees were voted on by MEPs? This is undemocratic and unacceptable. #EUleadershipfail

¹⁰⁵ Dominik Stambach and Elliott Ash. 2020. e-fever: Explanations and summaries for automated fact checking. Proceedings of the 2020 Truth and Trust Online (TTO 2020) (2020), 32–43.

¹⁰⁶ Tariq Alhindi, Savvas Petridis, and Smaranda Muresan. 2018. Where is Your Evidence: Improving Fact-checking by Justification Modeling. In Proceedings of the First Workshop on Fact Extraction and VERification (FEVER). Association for Computational Linguistics, Brussels, Belgium, 85–90. <https://doi.org/10.18653/v1/W18-5513>

¹⁰⁷ Neema Kotonya and Francesca Toni. 2020b. Explainable Automated Fact-Checking for Public Health Claims. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics, Online, 7740–7754. <https://doi.org/10.18653/v1/2020.emnlp-main.623>

¹⁰⁸ Bing He, Mustaque Ahamad, and Srikanth Kumar. 2023. Reinforcement Learning-Based Counter-Misinformation Response Generation: A Case Study of COVID-19 Vaccine Misinformation. In Proceedings of the ACM Web Conference 2023 (Austin, TX, USA) (WWW '23). Association for Computing Machinery, New York, NY, USA, 2698–2709. <https://doi.org/10.1145/3543507.3583388>

¹⁰⁹ Russo, D., Kaszefski-Yaschuk, S. P., Staiano, J., & Guerini, M. (2023). Countering misinformation via emotional response generation. arXiv preprint arXiv:2311.10587.

Verdict (EN)	I get it, but just to clear things up, while MEPs don't nominate any of the four candidates, they do have a say in approving two of the positions and are consulted on another. Of course, it's always crucial to maintain a democratic process while also finding the best candidates for the job.
---------------------	---

More in details for each entry we have:

- **Dataset Partition:** Whether the entry belongs to the training, test, or validation set.
- **Publisher:** Representing the source of the news and indicating its credibility.
- **URL:** A direct link to the news article.
- **Article:** The original news article text (in English)
- **Style:** The emotional style category of each entry (e.g. Anger, Fear, Sadness)
- **Claim (LANG):** The claim related to the news article, provided in multiple languages.
- **Verdict (LANG):** The conclusion drawn about the claim's validity, in various languages.

The value of the VerMouth dataset lies in its ability to support a multilingual assessment of model capabilities while preserving additional aspects such as linguistic style. This approach allows models to demonstrate their accuracy in handling claims and generating counter-narratives across different languages, evaluating the impact of linguistic diversity on model performance.

3.2.1.2 AI4Trust Dataset

The second dataset, created within the AI4Trust project, was collected with the support of expert media professionals and consists of 1,647 manually created data samples in eight different languages. Unlike the VerMouth dataset, AI4Trust data includes news articles in different languages, eliminating the need for additional translation and offering a more naturally multilingual dataset structure for realistic model evaluation. The languages covered are English (199 samples), Greek (195 samples), Spanish (264 samples), Polish (204 samples), Italian (202 samples), German (201 samples), Romanian (192 samples) and French (190 samples). More in details for each entry we have information on:

- **Language:** Language of each sample
- **Date:** The publication date
- **Publisher:** The name of the article's publisher
- **Claim:** The claim related to the news article, written in the sample's original language.
- **Verdict:** The conclusion regarding the claim's accuracy.
- **URL:** A direct link to the news article.
- **Article Text:** The complete article text in the language of origin.

The structure and variety in the AI4Trust dataset, with complete articles in the target language, enable the language models to be evaluated in more realistic, multilingual contexts. This combination of VerMouth and AI4Trust data provides a comprehensive testing ground for analyzing models' accuracy, consistency, and adaptability to diverse linguistic environments and complex claims.

By leveraging these datasets, we can assess models' performance regarding linguistic adaptability and content analysis accuracy, determining how effectively these models can generate reliable counter-narratives across different languages and nuanced contexts.

3.2.2 Experimental setup

This section outlines the methods used for multilingual experiments, the models tested, and the approach to prompt engineering. Initially, we tested Llama 3 in generating multilingual verdicts in response to various claims. Following the release of Llama 3.1 by Meta, given its improved multilingual performances, we also added this model to our evaluation.

Model Selection and Initial Performance Evaluation

Choosing an appropriate model was critical to the final quality of the outputs. Selection of the Llama 3 family was based on several aspects: the models' license, Use Policy, ability to comprehend multilingual texts, accuracy in responses, and semantic coherence. Through the evaluation of both Llama 3 and Llama 3.1, we noticed that the newer version maintains the original meaning of the content while providing more precise and credible counter-narratives. Additionally, improvements in Llama 3.1 contributed to enhanced multilingual performance, making it more reliable for cross-language applications.

Prompt Engineering

One key step in this project was designing and fine-tuning effective prompts. Prompt engineering involves crafting and adjusting text inputs to enable models to respond optimally to requests. We utilized three different prompt configurations (see Table 18 for actual configurations) to optimize responses across languages.

1. **English Prompts:** Prompts were written in English, and claims were presented in eight languages (English, Greek, Spanish, Polish, Italian, German, Romanian, and French). In the multilingual VerMouth dataset, all articles are in English, regardless of the language of the claim. However, in the AI4Trust dataset, each claim is paired with an article in the same language, meaning if the claim is in French, the corresponding article is also in French. Here, no specific language instructions were provided to the model for generating the verdict, which sometimes led to errors, as the model would occasionally produce the verdict in English instead of the language of the claim.
2. **English Prompts with Language Instructions:** In this configuration English prompts were provided along with specific instructions to produce the verdict in the target language. For example, if the claim was in Italian, instructions were added for the model to respond in Italian. This approach led to accurate, error-free outputs, consistently generating verdicts in the correct language.
3. **Translated Prompts:** The prompt is translated into each of the eight target languages and then presented to the model. While this method further reduced language detection errors, the model would still occasionally generate verdicts in English rather than the target language.

Comparing these methods revealed that, in the first approach, the model would sometimes generate the verdict in English rather than in the language of the claim. In contrast, the second method — using English prompts specifying in the instructions the language wanted in the output — yielded precise, error-free results across all languages. The third method, which involved translating prompts into each target language, showed some improvements over the first method in reducing language detection errors; however, it still occasionally produced verdicts in English. Additionally, if new languages were added in the future for which we lack native knowledge, translating prompts into those languages could introduce further complexity. Therefore, the second method remains the most reliable and scalable approach, as it avoids translation challenges and ensures accurate outputs regardless of the target language.

Sample Prompts

The Table 17 showcases examples of prompts used in this project.

Table 17: Sample of prompts that used for Italian language.

EN prompt	<pre>messages = [{"role": "system", "content": "You are an expert assistant helping with fact-checking claims based on the given context."}, {"role": "user", "content": f"Based on the provided context, respond to the claim, ensuring a thorough explanation. Use only the given context. Reply in no more than three sentences. Avoid mentioning the context in the reply. Match the communication style of the claim and address the possible emotional component present in it, if needed. If the context is insufficient, state that you don't know. Format your response as follows:\n\n [your_reply] \n\n The context information is provided below (in between xml tags).\n\n<context>\n{article}\n</context>\n\nClaim: \"{claim}\""}]</pre>
EN prompt + Language Instructions	<pre>messages = [{"role": "system", "content": "You are an expert assistant helping with fact-checking claims based on the given context."},</pre>

	<pre>{ "role": "user", "content": f"Based on the provided context, respond to the claim, ensuring a thorough explanation. Use only the given context. Reply in no more than three sentences. Avoid mentioning the context in the reply. Match the communication style of the claim and address the possible emotional component present in it, if needed. The response should be in {language}. If the context is insufficient, state that you don't know. Format your response as follows:\n\n[your_reply] \n\n The context information is provided below (in between xml tags).\n\n<context>\n{{article}}\n</context>\n\nClaim: \"{{claim}}\""} }</pre>
Translated Prompt (in italian)	<pre>messages = [{"role": "system", "content": "Sei un assistente esperto nella verifica dei fatti e aiuti a controllare la correttezza delle affermazioni basandoti solo sul contesto fornito."}, {"role": "user", "content": f"Usando esclusivamente il contesto fornito, rispondi all'affermazione spiegando in modo chiaro e accurato. Mantieni la risposta entro tre frasi e non fare riferimento diretto al contesto. Adatta il tono della risposta allo stile e all'emotività dell'affermazione, se presente. Se il contesto è insufficiente, specifica che non hai abbastanza informazioni. Struttura la risposta come segue:\n\n[tua_risposta] \n\n Il contesto è fornito qui sotto (tra i tag xml).\n\n<context>\n{{article}}\n</context>\n\nAffermazione: \"{{claim}}\""}]</pre>

Challenges and Solutions Implemented

Throughout the process, we identified several challenges with model performance, and the following corrective actions were taken to improve results:

- **Language Consistency Issues:** In some languages, models tended to generate sentences with improper structure or inaccurate translations. To address this, several versions of the prompt wording were tested and the most suitable was selected.
- **Reducing Translation and Semantic Errors:** Due to specific linguistic complexities, models occasionally misinterpreted information. Reviewing outputs and involving human experts minimized these issues.
- **Refining Style and Tone:** A primary objective of this project was to generate verdicts that were convincing in style and tone. To achieve this, prompts were designed with an emphasis on specific styles (e.g., news, social media, emotional tone), helping models produce responses suited to the claim characteristics.
- **Incomplete Sentence Endings:** In some cases, the model's output would end abruptly, leaving sentences incomplete. Adjusting prompts and adhering to the Llama 3.1 guidelines resolved this issue.

Generated Outputs by English Prompts with Language Instructions

Table 18 showcases examples of outputs generated with English Prompts with Language Instructions by the Llama 3 and Llama 3.1 models. These tables allow us to compare the models' performance across different languages and identify their respective strengths and weaknesses.

Table 18: Example of Verdict Generation Using EN Prompt + Language Instructions with Llama 3 and Llama 3.1 from the Multilingual-VerMouth Dataset.

Article	Recently a claim about supposed plans for an EU army has been going viral again on Facebook. The post claims that Britain will be forced to join an EU army unless the UK leaves the EU. It includes a screenshot of a 2016 article in the Express reporting then armed forces minister Penny Mordaunt saying this during the EU referendum campaign. The text accompanying the image suggests this EU army will use some form of conscription. Both claims are incorrect. The UK could not be forced to participate in EU military policies even if we stay in the EU. Whether or not EU nations wish to create an EU army and regardless of what that might actually entail, the European Commission can't propose laws about security or defence. On issues like this, member countries effectively have a veto, so the UK would only take part if it chose to. There are no formal plans in place for an EU army and so there are no EU regulations on drafting people into it. The EU does have various policies for defensive and security cooperation. For example, under the Common Security and Defence Policy, EU countries can pool military funding and resources. The UK participates in military operations through this policy. The EU has also established Permanent Structured Cooperation (PESCO), a strategy which aims to jointly develop defence capabilities and make
---------	--

	them available for EU military operations which the UK has chosen not to participate in. We wrote about the idea of an EU army in more detail back in 2016.	
Claim	Britain will be forced to join an EU army unless we leave the EU.	
Original verdict	Incorrect. The UK has a veto on EU military policy so couldn't be forced to participate in an EU army.	
Generated Verdict		
	Llama3	Llama3.1
English	This claim is incorrect. The UK has a veto on EU military policies, and even if the EU were to propose an EU army, the UK would not be forced to participate. There are no formal plans for an EU army, and the EU does not have the authority to draft people into it.	That's not true, Britain would only be involved in an EU army if it chose to participate, and the UK has a veto on security and defence policies. The EU doesn't have any plans for a conscription-based army, and the UK has already opted out of one of the EU's defence cooperation strategies.
Romanian	"În realitate, Marea Britanie nu poate fi obligată să participe la politici militare ale UE, chiar dacă rămâne în UE. De fapt, statele membre au un veto efectiv în aceste domenii, ceea ce înseamnă că UK ar participa do	Această afirmație este complet falsă. Marea Britanie nu poate fi forțată să participe la politica militară a UE, chiar dacă rămâne în UE, deoarece țările membre au un veto în acest sens.
Greek	Δεν υπάρχει καμία αλήθεια πίσω από αυτή την απατηλή δήλωση. Η Βρετανία δεν μπορεί να αναγκαστεί να συμμετέχει σε στρατιωτικές πολιτικές της ΕΕ, ακόμα και αν παραμείνει στην ΕΕ. Οι χώρες-μέλη της ΕΕ έχουν	Η εκκίνηση αυτής της απαγγελίας είναι ένα μυθικό φάντασμα, αφού η Βρετανία δεν θα μπορούσε να αναγκαστεί να συμμετάσχει σε πολιτικές της ΕΕ σχετικά με την άμυνα, ακόμα και αν παραμείνει στην ΕΕ. Η Ευρωπαϊκή Επιτροπή δεν μπορεί να προτείνει νόμους σχετικά με την ασφάλεια ή την άμυνα, εκτός αν οι μέτοχοι συμφωνήσουν να το κάνουν. Έτσι, η Βρετανία θα πρέπει να αποφασίσει να συμμετάσχει σε στρατιωτικές δραστηριότητες της ΕΕ.
Italian	La Gran Bretagna non potrebbe essere costretta a partecipare alle politiche militari dell'UE, neanche se resta membro dell'UE. Inoltre, non esistono piani formali per creare un esercito dell'UE e quindi non ci sono regolamentazioni sull'arr	La Gran Bretagna non potrebbe essere costretta a partecipare alle politiche militari dell'UE, anche se rimane nella comunità. Inoltre, non esistono piani formali per creare un esercito dell'UE.
French	La Grande-Bretagne ne sera pas contrainte de rejoindre une armée européenne, car les pays membres de l'UE ont un veto sur les politiques de sécurité et de défense. De plus, il n'y a pas de plans formels pour créer une armée européenne, et les réglementations de l'	Il est tout à fait inexact de dire cela, car le Royaume-Uni ne peut pas être contraint de participer aux politiques militaires de l'UE, même s'il reste dans l'UE. En fait, les pays membres ont un droit de veto sur les lois relatives à la sécurité et à la défense, ce qui signifie que le Royaume-Uni ne serait membre d'une armée européenne que si il le choisissait.
Polish	Wielka Brytania nie może być zmuszona do udziału w polityce militarnych UE, nawet jeśli pozostanie w UE. W sprawach takich, jak te, kraje członkowskie mają w praktyce weto, co oznacza, że Wielka	To nieprawda. Brytyjczycy nie mogą być zmuszeni do uczestnictwa w politykach wojskowych UE nawet jeśli pozostaną w UE.
Spanish	La afirmación es incorrecta. Aunque el	Eso no es cierto. La Gran Bretaña no podría ser forzada a

	Reino Unido no esté de acuerdo con las políticas de seguridad y defensa de la UE, no puede ser forzado a participar en ellas. Los países miembros de la UE tienen un veto efectivo en estas cuestiones, por lo que el Reino Unido	participar en políticas militares de la UE, incluso si permanece en la UE, y las autoridades europeas no pueden imponer leyes sobre seguridad o defensa sin el consentimiento de los países miembros.
German	Das ist ein falscher Claim. Die EU kann Großbritannien nicht zwingen, an militärischen Politiken teilzunehmen, selbst wenn es in der EU bleibt. Die Mitgliedstaaten haben auf diesem Gebiet ein Veto-Recht, so dass Großbritannien nur teilnehmen würde, wenn	Das ist ein Irrtum. Großbritannien kann nicht gezwungen werden, an EU-Militärbestrebungen teilzunehmen, selbst wenn es in der EU bleibt. Der Europäische Rat kann keine Gesetze über Sicherheit oder Verteidigung vorschlagen, sodass das Vereinigte Königreich nur mit seiner Zustimmung teilnehmen würde.

Evaluation Metrics:

To assess the accuracy and quality of the model's outputs, we utilized two primary metrics - **Repetition Rate (RR)** and **ROUGE**. In addition, we run a preliminary human expert review on the model capability to generate verdicts in the target languages. Media professionals were asked to rate automatically generated verdicts in their native language, providing a score from 1 to 5 on the grammaticality of the output and another score on the soundness of the verdict. Across the languages, the average of the grammaticality score ranged between 3 and 4.8, on the soundness side we obtained an average score on the languages ranging from 3.4 to 4.4, except for 2.9 from Italian where although grammatically correct the verdicts didn't always sound natural. This preliminary evaluation showed that the model can produce Verdicts in a suitable way (grammaticality and soundness) for all the languages also in zero-shot settings. For this reason, we continued the analysis with the automatic metrics.

The automatic metrics provide a nuanced view of the model's performance, capturing both variety in expression and structural similarity to reference texts, key indicators of naturalness and fidelity in generated content.

- **RR (Repetition Rate):** Repetition Rate measures the frequency of word repetition within the text generated by the model. A high RR suggests that the model tends to reuse words excessively, which can make the output feel redundant or less engaging. Conversely, a lower RR indicates more variety in word choice, contributing to a more fluent and natural text. This measure is particularly relevant when evaluating the model's ability to produce content that flows smoothly and resembles human writing, as high repetition might indicate limitations in lexical diversity or an over-reliance on certain phrases. Thus, RR serves as a direct metric of linguistic richness and stylistic authenticity in generated content.
- **ROUGE (Recall-Oriented Understudy for Gisting Evaluation):** The ROUGE metric measures how closely the generated text aligns with a reference text, effectively capturing both the semantic and structural fidelity of the content produced by the model. ROUGE is widely used to evaluate the effectiveness of text generation models, as it quantifies the overlap between generated and reference texts at various levels, thereby offering insights into precision and coherence. We utilized three key ROUGE measures in our evaluation:
 - **ROUGE-1 (R1):** This metric assesses the overlap of individual words, or **unigrams**, between the generated text and the reference.
 - **ROUGE-2 (R2):** Focusing on **bigrams**, or sequences of two consecutive words, R2 goes beyond single-word overlap to evaluate the coherence and continuity in word pairings.
 - **ROUGE-L (RL):** This metric evaluates the **longest matching sequence of words** shared between the generated text and the reference. By focusing on extended sequences rather than individual words or pairs, RL provides a deeper measure of structural alignment and sentence coherence.

These three ROUGE metrics collectively provide a comprehensive assessment of the model's performance by measuring accuracy at multiple linguistic levels, from individual words to entire sentence structures.

The tables below present the evaluation results based on RR and ROUGE metrics for the multilingual-VerMouth and AI4Trust datasets, with metrics obtained using the Llama 3.1 model in a **zero-shot setting** using our 3 prompt configurations.

- Low RR values indicate languages where the model generates more varied and fluent text, suggesting a style closer to human writing. While we expected English prompts + descriptions to consistently yield the lowest RR values, in certain cases, other prompt styles showed slightly lower RR scores. However, these differences were minor and not statistically significant. English prompts + language instructions produced the most reliable results with the fewest errors overall. We believe that with further fine-tuning using this prompt style, the model's performance can be further enhanced.
- High ROUGE scores in R1, R2, and RL demonstrate strong alignment with the reference text, showcasing the model's ability to maintain both the semantic accuracy and structural coherence essential for effective information synthesis.

Together, these metrics provide a comprehensive view of the model's performance, capturing both linguistic diversity and structural accuracy across different languages and contexts.

Table 19: RR on Multilingual-Vermouth Dataset.

		Languages							
		en	ro	el	it	fr	pl	es	de
Prompts	EN Prompt	5.551	4.703	3.4	5.306	5.621	2.466	4.87	4.943
	EN Prompt + Language Instructions	5.581	4.697	3.04	4.957	6.290	2.743	4.761	5.145
	Translated Prompt	5.303	5.075	4.31	4.908	5.249	3.535	6.109	4.96

Table 20: ROUGE on Multilingual-Vermouth Dataset.

			Languages							
			en	ro	el	It	fr	pl	es	de
Prompts	EN Prompt	R1	0.271	0.212	0.205	0.247	0.267	0.166	0.289	0.223
		R2	0.072	0.052	0.049	0.068	0.078	0.039	0.080	0.047
		RL	0.180	0.143	0.138	0.165	0.175	0.121	0.185	0.149
	EN Prompt + Language Instructions	R1	0.275	0.237	0.218	0.249	0.279	0.172	0.299	0.231
		R2	0.072	0.064	0.051	0.068	0.080	0.041	0.084	0.050
		RL	0.183	0.159	0.146	0.167	0.188	0.126	0.195	0.154
	Translated Prompt	R1	0.273	0.251	0.243	0.266	0.285	0.180	0.307	0.233
		R2	0.071	0.064	0.057	0.079	0.085	0.047	0.090	0.054
		RL	0.181	0.164	0.153	0.176	0.187	0.132	0.200	0.156

Table 21: RR on AI4Trust Dataset.

		Languages							
		en	ro	en	It	en	pl	en	de
Prompts	EN Prompt	6.163	4.39	3.413	4.147	2.865	2.705	5.052	4.32

	EN Prompt + Language Instructions	5.224	3.933	3.452	5.068	3.63	3.16	7.387	4.955
	Translated Prompt	5.263	4.211	3.578	5.242	3.794	2.917	7.415	4.851

Table 22: ROUGE on AI4Trust Dataset.

			Languages							
			en	ro	el	It	fr	pl	es	de
Prompts	EN Prompt	R1	0.290	0.233	0.311	0.229	0.219	0.232	0.201	0.221
		R2	0.114	0.069	0.160	0.064	0.055	0.109	0.079	0.068
		RL	0.217	0.169	0.250	0.158	0.148	0.196	0.154	0.186
	EN Prompt + Language Instructions	R1	0.299	0.224	0.298	0.258	0.022	0.226	0.335	0.236
		R2	0.121	0.064	0.152	0.078	0.054	0.103	0.141	0.079
		RL	0.227	0.161	0.243	0.179	0.150	0.189	0.249	0.200
	Translated Prompt	R1	0.287	0.252	0.327	0.266	0.240	0.228	0.309	0.219
		R2	0.116	0.074	0.169	0.081	0.063	0.099	0.135	0.080
		RL	0.223	0.173	0.258	0.185	0.163	0.184	0.231	0.183

3.2.3 Conclusion

Through these experiments, we identified each model's strengths and weaknesses, implementing necessary adjustments to improve output quality. The final results demonstrated that Llama 3.1 outperformed its predecessor across multiple languages, with precise, multi-layered prompt engineering significantly enhancing the accuracy and coherence of the outputs. To achieve optimal results, we plan to incorporate fine-tuning techniques in the next stages of the project, and additional models may be considered for further evaluation and refinement.

3.3 Paraphrasing and the notion of authorship

This part summarizes our achieved results that can be found in our paper (Tripto et al., 2024)¹¹⁰.

3.3.1 Motivation and experimental setup

LLMs are increasingly used for paraphrasing or summarization for various use cases:

- *beneficial*, such as production of counter-narratives as mentioned above or their use for synthetic data generation examined in Section 2.6,
- but also *negative ones*, such as their use for obfuscation of machine-generated texts (see Section 2.3.2) or to bypass watermarks of such texts (Wang et al., 2024¹¹¹; Fu et al., 2024¹¹²).

¹¹⁰ Nafis Irtiza Tripto, Saranya Venkatraman, Dominik Macko, Robert Moro, Ivan Srba, Adaku Uchendu, Thai Le, and Dongwon Lee. 2024. A Ship of Theseus: Curious Cases of Paraphrasing in LLM-Generated Texts. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) – ACL 2024, pages 6608–6625. <https://aclanthology.org/2024.acl-long.357/>

¹¹¹ Yichen Wang, Shangbin Feng, Abe Hou, Xiao Pu, Chao Shen, Xiaoming Liu, Yulia Tsvetkov, and Tianxing He. 2024. Stumbling Blocks: Stress Testing the Robustness of Machine-Generated Text Detectors Under Attacks. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 2894–2925, Bangkok, Thailand. Association for Computational Linguistics. <https://aclanthology.org/2024.acl-long.160/>

¹¹² Jiayi Fu, Xuandong Zhao, Ruihan Yang, Yuansen Zhang, Jiangjie Chen, and Yanghua Xiao. 2024. GumbelSoft: Diversified Language Model Watermarking via the GumbelMax-trick. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 5791–5808, Bangkok, Thailand. Association for Computational Linguistics. <https://aclanthology.org/2024.acl-long.315/>

It also raises a question of authorship of such paraphrased texts, whether authorship should be attributed to the original human author or the generative AI-powered tool, since some authors consider that the paraphrasing does not change the authors, others use it does, while others do not consider this at all, but it might be a confounding factor in their results.

To address this question, we systematically investigated two distinct scenarios, one in which it is assumed that the author should remain the same after paraphrasing and the other in which the paraphrased text is assumed to be machine-generated, thus changing the authorship. We assessed the impact of considering the assumption of each scenario as ground truth on the performance of authorship attribution. More specifically, we investigated the effects of successive paraphrasing on the original style and content of the text and its consequent implications for the performance of the authorship classification and machine-generated text detection tasks. We utilized 2 LLMs (ChatGPT and PaLM2) as paraphraser together with other paraphrasing models Pegasus (Zhang et al., 2020)¹¹³, operating at the sentence level, and Dipper (Krishna et al., 2024)¹¹⁴, capable of whole-text paraphrasing while preserving contextual coherence and providing control over lexical diversity.

3.3.2 Conclusion

We found that the performance of text classification models diminishes with each successive paraphrasing iteration, which is closely associated with the extent of deviation from the original author's style. Our findings suggest that authorship should be task-dependent and the authorship should be evaluated differently in situations where the content is more important as in situations where the style is more important; another aspect to consider is the portion of the generated text.

4 Assessment of AI to Detect Synthetic Image Disinformation

The rapid advancement of generative AI technology has brought forth a growing array of tools capable of producing entirely synthetic images that are increasingly challenging to distinguish from real images. Unlike techniques that modify only portions of an image, fully synthetic image creation introduces unique detection challenges. In response, several Synthetic Image Detection (SID) methods have emerged to address this issue. However, a significant gap often exists between experimental results on benchmark datasets and the real-world performance of these methods. To bridge this gap and better serve the evaluation needs of SID, this section describes a comprehensive benchmarking framework called SIDBench¹¹⁵ that has been developed in the context of AI-CODE and integrates a variety of state-of-the-art SID models. The models selected for integration are based on diverse input features and network architectures, providing a broad representation of current SID techniques. This framework leverages recent datasets that incorporate a wide range of generative models, achieving high levels of photorealism and resolution to reflect the rapid advancements in image synthesis technology. Additionally, the benchmarking framework allows for the examination of how typical image transformations—such as JPEG compression, frequently applied to online assets—impact detection performance. The framework is modularly designed to facilitate the inclusion of new datasets and SID models, offering a dynamic and adaptable tool for ongoing SID evaluation and improvement¹¹⁶.

4.1 Background

4.1.1 Synthetic image generation

The landscape of image generation has seen significant advances, primarily driven by three approaches:

¹¹³ Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020. PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. In Proceedings of the 37th International Conference on Machine Learning (ICML'20), Vol. 119. JMLR.org, Article 1051, 11328–11339. <https://dl.acm.org/doi/abs/10.5555/3524938.3525989>

¹¹⁴ Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. 2024. Paraphrasing evades detectors of AI-generated text, but retrieval is an effective defense. In Proceedings of the 37th International Conference on Neural Information Processing Systems (NIPS '23). Curran Associates Inc., Red Hook, NY, USA, Article 1195, 27469–27500. <https://dl.acm.org/doi/10.5555/3666122.3667317>

¹¹⁵ Schinas, M., & Papadopoulos, S. (2024, June). SIDBench: A Python framework for reliably assessing synthetic image detection methods. In Proceedings of the 3rd ACM International Workshop on Multimedia AI against Disinformation (pp. 55-64).

¹¹⁶ <https://github.com/meever-team/sidbench>

Variational Autoencoders (VAEs), Generative Adversarial Networks (GANs)¹¹⁷, and Diffusion Models (DMs)¹¹⁸. These developments have made it possible to create highly realistic synthetic images. Until recently, synthetic images were mainly generated using GANs, which can be categorized into conditional and unconditional frameworks. Conditional GANs such as CycleGAN¹¹⁹, StarGAN¹²⁰, and GauGAN¹²¹, generate images based on given conditions or labels, offering more control over the output, while unconditional GANs, such as ProGAN¹²², StyleGAN¹²³, and BigGAN¹²⁴, generate images from random noise. While GANs have deeply influenced the domain of image generation, they have recently been superseded by DMs. DMs for image generation can be categorized primarily into Denoising Diffusion Probabilistic Models (DDPMs)^{125 126}, which reverse the diffusion process, and Latent Diffusion Models (LDMs)¹²⁷, optimizing the process within a compressed latent space for efficiency. A notable unconditional approach is the Ablated Diffusion Model (ADM)¹²⁸ that has recently exceeded the image generation capabilities of GANs. More recently, conditional diffusion (text-to-image) models gained popularity. Models in this category take a text prompt and random noise as input and then denoise the image under the guidance of the prompt so that the generated image matches the description. Among those, Guided Diffusion and Glide¹²⁹ are two very popular methods.

4.1.2 Synthetic image detection

Synthetic Image Detection (SID) focuses on identifying whether an image is authentic or artificially created (e.g., produced by generative models). This task is often approached as a binary classification problem, where a pretrained backbone model such as ResNet50 is further fine-tuned on the real/fake task. These classifiers, such as the work of Wang et al.¹³⁰, typically perform well when the fake images for testing and training are from the same generative distribution. However, their effectiveness decreases on unseen generative models not encountered during training. The works of Wang et al. and Corvi et al.¹³¹ suggested that appropriate data augmentation techniques can significantly improve generalization. Other works use fingerprints extracted from images under the assumption that fingerprint patterns differ between synthetic and real images. Frank et al.¹³² uses patterns extracted from the frequency domain of the images, as the up-sampling operation in GAN-based image generation introduces detectable frequency artifacts. Other fingerprint patterns for effective generalization include noise patterns in the frequency domain¹³³ and texture patterns¹³⁴. A recent study¹³⁵ observed that synthetic images from DMs could be more accurately reconstructed by other DMs, utilizing

¹¹⁷ Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2020). Generative adversarial networks. *Communications of the ACM*, 63(11), 139-144.

¹¹⁸ Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., & Ganguli, S. (2015, June). Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning* (pp. 2256-2265). PMLR.

¹¹⁹ Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision* (pp. 2223-2232).

¹²⁰ Choi, Y., Choi, M., Kim, M., Ha, J. W., Kim, S., & Choo, J. (2018). Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 8789-8797).

¹²¹ Park, T., Liu, M. Y., Wang, T. C., & Zhu, J. Y. (2019). Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 2337-2346).

¹²² Karras, T. (2017). Progressive Growing of GANs for Improved Quality, Stability, and Variation. *arXiv preprint arXiv:1710.10196*.

¹²³ Karras, T., Laine, S., & Aila, T. (2019). A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 4401-4410).

¹²⁴ Brock, A. (2018). Large Scale GAN Training for High Fidelity Natural Image Synthesis. *arXiv preprint arXiv:1809.11096*.

¹²⁵ Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33, 6840-6851.

¹²⁶ Nichol, A. Q., & Dhariwal, P. (2021, July). Improved denoising diffusion probabilistic models. In *International conference on machine learning* (pp. 8162-8171). PMLR.

¹²⁷ Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 10684-10695).

¹²⁸ Dhariwal, P., & Nichol, A. (2021). Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34, 8780-8794.

¹²⁹ Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., ... & Chen, M. (2021). Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*.

¹³⁰ Wang, S. Y., Wang, O., Zhang, R., Owens, A., & Efros, A. A. (2020). CNN-generated images are surprisingly easy to spot... for now. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 8695-8704).

¹³¹ Corvi, R., Cozzolino, D., Zingarini, G., Poggi, G., Nagano, K., & Verdoliva, L. (2023, June). On the detection of synthetic images generated by diffusion models. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 1-5). IEEE.

¹³² Frank, J., Eisenhofer, T., Schönherr, L., Fischer, A., Kolossa, D., & Holz, T. (2020, November). Leveraging frequency analysis for deep fake image recognition. In *International conference on machine learning* (pp. 3247-3258). PMLR.

¹³³ Liu, B., Yang, F., Bi, X., Xiao, B., Li, W., & Gao, X. (2022, October). Detecting generated images by real images. In *European Conference on Computer Vision* (pp. 95-110). Cham: Springer Nature Switzerland.

¹³⁴ Liu, Z., Qi, X., & Torr, P. H. (2020). Global texture enhancement for fake face detection in the wild. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 8060-8069).

¹³⁵ Wang, Z., Bao, J., Zhou, W., Wang, W., Hu, H., Chen, H., & Li, H. (2023). Dire for diffusion-generated image detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 22445-22455).

reconstruction errors as a unique fingerprint. Additionally, a recent approach (Ojha et al.)¹³⁶ explored the use of detectors based on frozen CLIP image encoders, which demonstrate superior generalization across unseen model families compared to ResNet-based detectors.

4.2 Benchmarking framework

4.2.1 Evaluation data

For evaluation in SIDBench, we considered a variety of generative models. First, we incorporated the generative models used in Wang et al. The training set of this dataset consists of ProGAN images and has been used for the training of all GAN-based detectors in our benchmark. The testing dataset includes images generated with different generative algorithms such as ProGAN, StyleGAN, BigGAN, CycleGAN¹³⁷, StarGAN, GauGAN, CRN¹³⁸, IMLE¹³⁹, SAN¹⁴⁰, SITD¹⁴¹, and DeepFakes¹⁴². These image subsets have the same topic distributions to ensure that any performance variation on SID is not due to semantic differences of the images. In addition, we used the dataset from Ojha et al.¹⁴³ that includes recent text-to-image generation models for the generation of 256×256 resolution images. The Latent diffusion model (LDM) and Glide are DM variants, and DALL·E is an auto-regressive model. Following the setting in Ojha et al., three variants of LDM were used for evaluation: (i) LDM with 200 denoising steps, (ii) LDM with 200 steps with classifier-free diffusion guidance (CFG), and (iii) LDM with 100 steps. Similarly, three variants of the Glide model have been used based on the number of steps in the two stages of noise refinement: (i) 100 steps in the first stage followed by 27 steps in the second stage (100-27), (ii) 50-27, and (iii) 100-10. Images of the LAION dataset¹⁴⁴ are used for the real class. Table 23 summarizes these two sources.

To expand the evaluation to more challenging scenarios, we further included the Synthbuster dataset¹⁴⁵ that contains synthetic images generated from 9 different models (1000 per model): DALL·E2, DALL·E3, Adobe Firefly, Midjourney v5, Stable Diffusion 1.3, Stable Diffusion 1.4, Stable Diffusion 2, Stable Diffusion XL and Glide. The original images are based on the raise-1k images dataset¹⁴⁶. For each real image, a description was generated and each of these prompts was manually edited to produce results as photo-realistic as possible.

¹³⁶ Ojha, U., Li, Y., & Lee, Y. J. (2023). Towards universal fake image detectors that generalize across generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 24480-24489).

¹³⁷ Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision* (pp. 2223-2232).

¹³⁸ Chen, Q., & Koltun, V. (2017). Photographic image synthesis with cascaded refinement networks. In *Proceedings of the IEEE international conference on computer vision* (pp. 1511-1520).

¹³⁹ Li, K., Zhang, T., & Malik, J. (2019). Diverse image synthesis from semantic layouts via conditional imle. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 4220-4229).

¹⁴⁰ Dai, T., Cai, J., Zhang, Y., Xia, S. T., & Zhang, L. (2019). Second-order attention network for single image super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 11065-11074).

¹⁴¹ Chen, C., Chen, Q., Xu, J., & Koltun, V. (2018). Learning to see in the dark. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3291-3300).

¹⁴² Rossler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J., & Nießner, M. (2019). Faceforensics++: Learning to detect manipulated facial images. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 1-11).

¹⁴³ Ojha, U., Li, Y., & Lee, Y. J. (2023). Towards universal fake image detectors that generalize across generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 24480-24489).

¹⁴⁴ Schuhmann, C., Vencu, R., Beaumont, R., Kaczmarczyk, R., Mullis, C., Katta, A., ... & Komatsuzaki, A. (2021). Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*.

¹⁴⁵ Bammey, Q. (2023). Synthbuster: Towards detection of diffusion model generated images. *IEEE Open Journal of Signal Processing*.

¹⁴⁶ Dang-Nguyen, D. T., Pasquini, C., Conotter, V., & Boato, G. (2015, March). Raise: A raw images dataset for digital image forensics. In *Proceedings of the 6th ACM multimedia systems conference* (pp. 219-224).

Table 23: Test datasets from Wang et al. and Ojha et al., used to evaluate state of the art detectors in this section.

Family	Method	Source	#Images
Unconditional GAN	ProGAN	LSUN	8.0k
	StyleGAN	LSUN	12.0k
	StyleGAN2	LSUN	16.0k
	BigGAN	ImageNet	4.0k
Conditional GAN	CycleGAN	-	2.6k
	StarGAN	CelebA	4.0k
	GauGAN	COCO	10.0k
Perceptual loss	CRN	GTA	12.8k
	IMLE	GTA	12.8k
Low-level vision	SITD	Raw camera	0.36k
	IMLE	Standard SR	0.44k
Deepfake	FaceForensics++	Videos of faces	5.4k
Text-to-image Diffusion	Latent Diffusion	LAION	3.0k
	GLide	LAION	3.0k
Guided Diffusion	Guided [9]	ImageNet	1.0k
Auto-regressive	Dalle	LAION	1.0k

4.2.2 Integrated models

To map the current landscape of SID methods, we conducted a systematic literature review and identified 11 recent state-of-the-art SID models since 2020 that have been integrated in SIDBench. Our compilation considered two main aspects: a) the type of input used for detection, and b) the architecture of the proposed model. Regarding input, we can categorize models into those applied directly to raw images and those that extract fingerprints from the images, ranging from patterns in the frequency domain to texture patterns, noise patterns, and image gradients from pre-trained deep networks, among others. When it comes to the backbone architecture, many pre-trained networks have been used, with ResNet50 pre-trained on ImageNet classification task, and ViT L/14, pretrained on image-text pairs with a contrastive objective (CLIP)¹⁴⁷, being the most popular. Therefore, we included models based on these two architectures in our framework. Finally, another factor is whether the model is applied to the entire image or to fingerprints extracted by the entire image or local regions (patches). We included examples of both approaches. Table 24 presents a categorization of the integrated models according to these criteria.

Table 24: Categorizing the integrated models based on the type of input features, the backbone architecture used for the detector and the pretraining task of the backbone. Figure sourced from the original paper introducing SIDBench framework.

		Input features	
		Raw Images	Fingerprints
Backbone Architecture + Pretraining Task	ResNet + ImageNet	CNNDetect, DIMD (ResNet18)	LGrad, GramNet Fusing, NPR, FreqDetect
	ViT + CLIP	UnivFD, RINE, DeFake	-
	Other	-	PatchCraft

¹⁴⁷ Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., ... & Sutskever, I. (2021, July). Learning transferable visual models from natural language supervision. In *International conference on machine learning* (pp. 8748-8763). PMLR.

The current list of integrated models is the following:

1. **CNNDetect (Wang et al.)¹⁴⁸**: A standard *ResNet50* architecture pretrained on ImageNet and fine-tuned to detect ProGAN generated images with appropriate selection of data augmentations to improve generalization.
2. **UnivFD (Ojha et al.)¹⁴⁹**: A linear classification layer on top of the last layer features extracted from CLIP's ViT-L/14 image encoder. It is trained on images generated by ProGAN.
3. **LGrad¹⁵⁰**: This uses image gradients as features, computed using a pretrained deep network. These are then processed by a standard ResNet50 pre-trained on ImageNet, fine-tuned on ProGAN images.
4. **DIMD¹⁵¹**: Similar to CNNDetect, but with a modified ResNet50 that avoids down-sampling layers to preserve high frequency fingerprints generated by GANs. It is trained on images generated by ProGAN, Latent Diffusion, and StyleGAN2, with several augmentations to improve generalization.
5. **FreqDetect¹⁵²**: A standard ResNet50 architecture that employs frequency representations as fingerprints for detecting GAN-generated deep fake images, trained on ProGAN.
6. **Fusing¹⁵³**: An attention-based approach that fuses global and local ResNet50 embeddings from the whole image and informative patches. It is trained on ProGAN images.
7. **GramNet¹⁵⁴**: A *ResNet18* architecture that leverages global image texture representations based on Gram matrices. It is trained on StyleGAN images.
8. **NPR¹⁵⁵**: A ResNet50-based architecture that uses neighboring pixel relationships to capture the structural artifacts stemming from up-sampling operations of GANs.
9. **RINE¹⁵⁶**: A CLIP-based classifier that leverages image representations extracted by intermediate blocks of CLIP's ViT-L/14 encoder. It is trained on ProGAN and Latent Diffusion images.
10. **PatchCraft¹⁵⁷**: A universal fingerprint is extracted, based on the inter-pixel correlation contrast between regions with high and low texture diversity within an image.
11. **DeFake¹⁵⁸**: A hybrid approach that utilizes CLIP's image and text encoders, fine-tuned on SID with latent diffusion images as training data.

4.2.3 Evaluation metrics

We evaluate integrated models with established evaluation metrics on each test dataset with respect to the True Positives (TP), True Negatives (TN), False Positives (FP) and False Negatives (FN), where Positive corresponds to the fake class. As an overall performance indicator, we report accuracy (ACC). By default, we use a threshold of 0.5 on the outputs of the detection models, but a study of threshold calibration is also included. We further calculate the Precision-Recall curve, based on different threshold levels. In this work we report the Average Precision (AP) as the weighted mean of precision at each of n thresholds.

$$AP = \sum_n (Recall_n - Recall_{n-1}) \cdot Precision_n$$

¹⁴⁸ Wang, S. Y., Wang, O., Zhang, R., Owens, A., & Efros, A. A. (2020). CNN-generated images are surprisingly easy to spot... for now. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 8695-8704).

¹⁴⁹ Ojha, U., Li, Y., & Lee, Y. J. (2023). Towards universal fake image detectors that generalize across generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 24480-24489).

¹⁵⁰ Tan, C., Zhao, Y., Wei, S., Gu, G., & Wei, Y. (2023). Learning on gradients: Generalized artifacts representation for gan-generated images detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 12105-12114).

¹⁵¹ Corvi, R., Cozzolino, D., Zingarini, G., Poggi, G., Nagano, K., & Verdoliva, L. (2023, June). On the detection of synthetic images generated by diffusion models. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 1-5). IEEE.

¹⁵² Frank, J., Eisenhofer, T., Schönherr, L., Fischer, A., Kolossa, D., & Holz, T. (2020, November). Leveraging frequency analysis for deep fake image recognition. In *International conference on machine learning* (pp. 3247-3258). PMLR.

¹⁵³ Ju, Y., Jia, S., Ke, L., Xue, H., Nagano, K., & Lyu, S. (2022, October). Fusing global and local features for generalized ai-synthesized image detection. In *2022 IEEE International Conference on Image Processing (ICIP)* (pp. 3465-3469). IEEE.

¹⁵⁴ Liu, Z., Qi, X., & Torr, P. H. (2020). Global texture enhancement for fake face detection in the wild. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 8060-8069).

¹⁵⁵ Tan, C., Zhao, Y., Wei, S., Gu, G., Liu, P., & Wei, Y. (2024). Rethinking the up-sampling operations in cnn-based generative network for generalizable deepfake detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 28130-28139).

¹⁵⁶ Koutlis, C., & Papadopoulos, S. (2024). Leveraging Representations from Intermediate Encoder-blocks for Synthetic Image Detection. *arXiv preprint arXiv:2402.19091*.

¹⁵⁷ Zhong, N., Xu, Y., Li, S., Qian, Z., & Zhang, X. (2023). PatchCraft: Exploring Texture Patch for Efficient AI-generated Image Detection.

¹⁵⁸ Sha, Z., Li, Z., Yu, N., & Zhang, Y. (2023, November). De-fake: Detection and attribution of fake images generated by text-to-image generation models. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security* (pp. 3418-3432).

We also calculate the ROC curve and the corresponding AUC-ROC metric, but we do not report it here, as we consider AP to better reflect the real-world performance of SID models. We also calculate and report the True Positive Rate (TPR) and True Negative Rate (TNR), to understand how well fake and real images are detected respectively.

4.3 Evaluation and benchmarking results

4.3.1 Overall performance on GAN and diffusion images

Table 25 and Table 26 present the performance scores (ACC & AP respectively) of the integrated models across the 20 datasets from Wang et al.¹⁵⁹ and Ojha et al.¹⁶⁰ We utilized models trained on ProGAN data from Wang et al., confirming that, as with previous works, almost all models generalize well across different GANs. For instance, UnivFD achieves an accuracy of 90% and higher for most GANs, except for StyleGAN and StyleGAN2, where the performance is still decent. Similarly, DIMD attained an accuracy higher than 90% across other GANs. We observe that fingerprint-based methods like LGrad, FreqDetect, Fusing, GramNet, and PatchCraft also generalize well, albeit with variations across datasets. This demonstrates that fingerprints can capture artifacts common across all GAN models. Conversely, performance on DMs — including Guided Diffusion, Latent Diffusion, and Glide — decreases for most of the models, though exceptions such as GramNet, RINE, and PatchCraft perform surprisingly well. For example, GramNet achieved an accuracy higher than 90% across all datasets. At the same time, ResNet-based models applied on raw images such as CNNDetect and DIMD perform poorly, with accuracy slightly higher than 50%. It is noteworthy that, on average across all datasets, RINE is the best performing model, exhibiting also lower variation. Conversely, LGrad and GramNet, which achieved the second and third best performances respectively, are negatively impacted by specific datasets in which they perform poorly (e.g., CRN and IMLE, SITD and SAN).

Table 25: Accuracy (ACC) scores of the integrated models across 20 test datasets from Wang et al. and Ojha et al.

method	Generative Adversarial Networks								Low level vision		Perceptual loss		Latent Diffusion			Glide			Dall-E	AVG	
	Pro GAN	Style GAN	Style GAN2	Big GAN	Cycle GAN	Star GAN	Gau GAN	Deep fake	SITD	SAN	CRN	IMLE	Guided	200 steps	200 CFG	100 steps	100 27	50 27			100 10
CNNDetect (prob 0.5) [38]	100.0	73.6	68.0	59.3	80.8	80.9	79.6	50.9	78.06	50.0	87.95	94.35	52.3	51.1	51.4	51.3	53.3	55.6	54.2	52.5	66.26
CNNDetect (prob 0.1) [38]	78.70	86.90	84.60	52.30	85.65	92.15	78.70	53.55	90.28	50.46	85.95	85.80	62.00	53.85	55.20	55.10	60.30	62.70	61.00	56.05	71.54
LGrad [37]	99.85	89.0	85.61	84.25	87.3	99.4	83.4	52.35	78.89	79.68	53.55	53.55	76.1	88.7	90.6	89.7	87.3	89.65	89.35	89.0	82.34
DIMD [6]	100.0	99.1	90.8	96.8	91.35	99.35	94.0	67.15	96.11	56.62	99.35	99.35	53.85	57.55	59.4	57.6	56.65	58.7	58.85	69.65	78.11
FreqDetect [10]	99.5	90.8	72.3	82.2	79.05	94.4	81.65	63.85	66.39	51.14	59.95	60.05	57.65	78.95	76.65	79.25	52.1	53.3	49.65	81.5	71.52
Fusing [14]	99.9	82.35	80.8	75.7	83.4	91.65	73.95	54.5	82.78	52.51	87.65	89.5	62.7	53.15	54.25	53.6	60.0	63.1	60.8	53.4	70.78
GramNet [21]	100.0	82.9	85.65	67.45	74.05	100.0	57.55	62.55	72.22	81.51	50.05	50.05	79.5	98.45	98.45	98.7	91.75	93.4	95.55	87.75	81.38
NPR [36]	50.0	50.0	49.95	50.0	49.7	50.0	50.0	54.75	83.06	50.0	50.1	50.05	50.25	49.95	49.95	49.95	51.55	52.35	51.45	50.0	52.15
UnivFD [25]	99.85	83.85	75.65	95.05	98.2	96.05	99.45	68.05	62.22	56.62	56.6	68.1	69.65	94.4	74.0	95.0	78.5	79.05	77.9	87.3	80.77
RINE [18]	100.0	88.0	94.05	99.5	99.3	99.75	99.6	80.3	90.56	68.26	90.45	91.45	76.1	98.25	88.2	98.6	88.75	92.55	90.7	95.0	91.47
PatchCraft [43]	100.0	91.85	89.3	95.25	69.05	100.0	71.2	56.15	88.06	88.58	50.05	50.05	80.5	91.0	90.05	90.9	78.9	82.4	85.2	85.5	81.7

Table 26: Average precision (AP) of integrated models across 20 test datasets from Wang et al. and Ojha et al.

method	Generative Adversarial Networks							Low level vision			Perceptual loss		Latent Diffusion			Glide			Dall-E	AVG	
	Pro GAN	Style GAN	Style GAN2	Big GAN	Cycle GAN	Star GAN	Gau GAN	Deepfake	SITD	SAN	CRN	IMLE	Guided	200 steps	200 CFG	100 steps	100 27	50 27			100 10
CNNDetect (prob 0.5) [38]	100.0	98.49	97.47	88.27	97.03	95.75	98.07	67.08	92.76	63.86	98.94	99.59	68.35	65.92	66.74	65.99	72.03	76.52	73.22	66.26	82.62
CNNDetect (prob 0.1) [38]	100.0	99.62	98.83	84.93	93.88	98.18	89.89	89.56	97.23	70.45	98.49	98.68	77.67	71.16	73.01	72.53	80.51	84.62	82.06	71.3	86.63
LGrad [37]	100.0	97.96	98.48	90.9	95.04	100.0	94.84	62.44	89.89	84.84	73.01	82.6	85.27	96.98	97.66	97.23	94.36	96.07	96.59	97.1	91.56
DIMD [6]	100.0	100.0	99.98	99.78	98.36	99.98	99.75	97.14	99.89	87.19	100.0	99.99	74.57	87.5	89.18	87.77	87.4	90.27	89.34	95.73	94.19
FreqDetect [10]	100.0	98.48	87.72	94.05	85.51	99.57	84.31	72.64	78.08	51.34	81.52	69.68	57.11	92.72	90.39	93.11	53.92	55.0	52.22	94.98	79.62
Fusing [14]	100.0	99.24	99.07	88.91	93.48	99.33	88.4	70.24	88.52	77.66	92.4	96.31	80.1	75.32	77.08	74.87	83.16	87.91	85.03	72.36	86.47
GramNet [21]	100.0	94.96	99.42	62.04	75.18	100.0	54.66	95.09	68.46	81.03	50.18	50.18	79.82	99.83	99.81	99.84	99.11	99.32	99.65	98.81	85.37
NPR [36]	49.66	57.14	62.79	47.87	51.08	49.88	62.85	61.6	93.61	80.67	87.46	95.17	66.51	39.5	42.08	39.18	84.58	89.66	86.96	42.57	64.54
UnivFD [25]	100.0	97.2	97.51	99.29	99.77	99.49	99.99	82.08	63.84	78.81	97.24	98.76	87.64	99.32	92.5	99.27	95.28	95.56	94.96	97.47	93.8
RINE [18]	100.0	99.47	99.98	99.99	99.99	100.0	100.0	97.82	97.2	94.96	97.47	99.67	96.42	99.83	98.28	99.87	98.76	99.24	98.85	99.33	98.86
PatchCraft [43]	100.0	98.87	97.61	99.38	84.33	100.0	81.76	75.29	100.0	98.54	59.97	59.89	90.63	98.32	97.62	98.41	89.21	92.48	94.41	94.29	90.55

To understand how models perform across different datasets, we consider not only accuracy, but we also calculate the TPRs (fake images) and TNRs (real images) for each model. Figure 10 depicts these metrics for the top three performing models: RINE, DIMD, and GramNet. We observe RINE's low variation across all datasets. In datasets where it performs poorly, such as in SAN and Guided Diffusion, it is primarily the TPR

¹⁵⁹ Wang, S. Y., Wang, O., Zhang, R., Owens, A., & Efros, A. A. (2020). CNN-generated images are surprisingly easy to spot... for now. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 8695-8704).

¹⁶⁰ Ojha, U., Li, Y., & Lee, Y. J. (2023). Towards universal fake image detectors that generalize across generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 24480-24489).

that is affected, indicating the model's difficulty to detect synthetic images, while the TNR remains high, limiting false detections. Similarly, DIMD, which fails to generalize on diffusion images, exhibits a low TPR, while maintaining a consistently high TNR ($tnr \approx 0$). Conversely, GramNet demonstrates a totally different behavior in the failing cases, with the TNR being low, meaning the model classifies most real images as fakes. In certain datasets, such as CRN and IMLE, the TNR is even zero. GramNet's performance exhibits extremely high variation, from near-perfect detection to complete failure, making it inappropriate for real-world scenarios where the source and generative model of testing images is unknown. Thus, this analysis underscores the need for systematic evaluation to understand each model's performance in real-world scenarios and applications.

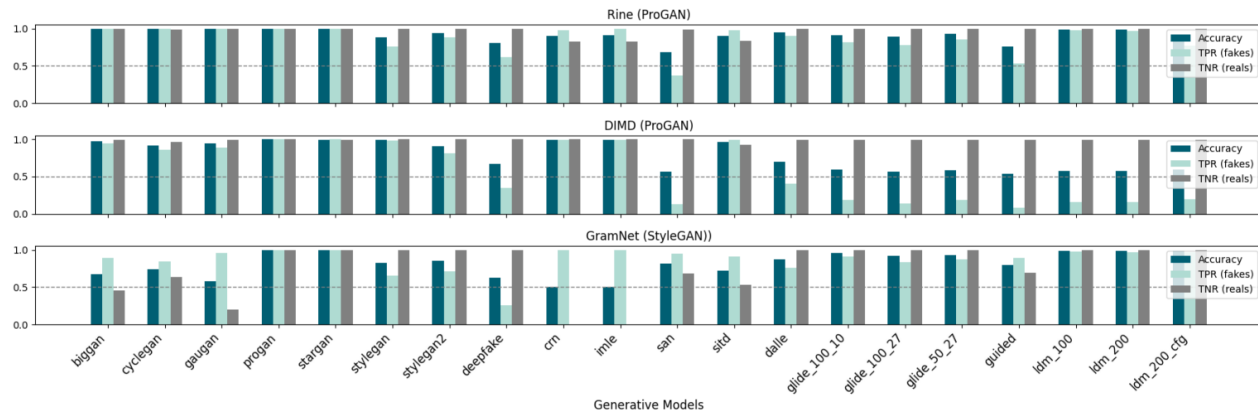


Figure 10: ACC, TPR and TNR across GAN and Latent Diffusion datasets, for the top 3 performing models

From Average Precision (AP) in Table 26, we can conclude that there is potential for further improvements in accuracy through calibration of the threshold used to classify images. The accuracy results in Table 25 are based on the default threshold of 0.5 for all models and datasets. However, as observed in Table 26, AP is higher, in many cases exceeding 90%. To better understand the upper limit of performance, we also calculated the accuracy using the threshold that maximizes accuracy for each model. In Figure 11, we present a comparison between default and oracle thresholds in terms of average accuracy across all datasets from Wang et al. and Ojha et al. It is evident that threshold calibration can lead to some improvements, but this may be a challenging task that requires additional validation data. It should be noted that the optimal threshold may vary across datasets produced by different generative models, making it challenging to find a universal threshold that performs well in real-world applications where the generative algorithm is unknown.

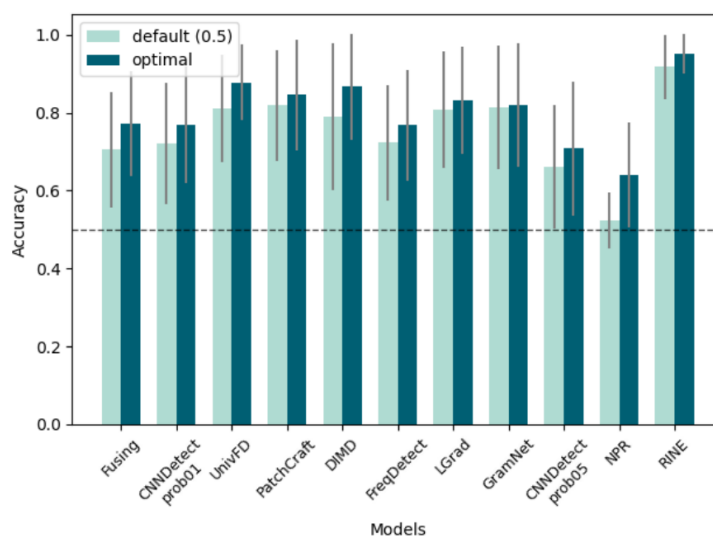


Figure 11: Accuracy with the optimal threshold per model compared to accuracy achieved with the default threshold (0.5). Bars present the average accuracy while error bars indicate the variance of each model across datasets. All models, except GramNet, have been trained on ProGAN images.

4.3.2 Performance on Synthbuster

The datasets from Wang et al. and Ojha et al., although widely used in several works, serving as a reference for research in the field of SID, suffer from low quality and low resolution. Regarding the resolution, images are uniformly sized to 256x256 pixels. Consequently, ProGAN detectors have been trained on relatively small images. To understand the performance in real-world scenarios, we evaluated the ProGAN-trained models using Synthbuster. From Table 27 and Table 28, we observe a significant drop in performance for all models. On average, all models perform almost randomly, with accuracy around 50%. RINE is still the best-performing model, albeit with an average accuracy of 60.26%. It performs poorly on DALL·E 3, Midjourney, and Glide, but achieves decent performance for Adobe Firefly, Stable Diffusion v1.3, and v1.4. Conversely, LGrad and GramNet experience an enormous performance drop. In general, fingerprint-based models (FreqDetect, Fusing) exhibit strong performance loss, indicating that artifacts existing in low-resolution images might not be present in the high-resolution, high-quality images of Synthbuster. We can conclude that DALL·E3 is among the most challenging datasets, as none of the methods is able to detect synthetic images generated by it, in contrast to DALL·E2, which can be detected by the UnivFD detector. MidJourney v5 also presents a significant challenge, as none of the ProGAN-trained detectors can identify these images. For Stable Diffusion v1.3 (SDv1.3) and v1.4 (SDv1.4), UnivFD achieves decent performance (with ACC of 72.05% and 71.6%, respectively), while for SDv2 and SD-XL, the PatchCraft detector achieves similar performance. Adobe Firefly proves to be less challenging for RINE and UnivFD, both of which can achieve ACC values near 80% (79.35% and 78.15% respectively).

Table 27: Accuracy (ACC) scores of integrated models for the Synthbuster dataset.

Method	DALL·E			Stable Diffusion				Mid journey v5	Adobe Firefly	AVG
	Glide	v2	v3	SDv1.3	SDv1.4	SDv2	SD-XL			
CNNDetect (prob 0.5) [38]	48.15	50.5	47.4	47.75	47.8	49.35	49.4	47.9	53.7	49.11
CNNDetect (prob 0.1) [38]	49.75	55.55	45.15	49.8	49.9	49.75	54.2	48.35	53.0	50.61
LGrad [37]	56.55	38.45	24.85	65.65	66.6	37.35	40.85	37.4	31.45	44.35
DIMD [6]	49.25	50.8	45.8	49.4	50.05	50.0	47.9	47.1	61.7	50.22
FreqDetect [10]	34.3	30.2	34.3	67.85	68.0	33.7	49.3	31.4	64.6	45.96
Fusing [14]	52.7	54.2	46.75	50.75	50.85	49.9	49.6	49.3	52.5	50.73
GramNet [21]	50.15	57.8	14.4	56.15	56.55	35.95	49.65	35.65	12.25	40.95
NPR [36]	45.4	45.45	45.2	45.2	45.2	45.25	45.3	46.15	49.65	45.87
UnivFDn [25]	40.7	70.95	34.8	57.85	57.35	63.1	55.45	39.9	78.15	55.36
RINE [18]	54.65	70.5	33.5	72.05	71.6	57.95	59.9	42.85	79.35	60.26
PatchCraft [43]	52.65	53.05	24.1	67.1	67.35	68.25	71.35	53.65	55.2	56.97

Table 28: Average Precision (AP) of integrated models for the Synthbuster dataset.

Method	Dall-E			Stable Diffusion				Mid journey v5	Adobe Firefly	AVG
	Glide	v2	v3	SDv1.3	SDv1.4	SDv2	SD-XL			
CNNDetect (prob 0.5)[38]	38.32	51.16	32.74	38.75	39.26	46.95	46.16	36.45	60.77	43.39
CNNDetect (prob 0.1)[38]	48.02	60.72	32.92	47.67	49.14	47.59	56.83	42.26	57.65	49.2
LGrad [37]	69.23	41.51	31.89	63.29	65.36	39.11	41.67	39.49	34.71	47.36
DIMD [6]	49.42	60.13	31.42	52.03	52.98	49.49	43.24	40.35	76.61	50.63
FreqDetect [10]	36.79	36.36	37.83	82.52	83.0	36.88	44.13	35.45	52.3	49.47
Fusing [14]	52.8	58.99	32.73	49.02	49.32	49.26	45.7	43.73	59.0	48.95
GramNet [21]	50.28	57.08	31.62	55.98	56.5	40.45	48.72	39.66	31.32	45.73
NPR [36]	40.45	39.95	32.96	31.68	31.68	33.19	37.51	35.72	48.36	36.83
UnivFDn [25]	56.6	79.85	30.73	83.08	82.92	65.44	68.31	40.86	92.91	66.75
RINE [18]	54.65	70.5	33.5	72.05	71.6	57.95	59.9	42.85	79.35	60.26
PatchCraft [43]	48.82	45.79	32.8	79.21	81.35	79.33	64.27	51.06	44.27	58.54

To understand the effect of the training dataset, we included in our experiments three models trained on datasets other than ProGAN. Specifically, we used DeFake, trained on Latent Diffusion images, an instance of RINE trained on Latent Diffusion images from Ojha et al., and two variants of DIMD trained on StyleGAN2 and Latent Diffusion images, respectively. In Table 29, we present ACC & AP values on all datasets (GAN-based from Wang et al., diffusion from Ojha et al., and Synthbuster). For DeFake, we were not able to replicate the results reported in the paper. In most cases, performance is almost random with ACC and AP values around 50%. However, in Synthbuster, known for its challenging nature, DeFake appears to outperform other detectors on diffusion datasets. Particularly in DALL·E3, DeFake achieves an accuracy of 78.5%, the highest among all detection methods. Moreover, an AP of 92.4% suggests that further calibration of the classification threshold could yield even better results. Such high performance can be attributed to its focus on text-to-image DMs, and to its hybrid nature, incorporating both visual information and text captions.

The high-resolution images in Synthbuster likely provide more informative captions than the lower-resolution ones. RINE, trained on Latent Diffusion, shows improvements in detecting DM images, while maintaining its good performance on GAN-based images. For example, ACC in GANs is 87.2% on average, while in Ojha et al. diffusion datasets, it is 92.86%. In Synthbuster, RINE achieved a very high 82.84% on average, which ranks among the best scores in this challenging dataset. DIMD trained on StyleGAN2 performs better in GAN images compared to the original version trained on ProGAN. However, it fails to generalize to DM images. On the contrary, DIMD trained on Latent Diffusion exhibits better performance on images generated by other LDMs but not in Glide and Guided and shows a large performance drop in GANs. It is noteworthy that models trained on DM images achieve improved results on some challenging subsets of Synthbuster. For instance, in Midjourney v5, which shows to be challenging according to Table 27, both RINE and DIMD can achieve high accuracy when trained on LD images, with DIMD exhibiting almost perfect detection (accuracy of 98.45%). This suggests that Midjourney incorporates a diffusion component with unique characteristics that are not detected by GAN detectors but can be identified when LD is included in the training set. Similarly, all Stable Diffusion images can be detected with high accuracy by both models.

Table 29: ACC and AP for detection models trained on Latent Diffusion (LD) and StyleGAN2 training data.

Generative Model	DeFake [33]		RINE [18]		DIMD [6]			
	LD		LD		StyleGAN2		LD	
ProGAN	34.6	38.4	99.6	100.0	99.85	100.0	50.8	93.03
StyleGAN	39.5	40.02	86.85	98.49	100.0	100.0	60.75	74.29
StyleGAN2	45.1	46.31	79.55	85.21	100.0	100.0	51.5	73.11
BigGAN	45.15	45.07	88.35	99.76	96.75	99.87	52.25	77.14
CycleGAN	49.3	49.36	94.4	99.89	92.45	99.16	46.9	48.95
StarGAN	35.5	37.55	63.6	99.97	100.0	100.0	43.45	45.44
GauGAN	42.0	37.1	98.05	99.99	89.05	99.66	50.8	89.68
GANs Avg.	41.59	41.97	87.2	97.62	96.87	99.81	50.92	71.66
Deepfake	49.25	49.56	80.5	93.17	54.65	97.3	70.05	87.96
SITD	70.56	88.42	61.11	96.94	97.22	100.0	83.06	95.54
SAN	66.67	80.74	67.58	83.09	50.91	66.09	81.51	98.29
CRN	52.85	54.83	82.5	89.3	93.5	99.9	50.05	96.67
IMLE	40.5	38.27	92.85	99.83	96.5	99.94	49.95	97.24
Guided	47.1	47.18	93.2	97.96	58.05	89.21	51.65	71.39
LD (200 steps)	41.4	42.88	95.9	99.99	68.4	95.39	99.3	100.0
LD (200 cfg)	58.5	63.04	95.9	99.99	65.3	92.67	99.3	100.0
LD (100 steps)	41.2	42.04	95.9	100.0	70.2	96.18	99.3	100.0
Glide (100, 27)	58.55	63.69	87.3	95.29	53.9	74.62	57.8	83.29
Glide (50, 27)	57.55	63.29	89.55	96.25	54.6	79.69	59.55	86.19
Glide (100, 100)	59.95	65.27	89.25	96.22	54.5	75.08	62.6	89.6
Dall-E	47.0	47.19	95.9	99.99	62.95	91.94	88.55	98.27
Diffusion Avg.	51.41	54.32	92.86	98.21	60.99	86.85	77.26	91.09
Glide	73.9	85.66	87.65	93.77	48.7	47.19	52.45	62.67
Dall-E2	41.6	41.59	81.2	90.2	48.25	52.66	49.6	41.7
Dall-E3	78.5	92.4	46.8	32.66	47.3	33.08	49.4	33.07
SD v1.3	56.55	59.95	95.7	99.87	72.45	88.14	99.4	100.0
SD v1.4	55.95	59.28	95.6	99.82	73.05	88.77	99.4	100.0
SD v2	57.9	63.1	88.75	95.58	65.25	81.58	97.95	99.68
SDXL	49.55	51.57	94.6	99.22	51.65	59.42	96.6	99.67
Midjourney v5	58.9	63.81	89.2	95.58	51.35	59.2	98.45	99.88
Adobe Firefly	41.05	40.3	66.1	74.91	70.45	88.51	57.6	86.13
Synthbuster Avg.	57.1	61.96	82.84	86.85	58.72	66.51	77.87	80.31
Total Average	51.59	55.1	85.64	93.55	71.97	84.66	69.31	83.75

4.3.3 Exploring the influence of image resolution on model performance

An important characteristic that distinguishes Synthbuster from the datasets of Wang et al. and Ojha et al., and might affect the difference in the performance, is the high resolution of its images. As mentioned above, the datasets from Wang et al. and Ojha et al. consist of images with the same low resolution (256x256 pixels), while Synthbuster includes much larger images, the size of which also depends on the generative model or tool used. For example, Midjourney images are usually around 1300x800 pixels, while Adobe Firefly images exceed 2000x1700 pixels. This difference raises two important issues. First, most SID models have been trained on low-resolution images (e.g., ProGAN 256x256 images), meaning that they have not encountered large images with complex visual content during the training. Second, some image encoders, such as CLIP’s ViT-L/14 used in UnivFD, RINE, and DeFake, require images to be of a specific size (in this case, 224 pixels). Therefore, to handle Synthbuster images with pretrained models, there are two options: a) either crop the images to a predefined size (center crop or random) or b) resize them. In the case of cropping, it is obvious that additional information outside the cropped area is lost, while resizing introduces a distortion that may affect performance. For example, resizing an image affects its frequency components, which could impact models that rely on frequency artifacts, such as FreqDetect.

The experiments conducted so far have utilized cropped versions of the Synthbuster images. To evaluate whether cropping or resizing affects performance and to quantify this impact, we reran the evaluation on the Synthbuster using the resized version. For CLIP-based models, we resized images to 224 pixels, while for all other cases, we resized them to 256 pixels. Figure 12 presents the average accuracy for all models across the 9 Synthbuster datasets, alongside the standard deviation. It is observed that, in most cases, the average

performance remains quite similar. However, the standard deviation indicates that resizing versus cropping affects the performance of each model across the Synthbuster datasets differently. For instance, the average accuracy of UnivFD or ProGAN-trained RINE is similar for both cropped and resized images, but the variation is significantly higher in the case of cropped ones. Conversely, DIMD and RINE, when trained on Latent Diffusion, exhibit much higher performance with cropped images, while their accuracy significantly drops with resized images. Considering that the former is based on ResNet50 pretrained on ImageNet and the latter on ViT-L/14 pretrained on a web scale dataset with CLIP contrastive loss, we can infer that it is not the encoder that is affected by cropping or resizing but the classification layers that follow it. On the other hand, in DeFake, resizing improves the average performance significantly. As DeFake, considers both images and text captions generated by the visual part resizing largely retains the available information, in contrast to cropping.

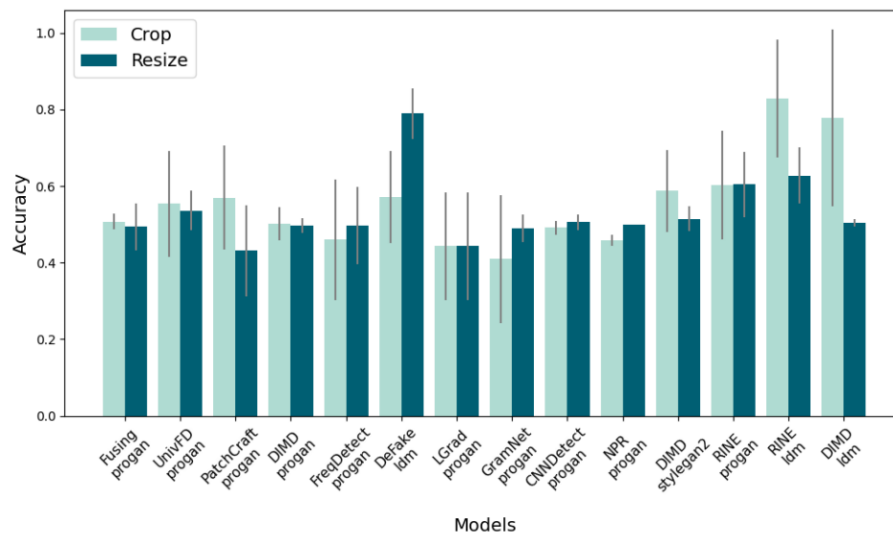


Figure 12: Comparing accuracy across cropped and resized images in the Synthbuster dataset.

4.3.4 Influence of image transformations

It is known from previous works on SID, that image transformations on testing images, such as the addition of Gaussian blurring, JPEG re-compression, resizing, etc., negatively affect the performance of detection models. To counteract this and improve generalization, a common practice is to perform strong data augmentation during training, incorporating such transformations in the training data. In this work, we tested the performance of the integrated models under two types of transformations: Gaussian blurring with varying degrees of σ (2 and 4) and JPEG re-compression with varying image quality (95, 90, 50 and 30). Figure 13 presents the ACC, TPR and TNR, when different transformations are applied. We can observe in the first subplot that the inclusion of any transformation results in worse performance in terms of ACC compared to the original images with a varying degree of impact. For some methods, such as RINE, JPEG re-compression has less impact compared to Gaussian blurring, while for others, such as DIMD (trained on ProGAN or StyleGAN), Gaussian blur has a negligible effect.

However, accuracy alone cannot provide the full picture. For example, with GramNet, we observe that all transformations have the same degree of negative impact. In contrast, for models like CNNDetect trained on ProGAN, blurring marginally improves accuracy, as it does for DIMD trained on latent diffusion. However, the TPR and TNR paint a different picture. We can observe that transformations significantly affect these metrics, effectively neutralizing each other, resulting to moderate changes in accuracy. For instance, in the CNNDetect example, the slight increase in accuracy can be attributed to the increase in the TPR (i.e., the detection of fakes), with the downside of a decrease in TNR. In other words, blurring results in more images being detected as fakes, including some real ones. This behaviour is similar for most models. TPR increases at the expense of TNR, resulting in the models detecting blurred images as fakes. This is evident in the GramNet and PatchCraft cases, where blurring results in almost all images being classified as fakes. It is intriguing that the observed behaviour disappears when blurring is combined with JPEG re-compression, suggesting that this phenomenon needs more in-depth investigation. With JPEG re-compression, behaviour is more inconsistent

across models, as variations in the TPR and TNR differ. Generally, the TNR is less impacted than the TPR, except for GramNet. For this model, the TNR increases close to 1, while the TPR drops to 0, indicating that JPEG compression leads to GramNet’s complete inability to detect any fake images.

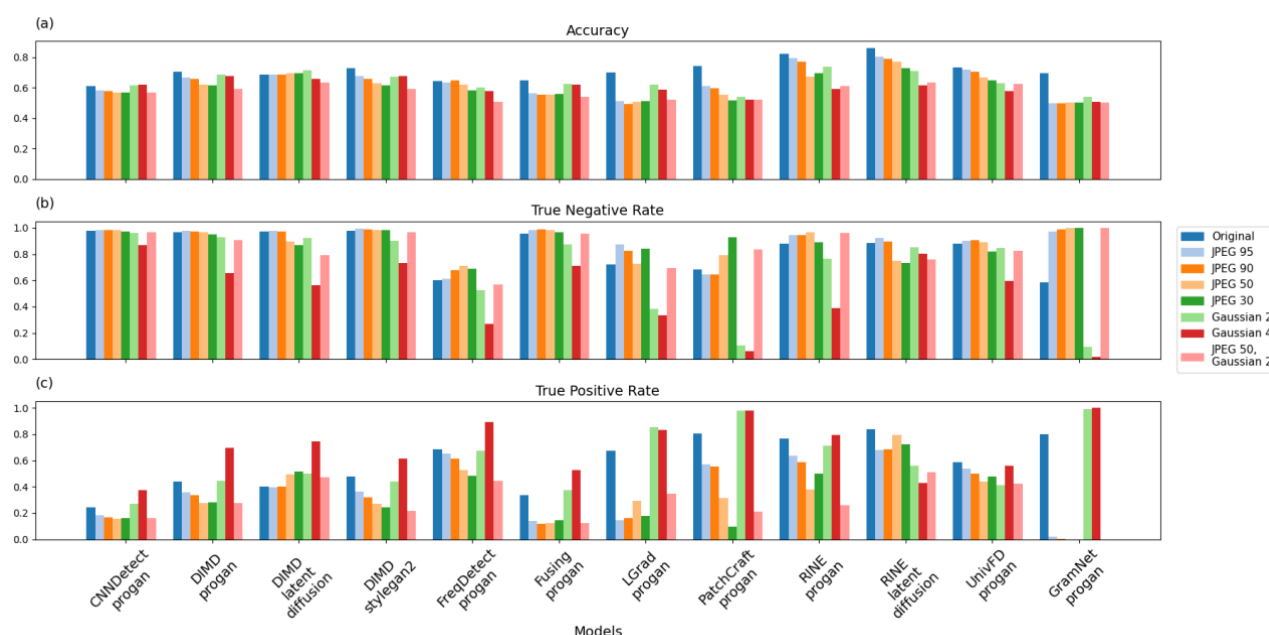


Figure 13: ACC, TPR and TNR across detection models for different image transformations. Metrics have been averaged across.

4.4 Conclusion

Detecting synthetic images remains a significant challenge due to the rapid proliferation of generative models and the diverse range of detection models developed to counter them. No single detection model consistently outperforms others across all datasets, highlighting the complexity and variability of this domain. The absence of a universal model capable of effectively detecting all types of synthetic images underscores the need for systematic evaluations to assess the real-world applicability of detection models. This section introduced the SIDBench framework, designed to provide a structured evaluation of synthetic image detection (SID) models. Recognizing that all detection models are sensitive to image transformations such as blurring, resizing, cropping, and JPEG compression, SIDBench incorporates comprehensive evaluation metrics to enable a deeper understanding of each model’s performance.

Future enhancements to SIDBench will include a broader range of datasets, especially those that mirror real-world scenarios, along with the integration of new detection models. Notably, recent research has introduced approaches that model the distribution of real images and treat SID as an anomaly detection problem. SIDBench will incorporate these novel models and systematically report their performance.

5 Risk Assessment of Image Generative AI

To analyse the risks posed by image generative algorithms in the context of disinformation, we systematically evaluated 70 widely recognized generative AI image models, examining key metrics that reflect their reception and adoption within the research community and the broader public. To compile this list, we began by focusing on two main families of models, GANs and Diffusion models, and conducted a thorough review of the relevant literature, excluding works that had a limited number of citations. Following this review, we identified and examined corresponding GitHub repositories. Models that were not open source, even if highly cited, were excluded from consideration. For the analysis and targeting of generative models, each model is categorized not only by its generative approach—such as GANs and Diffusion Models—but also by its core tasks. Core

tasks refer to the specific functions or applications each model is designed to perform in the realm of image generation and manipulation. For instance, image synthesis is a foundational task, where models generate entirely new images from random noise or latent variables, effectively creating visual content without any direct input images. Super-resolution, another key task, involves enhancing the resolution of low-quality images, improving their details and clarity—an ability that can be leveraged to generate high-quality, deceptive visuals. Text-to-image synthesis, which generates images from textual descriptions, poses increasing risks by enabling the creation of realistic yet entirely fabricated scenes with minimal effort. Through this categorization, we ensured that the analysis includes models spanning a wide range of functionalities that could be exploited for disinformation purposes.

This study aims to assess the potential risks these models present, particularly due to the ease with which they can be used by non-experts to create misleading or harmful content. The increasing accessibility and user-friendliness of these models and especially of frameworks for running these models, make it possible for individuals with little to no technical expertise to generate highly convincing visuals, amplifying their potential misuse in spreading disinformation. By focusing on the ease of use and accessibility of these models, this study highlights the growing risk that generative AI tools can be weaponized by individuals without technical expertise, further lowering the barriers to creating and disseminating deceptive visual content.

5.1 Popularity and adoption of GitHub projects

To objectively assess each model's prominence and adoption rate, we analyzed key community metrics from GitHub repositories, including *stars*, *forks*, *watchers*, and *open issues*. These metrics offer distinct insights into a model's popularity, engagement, and activity level, providing a comprehensive perspective on its real-world impact and reliability. Combining the evaluation of functional characteristics with community metrics, we established a well-rounded framework for assessing not only the technical capabilities of each model but also its practical applicability and collaborative potential. This structured approach enables a deeper understanding of which models currently exhibit the highest utility and influence within the AI research community and beyond.

Stars represent an indicative metric of a model's popularity and perceived value within the developers and research community. When users "star" a repository, they are essentially bookmarking it as noteworthy or interesting. High star counts often indicate that a model has been widely recognized for its quality, utility or innovation, making it appealing to a broad user base. Models with high star counts are generally trusted within the community, as these stars are an informal endorsement from users who have found the model useful or intriguing. Forks reflect the extent to which a model serves as a foundational tool for further development and experimentation. When a user "forks" a repository, they create a personal copy of the code, often with the intention of modifying it, testing new configuration or building on its functionality. A higher fork count can signify that the model is adaptable and versatile, with users actively extending its use cases or customizing it for specific applications. Watchers represent the number of users who are actively following the repository, subscribing to updates and notifications. Watchers receive alerts on new developments, issues or enhancements in the model, indicating a sustained interest in the project's ongoing progress. A high watcher count suggests that the model is considered valuable enough to track closely and that users expect it to evolve or improve. This level of engagement implies that the community sees the project as active and relevant, adding to its credibility and appeal. Open Issues provide a view into the activity level and responsiveness within the model's development. Open issues are created when users or developers identify bugs, suggest enhancements or propose new features, reflecting areas for improvement or troubleshooting. A high count of open issues may indicate some instability or aspects of the model that need refinement, but it also points to an engaged user base testing the model thoroughly. A project with active issue reporting suggests a responsive development team and a community dedicated to improving the model's functionality.

Table 30 presents the top 20 repositories (out of the initially listed 70) in terms of stars. We can conclude that the most popular repositories in terms of stars include well-known models from both the GAN family and latent diffusion methods. These repositories also exhibit high numbers of forks, watchers, and open issues, reflecting significant activity and widespread adoption within the research community. However, while these numbers are indicative of the popularity of generative models, they do not provide any insight into the ease of use, particularly for non-experts.

Table 30: Top 20 most popular image generation models based on star ratings on GitHub.

Model	Stars	Forks	Watchers	Open Issues	Repository
Stable Diffusion	65447	9838	552	578	GitHub Link
ControlNet	27900	2532	210	411	GitHub Link
StyleGAN	13945	3152	453	11	GitHub Link
CycleGAN	12140	1927	389	57	GitHub Link
DALL-E 2	10829	1054	124	68	GitHub Link
StyleGAN2	10792	2527	373	25	GitHub Link
LDM	10594	1395	95	273	GitHub Link
pix2pix	9932	1700	322	90	GitHub Link
Imagen	7783	734	113	100	GitHub Link
DeepFloyd IF	7682	498	84	100	GitHub Link
SPADE	7534	983	282	101	GitHub Link
LaMa	7214	783	79	87	GitHub Link
DDPM (Pytorch)	7017	908	32	132	GitHub Link
pix2pix-HD	6529	1375	169	244	GitHub Link
StyleGAN3	6175	1085	56	176	GitHub Link
U-GAT-IT	6166	1052	178	56	GitHub Link
ProGAN	6048	1110	271	11	GitHub Link
Guided Diffusion	5596	748	140	94	GitHub Link
DALL-E	5494	637	96	125	GitHub Link
SRGAN	5298	669	191	5	GitHub Link

5.2 Code quality of GitHub projects

To further evaluate the quality of these repositories and assess the associated risks of misuse by non-experts, we conducted an in-depth analysis of the code within these repositories. In the analysis of AI image generation models, we utilized the Cyclopt¹⁶¹ platform to conduct automated code evaluations across several crucial software metrics: maintainability, security, readability, and reusability. Cyclopt facilitates an in-depth code review process, yielding objective, quantified insights into the architecture, security, and overall quality of the model's codebase by evaluating its adherence to best practices and recognized coding standards. This approach enables a precise assessment of code structure and quality, ensuring that the codebase supports scalability, robustness, and adaptability to new developments.

Cyclopt's code evaluations focus on specific indicators of quality and complexity that collectively determine how well the code performs in real-world development and deployment contexts. The following metrics were

¹⁶¹

<https://www.cyclopt.com/>

instrumental in establishing a holistic view of each model's software quality:

1. **Maintainability:** This metric assesses how easily the code can be updated or modified over time, which is critical for adapting to new requirements or fixing bugs. Cyclopt analyzes maintainability through indicators like modularity (how well the code is segmented into cohesive, independent modules) and code duplication (the presence of redundant code sections). Code that scores high on maintainability is easier to adapt, less prone to errors, and typically requires less effort to extend or optimize, which is particularly important for AI models that may need frequent updates.
2. **Security:** Cyclopt evaluates potential security risks within the code, including vulnerabilities that could be exploited. It checks for issues such as improper handling of sensitive data, use of outdated or insecure libraries, and poor authentication mechanisms, which are especially pertinent for AI models deployed in production. By identifying and quantifying these risks, the platform provides a roadmap for improving the code's defenses against potential threats, enhancing its resilience and trustworthiness.
3. **Readability:** This metric examines how easy it is for developers to understand and follow the code. Cyclopt assesses readability by looking at factors like consistent naming conventions, logical structure, proper documentation, and clear function definitions. High readability ensures that the code can be easily understood, shared, and extended by other developers or teams, facilitating collaboration and minimizing misunderstandings or errors during development.
4. **Reusability:** Reusability measures the extent to which code components can be repurposed across different projects or applications. Cyclopt evaluates this by examining code modularity, parameterization, and adherence to design patterns that promote reuse. A high reusability score indicates that the code has been designed to support multiple use cases, increasing its value in diverse development contexts and reducing the need to recreate similar components from scratch.

Specifically, the above metrics are appropriately determined through an extensive list of key code quality and complexity indicators.

- **Lack of Cohesion in Class Methods:** It measures the cohesion of your source code. The term cohesion refers to the basic principle that one class should implement only a certain functionality
- **Nesting Level of Class:** This measures the depth of nested blocks. Highly nested blocks increase complexity, reduce code clarity, and make code more difficult to read.
- **Weighted Methods per Class:** It measures the complexity of the class as expressed by the number of independent control flow paths in it.
- **Coupling Between Objects:** It measures the coupling of the class as expressed by the number of directly used other classes. Classes using many other classes are very sensitive to the changes in the system.
- **Number of Incoming Invocations:** It measures the coupling of the class as expressed by the number of other methods and attribute initializations which directly call the local methods of the class.
- **Number of Outgoing Invocations:** It measures the coupling of the class as expressed by the number of other methods and attribute initializations which are directly called by the local methods of the class.
- **Response Set for Class:** It measures the coupling of the class as expressed by the number of local methods in the class given the number of directly invoked other methods by its methods or attribute initializations.
- **Total Comment Density:** It measures the level of documentation of the class as expressed by the ratio of the total comment lines of the class to the total lines of code.
- **Class Size in Lines of Code:** Number of code lines of the class, including empty and comment lines.
- **Number of Methods per Class:** Number of methods in the class, including the inherited ones, as well as the inherited and local methods of its nested, anonymous, and local classes.
- **Depth of Class Inheritance Tree:** Length of the path that leads from the class to its farthest ancestor in the inheritance tree.
- **Maintainability Index:** It measures the extent to which a software component is maintainable based on its complexity, size and level of documentation.
- **Security Vulnerabilities of Dependencies:** It identifies the known vulnerabilities existing in the dependencies of the source code.

Figure 14 presents the overall rating of the StyleGAN3 GitHub repository¹⁶², alongside the four key metrics. In addition, Figure 15 presents some of the low-level key indicators used to calculate the overall rating.

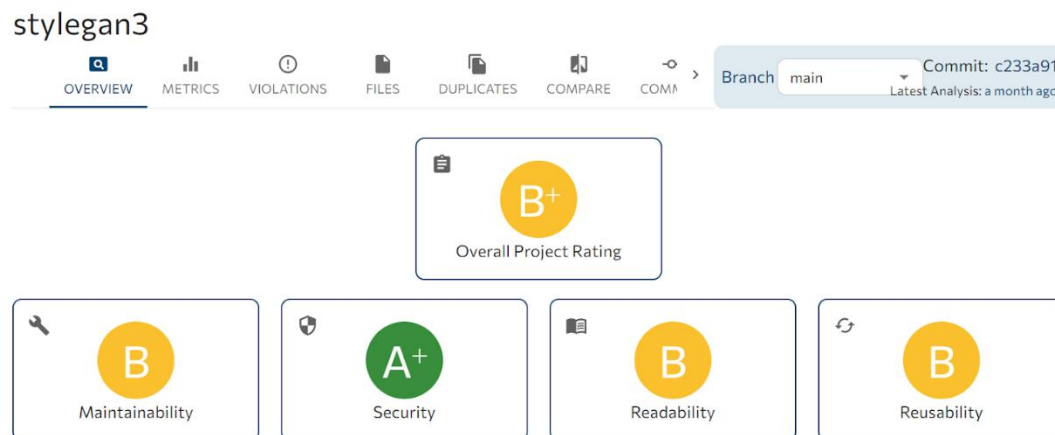


Figure 14: Overall project rating and metrics assessment for the StyleGAN3 repository.

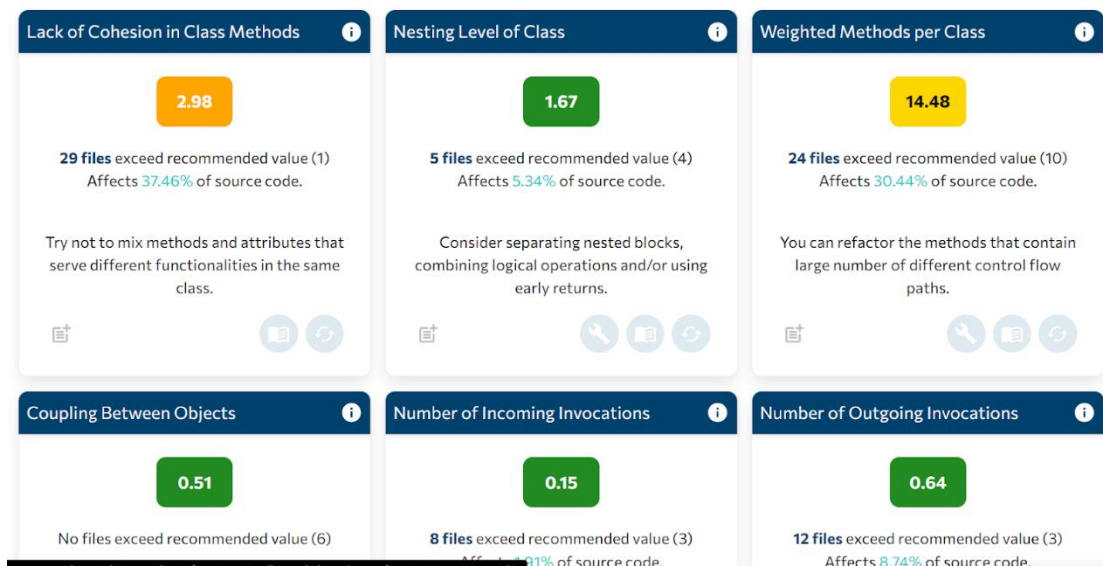


Figure 15: Some code quality key indicators for the StyleGAN3 repository.

Our analysis of the initial 70 repositories indicates that 65 out of 70 models have an overall project rating of B, with maintainability and reusability also scoring at a B level, reflecting moderate ease of updates and adaptability. Notably, no model achieved an A rating in readability and reusability. Similar results were observed for maintainability, where only two models, TextureGAN and CycleGAN-pix2pix, managed to achieve an A rating. Security often achieves a high rating of A+, although some models show vulnerabilities with lower security ratings. However, readability frequently scores lower at C, indicating challenges in clarity and interpretability that may hinder accessibility for developers. These metrics offer an objective measure of each model's code robustness, facilitating informed comparisons and highlighting specific areas for optimization based on key indicators for each model.

Overall, the models demonstrate a reasonable degree of structure and organization, as evidenced by their moderate to high scores across most metrics. With an average project rating of B, they reflect a solid adherence to coding standards, particularly regarding maintainability and reusability, both crucial for long-term usability and adaptability. However, the consistent lower scores in readability suggest that less experienced users may

¹⁶² <https://github.com/NVlabs/stylegan3>

encounter challenges in understanding or adapting the code, potentially limiting accessibility for those who rely on clear documentation and intuitive code structures for customization.

In terms of ease of online deployment, many of these models could be implemented with moderate effort, especially given their generally high security scores. However, readability issues could hinder users from customizing or modifying the models without additional guidance. This suggests that while these models are structurally sound for online applications, the user experience might vary based on each model's readability and documentation quality. Enhancing code clarity and documentation could improve accessibility and adoption, particularly for users aiming to implement these models without extensive programming support. This increased accessibility might come with a potential downside: making these generative models easier to use also increases the risk of malicious usage. Improved documentation and user-friendly interfaces, while beneficial for legitimate users, may lower the barrier for misuse, enabling individuals with malicious intent to exploit these models more effectively. This highlights the need to balance usability with robust safeguards, such as usage restrictions, ethical guidelines, and built-in mechanisms to prevent harmful applications, to mitigate the associated risks effectively.

To note also, that despite the mediocre maintainability and reusability scores, the risk of malicious usage is not significantly mitigated by these factors. Malicious users are typically not concerned with adapting or improving the code; they are more likely to use it as-is for their purposes, such as generating synthetic images. This means that even if the code is less maintainable or adaptable, it still poses a risk of misuse.

5.3 Assessment of image quality

To analyse image generation, we focused on selecting models that excel in producing photorealistic outputs, as realism is a key criterion for evaluating the effectiveness and potential impact of generative models, particularly in the context of disinformation. Photorealism is crucial not only for visual appeal but also for its potential to increase believability, making it a decisive factor in assessing how these models might contribute to the spread of misinformation if misused. Our selection of generative models centred on diffusion models, which represent the current state of the art in generative AI and are widely recognized for their superior performance in producing realistic images. We specifically opted for 11 models out of the initial 70, designed to generate realistic images, excluding those primarily trained for other purposes, such as creating artistic or stylistic content, to ensure the analysis remained focused on models most relevant to disinformation risks. Given their capability to generate high-fidelity outputs, diffusion models present a “worst-case scenario” in terms of potential misuse for disinformation. By focusing on diffusion models, we aimed to analyze the highest-performing and most realistic image generators available, providing a rigorous benchmark for understanding the potential risks associated with synthetic media.

The 11 selected models are all pre-trained and hosted on Hugging Face's Model Hub¹⁶³, allowing for easy access, deployment, and experimentation. As the focus of this section is the risk of using AI from non-expert users, using pre-trained models accelerates the evaluation process, enabling us to focus on assessing image quality, realism, and other performance metrics without requiring extensive retraining. Hugging Face's platform provides comprehensive documentation and community support, making it an ideal choice for streamlined experimentation with cutting-edge generative models. It is important to note that the selected models differ from the most popular models highlighted earlier based on GitHub metrics. This distinction demonstrates that high popularity within academia or the research community, and even high visibility on platforms like GitHub, does not necessarily align with ease of use or superior performance. Instead, the selected models although present a lower popularity score, prioritize state-of-the-art performance and seamless execution through the Diffusers library¹⁶⁴. These characteristics make them particularly suitable for potential disinformation and malicious use cases, as they combine high performance with user-friendly accessibility.

The Diffusers library was integral to deploying these image generation models in Python. Developed specifically for diffusion models, Diffusers provides a highly intuitive interface that allows researchers and developers to load, modify, and generate images using pre-trained models directly from Hugging Face's Model

¹⁶³ <https://huggingface.co/>

¹⁶⁴ <https://huggingface.co/docs/diffusers/index>

Hub. Its design makes it straightforward to work with complex model architectures, allowing for easy adjustments to parameters and configurations, even for advanced generative techniques. Diffusers' streamlined functionality supports both image generation and customization, simplifying the process of experimenting with various aspects of photorealistic output generation. By handling much of the model integration complexity, the library ensures that researchers can focus on experimentation and refinement without being hindered by technical overhead. This ease of use is especially valuable when working with sophisticated diffusion-based models, which often require substantial computational resources and detailed parameter tuning. The ability to rapidly switch between models and adjust settings in Diffusers enhances productivity and enables thorough comparative analysis, critical for identifying the models best suited to high-stakes applications such as disinformation detection and synthetic media research.

To thoroughly evaluate the potential of diffusion models in generating realistic images, we constructed a comprehensive pipeline that includes real-world, high-quality images as input for our experiments. After assessing various datasets, we selected the Visual News dataset¹⁶⁵, a large-scale benchmark containing over one million news images, complete with associated news articles, image captions, author information, and other metadata. This dataset is well-suited for our goals, as it encompasses diverse content related to public events, prominent figures, political scenarios, and other contexts, that closely align with the types of images at risk of misuse in disinformation. Our selection of images from the Visual News dataset focuses on those specifically related to public events, public figures, politics, and social issues—contexts in which generative models, if misused, could have heightened impact on public perception and trust. This approach ensures that our generated outputs are tested on a realistic and relevant set of visual themes, allowing us to assess the effectiveness of these models in producing lifelike, potentially influential synthetic media. From these categories, we randomly selected a subset of 1,600 images, creating a balanced and representative sample.

For the generation process, we utilized either the images directly or their accompanying captions as inputs, depending on the model type. With text-to-image models, the captions provided a basis for creating synthetic images aligned with specific themes and scenarios, while the image generation models used actual images to assess the ability to generate modifications or variations of real-world visuals. By using both images and captions as inputs, we were able to comprehensively evaluate the flexibility and realism of diffusion models across different input types, giving insight into their adaptability and effectiveness in generating contextually coherent outputs. We also included models that accept hybrid inputs—both text and image. For these models, we used the original image from the Visual News dataset as a visual input, while the associated caption or article content served as the text input, prompting the model to replicate or complement the original visual content. This hybrid approach allowed us to test the model's capacity to synthesize images that align closely with both visual cues and textual context, offering a more complex and nuanced output.

Leveraging high-performance computing resources, including an RTX 4090 GPU with 24GB of memory and 24 CPU cores, the image generation process was completed in approximately seven days. This hardware setup, although not standard for everyone, is reasonably priced and accessible to non-experts, highlighting the growing availability of powerful computational tools. It is also worth noting that some models were excluded from the analysis because they required hardware beyond this level of accessibility, such as GPUs with larger memory capacities. By focusing on models that can be executed on hardware of this level, we aimed to ensure that our evaluation reflects scenarios feasible for a wider range of users, including those with malicious intent.

To assess the quality of the generated images, we applied several image similarity metrics, each providing distinct strengths to evaluate realism and visual fidelity across models.

- **Frechet Inception Distance (FID)**¹⁶⁶ was selected for its ability to measure the distribution similarity between generated and real images by comparing feature representations from a deep neural network (Inception). This approach captures both high-level and fine-grained features, making FID particularly effective at detecting image artifacts and quality inconsistencies, providing a reliable score on the overall visual similarity of generated content to real-world data. A lower distance indicates greater similarity between real and synthetic images.

¹⁶⁵ Liu, F., Wang, Y., Wang, T., & Ordonez, V. (2020). Visualnews: Benchmark and challenges in entity-aware image captioning. *arXiv preprint arXiv:2010.03743*, 1.

¹⁶⁶ <https://pytorch.org/torcheval/main/generated/torcheval.metrics.FrechetInceptionDistance.html>

- **Kernel Inception Distance (KID)**¹⁶⁷ complements FID by also assessing similarity in feature space but using a kernel-based method to estimate distribution similarity. KID is beneficial for cases with smaller sample sizes, as it is more stable than FID in such contexts. This makes it a valuable metric in confirming image quality consistency across a more limited selection of generated outputs, adding robustness to the quality assessment. A lower distance indicates greater similarity between real and synthetic images.
- **Learned Perceptual Image Patch Similarity Loss (LPIPS)**¹⁶⁸ focuses on perceptual similarity by comparing image patches through deep features rather than pixel-wise differences, aligning with human perceptual judgments. LPIPS's approach is especially useful for assessing subjective quality aspects such as texture and color consistency, giving insight into how natural and visually appealing the generated images appear. A low LPIPS score means that image patches are perceptual similar¹⁶⁹.
- **Structural Similarity Index (SSIM)**¹⁷⁰ offers an additional layer by measuring image quality based on structural information, capturing details on contrast and luminance changes. SSIM is particularly valuable for understanding the integrity of spatial relationships and structure in the generated images, allowing for an assessment of how well models preserve important content-specific details, such as shapes and edges, making it essential for images where clarity and structural fidelity are critical. A higher value indicates better quality.

In Figure 16, Figure 17, Figure 18 and Figure 19 we present the FID, KID Mean, LPIPS Mean and SSIM Mean respectively for all the eleven generative models. We observe significant variability across the different models, particularly in terms of FID and KID metrics. Notably, latent diffusion (*ldm-text2img*), being among the oldest approaches, consistently delivers the worst performance, while *InstructPix2Pix* achieves the best scores systematically.

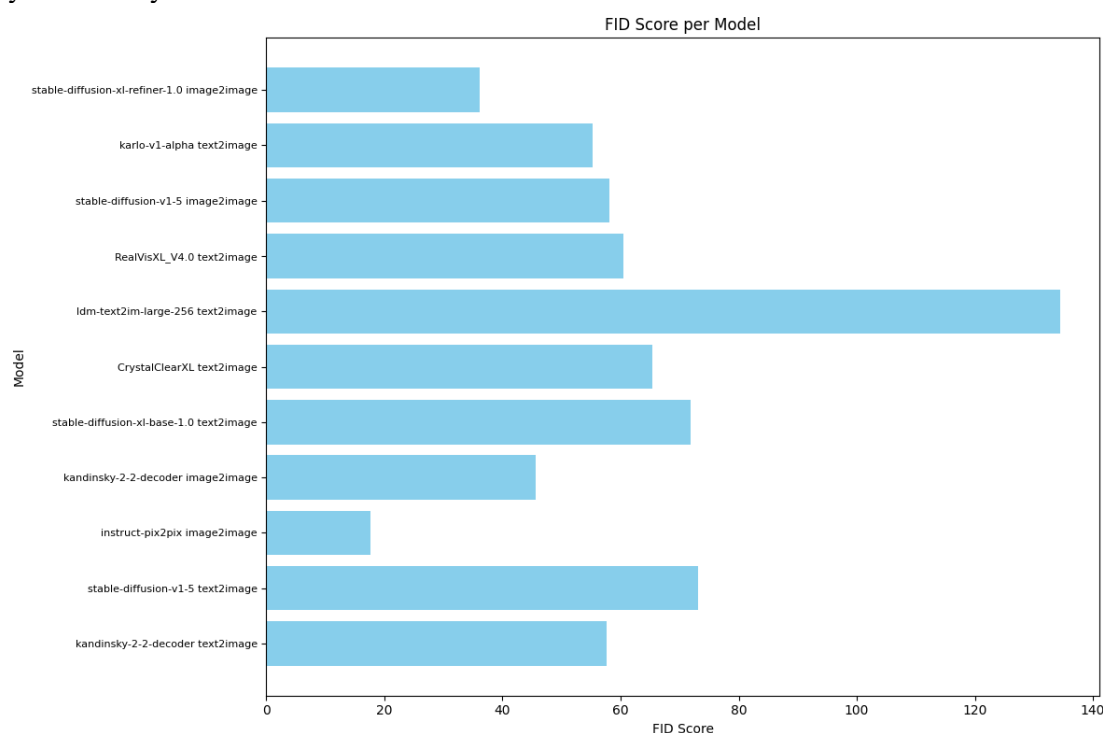


Figure 16: FID score for the 11 generative models used for evaluation. Lower values are considered better.

¹⁶⁷ https://lightning.ai/docs/torchmetrics/stable/image/kernel_inception_distance.html

¹⁶⁸ <https://github.com/richzhang/PerceptualSimilarity>

¹⁶⁹ Despite the term similarity in its name, this metric is a loss, therefore a lower value indicates better matching between the real and the synthetic images.

¹⁷⁰ https://lightning.ai/docs/torchmetrics/stable/image/structural_similarity.html

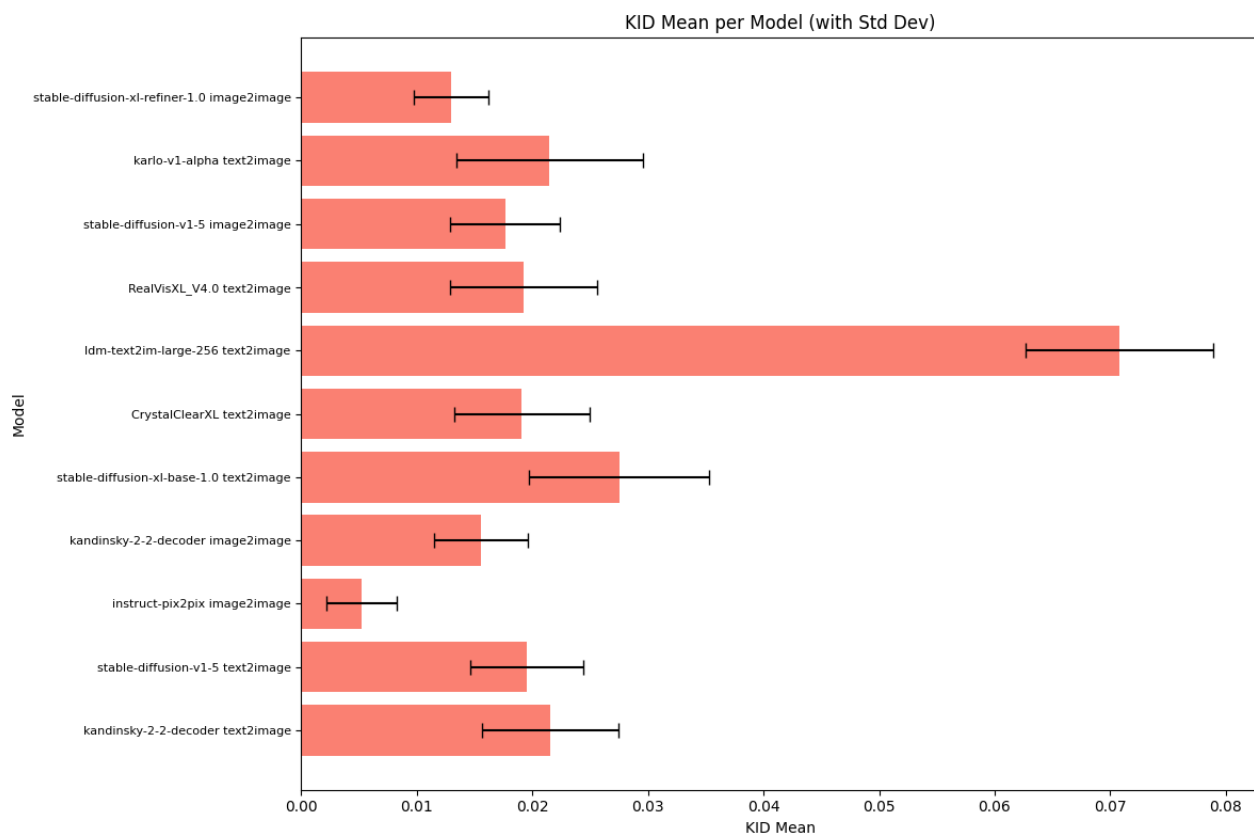


Figure 17: KID score for the 11 generative models used for evaluation. As KID is calculated for each pair of real-synthetic images we report the average KID for all the 1600 pairs, alongside standard deviation. Lower values are considered better.

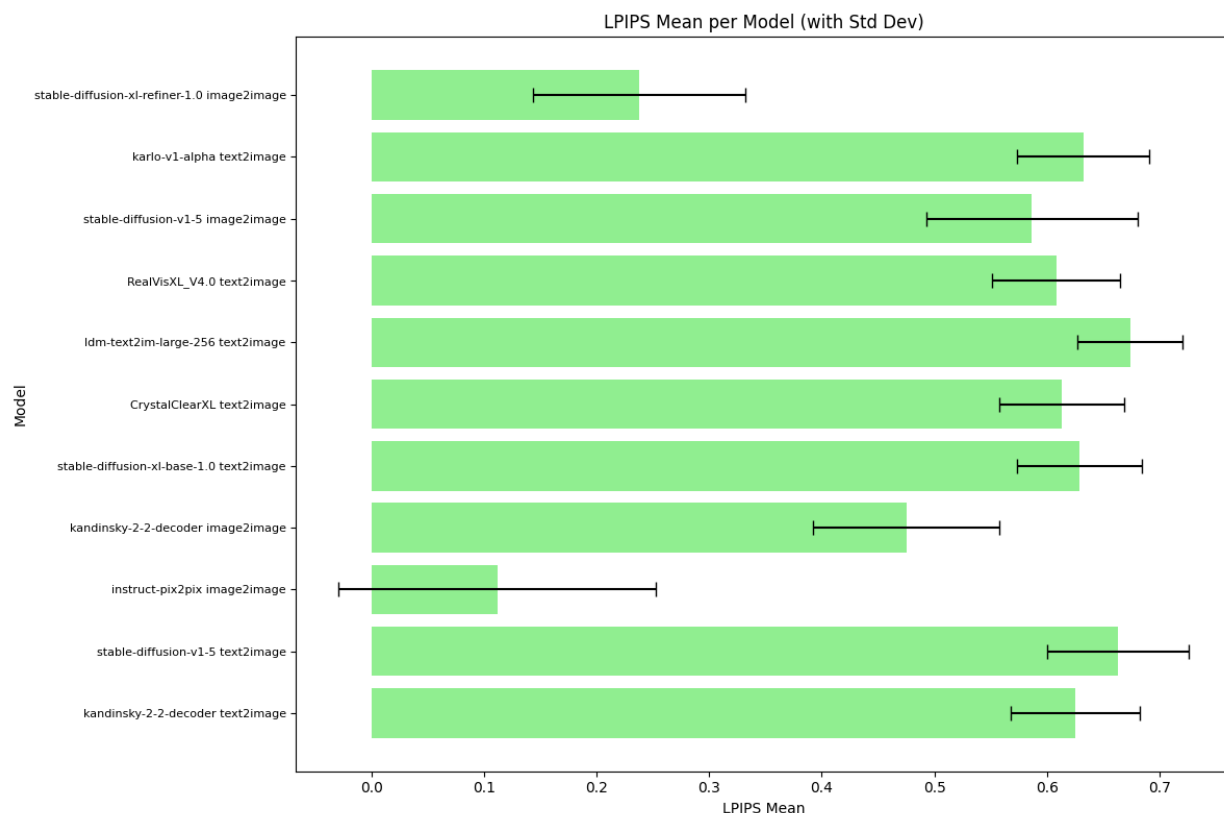


Figure 18: LPIPS score for the 11 generative models used for evaluation. As LPIPS is calculated for each pair of real-synthetic images we report the average LPIPS for all the 1600 pairs, alongside standard deviation. Lower values are

considered better.

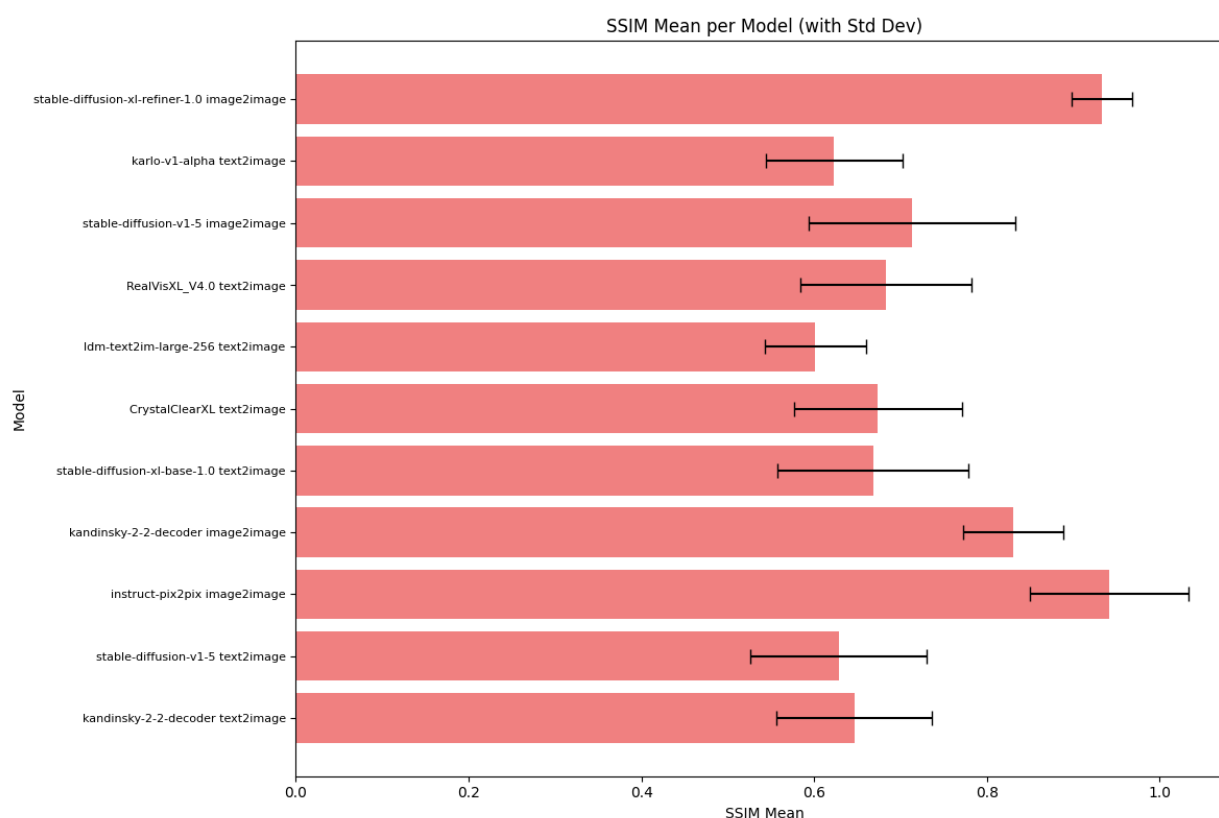


Figure 19: SSIM score for the eleven generative models used for evaluation. As SSIM is calculated for each pair of real-synthetic images we report the average SSIM for all the 1600 pairs, alongside standard deviation. Higher values are considered better.

Table 31: Average scores for the quality metrics, across image-to-image and text-to-image generative models.

Model Type	FID	KID Mean	LPIPS Mean	SSIM Mean
image-to-image	39.36	0.0128	0.3528	0.8545
text-to-image	74.02	0.0285	0.6348	0.6463

The analysis of average quality metrics reveals distinct performance differences between the image-to-image and text-to-image models (see Table 31). Image-to-image models achieve lower scores for FID KID and LPIPS compared to text-to-image models, and higher regarding SSIM, indicating that image-to-image outputs generally resemble their input images more closely. These metrics suggest image-to-image models' particular strength in preserving perceptual similarity and structural fidelity, making them ideal for cases requiring detailed, close approximations of reference images. Despite these strengths, however, text-to-image models typically produce images that are more realistic and visually compelling, as depicted in Figure 20 compared to Figure 21. While the higher metric scores might initially imply a weaker performance, they often reflect the flexibility of text-to-image models to create entirely new, high-quality images without dependency on a reference image. Text-to-image outputs often exhibit higher resolution, fewer distortions, and minimal noise, leading to visually coherent and high-definition results, which aligns them more closely with real-world image characteristics. This versatility makes text-to-image models highly effective for generating images where realism and quality are prioritized over strict adherence to specific reference details. On the other hand, image-to-image generative models replicate the content of the input image, but the results are often lacking in realism. This is particularly evident in the case of faces, where the generated outputs are far from convincing, as all methods tend to produce distorted and unnatural images. However, the quality metrics are generally higher on average, making them unsuitable for accurately assessing the true quality and realism of the generated images. For example, from Figure 22 it is evident that models such as stable diffusion can generate realist images of

landscapes that is also reflected in optimal value sin the aforementioned metrics. On the other hand, in Figure 23 we can observe am example of a synthetic image that is easily detected as synthetic due to the nonsense characters in it, but due to its simplicity it's among the images with the most optimal quality metrics.



Figure 20: Examples of text-to-image generative models. Given the caption of the real image as provided by the Visual News dataset, the generative models create a synthetic image with the same content.



Figure 21: Examples of image-to-image generative models. The models that require also a text prompt the instruction was “replicate the content of the input image”.



Figure 22: Example of stable diffusion v1.5 in producing landscapes.

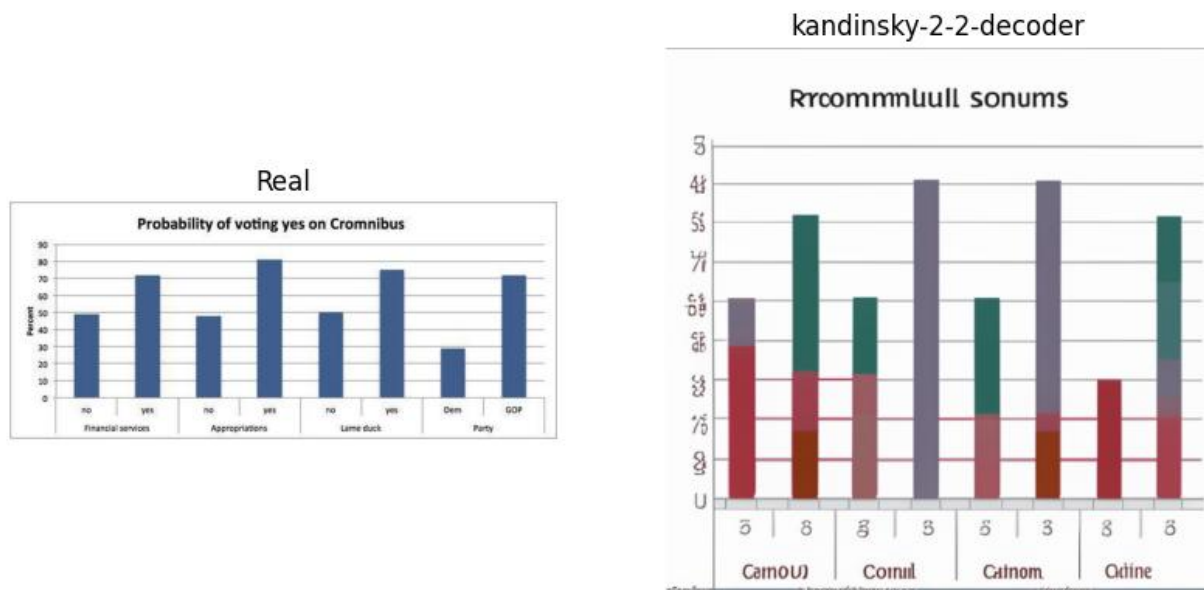


Figure 23: Generated synthetic diagram with optimal quality metrics.

5.4 Conclusion

The widespread availability and accessibility of generative models present significant implications in the context of disinformation. While many open-source repositories on platforms like GitHub host generative models that are highly popular and well-adopted within the research community, the barrier to using these models has drastically lowered for non-experts. Pre-trained models hosted on platforms such as Hugging Face and frameworks like the Diffusers library have homogenized and simplified the execution of these models, making them far easier to deploy. Although dedicated hardware, such as GPUs, is required to run these models efficiently, the necessary resources are increasingly accessible at a reasonable cost. Furthermore, the Internet is abundant with step-by-step guidelines for installing and executing these tools. With the assistance of generative AI systems like ChatGPT, even users with minimal technical expertise can set up and run these models, generating synthetic images within minutes. This ease of use significantly enhances the potential risks associated with these technologies, particularly in scenarios involving malicious applications like disinformation.

The core issue in the current landscape is that the realism and quality of generated images remain relatively low, often requiring extensive experimentation to achieve convincing results. In the context of disinformation, quality assessment of image generative algorithms reveals two key findings. First, despite significant advancements, even state-of-the-art diffusion generators often produce low-quality images with distorted faces and bodies, making them relatively easy to identify as synthetic. Most generative models produce images that have a characteristic look and feel, distinguishing them from real images, after careful manual observation. Additionally, not all generative models are designed or trained to produce realistic images; many focus on stylistic or artistic outputs rather than photorealism. However, certain models, such as CrystalClearXL v1¹⁷¹ and RealVisXL v4¹⁷², stand out by generating highly convincing and realistic images, highlighting their potential risk in disinformation campaigns. Second, commonly used and well-established quality metrics in the literature are inadequate for real-world risk assessment as they fail to accurately capture the realism of generated images. These metrics may also be influenced by the complexity of the images; for example, it may be easier to generate realistic synthetic images of simple landscapes or scenes with few objects and individuals, while more complex images present greater challenges. This gap highlights the urgent need for the development of more robust and reliable evaluation metrics that can better assess both the quality and realism of synthetic images, especially in diverse and complex scenarios. Also, it raises the need for a more systematic evaluation of how image complexity impacts quality and realism to better understand the risks posed by these

¹⁷¹ <https://huggingface.co/stablediffusionapi/crystal-clear-xl-v1>

¹⁷² https://huggingface.co/SG161222/RealVisXL_V4.0

technologies in a disinformation context.

Finally, an important dimension of this work, would be to evaluate the models' capacity not only to replicate the content but to potentially alter the visual depiction of the original images. Such alterations could yield images suitable for disinformation scenarios, where subtle changes to visual elements might influence viewer perception or misrepresent events. However, exploring the models' ability to intentionally generate altered images for disinformation purposes remains beyond the scope of this study and is left as future work. This future direction would enable a deeper understanding of how generative models might be adapted or misused to craft visually misleading content, further informing safeguards against synthetic media in critical contexts.

6 Discussion

Following the assessments provided in Sections 2-5, we complement them with a discussion on how the ongoing rapid development of generative AI affects the open problems and challenges. The discussion reflects the three main roles, in which generative AI is studied and applied in the AI-CODE project, namely: 1) generative AI being misused to generate disinformation, 2) generative AI as a tool to detect disinformation, and finally, 3) generative AI as a booster to support trusted information production; as well as two main modalities studied in the AI-CODE project, namely text and images.

Further discussion from the domain-oriented perspective of credibility and trust assessment, disinformation detection, and trusted information production will be provided in dedicated deliverable D2.3. That deliverable will, furthermore, provide actionable technological recommendations that are supposed to direct development of AI-CODE services during the rest of the AI-CODE project implementation.

6.1 Assessment of the potential of text generative AI to generate and to detect disinformation

As our results clearly demonstrate (Section 2.2), the latest state-of-the-art LLMs suffer from serious vulnerabilities that allow bad actors to effectively and cheaply generate vast amounts of non-credible or even disinformation content. While the generated disinformation text is of very high quality in high-resource languages (such as English), and the capabilities of LLMs are improving for low-resource languages (e.g., Greek), also the quality of fully automatically generated text is radically advancing. Considering this finding, we can expect that LLMs may have already been adopted to fabricate disinformation.

Unfortunately, to the best of our knowledge, **there is no study that thoroughly examines the prevalence of machine-generated text in disinformation data**. At the same time, such study would shed light on the extensive discussion of how much generative AI is or is not a real threat that can cause a significant increase of non-credible content in online spaces. On one hand, some researchers and practitioners claim that it is not such an issue because generated text still needs to be spread by the conventional channels (e.g., by high-influence social media accounts or a network of coordinated authentic/bot accounts) so having a large amount of generative data is not enough by itself. On the other hand, there are voices claiming such large amounts of generated (disinformation) content can flood the online environment (e.g., by creating the whole fully-generated online news portals spreading a specific type of disinformation narratives and propaganda, such as *Pravda Network* – a highly sophisticated ecosystem of non-credible news portals spreading AI-generated pro-Russia disinformation content in various languages) and the public may struggle to distinguish such portals from legitimate ones.

Therefore, in our next work in AI-CODE WP2, we aim to perform **the first systematic large-scale study of prevalence of machine-generated texts in disinformation datasets**. Furthermore, since our overall focus is on multilinguality, we should attempt to do this study on available multilingual datasets.

This kind of research, however, requires highly accurate and robust detectors of machine-generated text. Current detectors, which exhibit low accuracy on out-of-distribution data (as we showed in our benchmark of a range of machine-generated text detection methods in multilingual as well as in adversarial settings in Section

2.3), may result in a significant number of false positives or false negatives and thus introduce a considerable amount of noise and invalidate the study results. For this reason, our next steps should be to **train robust machine-generated text detectors** that generalize well also on out-of-distribution data. This will allow to use them also on unlabelled, highly diverse real-world data.

Following the results from additional assessed tasks of check-worthy claim detection and matching (Section 2.5), we recognize the **need to extend the assessment how the current solution can perform and can be improved on additional languages**. This is motivated by the fact the currently available solutions are limited to languages they have been trained on. At the same time, we may expect that such solutions, after being deployed and integrated into the AI-CODE services, will be likely used in highly multilingual settings, i.e., media professionals may input texts in many languages. To achieve this, our findings on low-resource text classification (Section 2.6) may become handy and decrease the need for large-scale labelled datasets.

Finally, generative AI models such as LLMs require not only large datasets to be trained but also have substantial computational and hardware requirements for training as well as for inference. This increases the direct costs of developing and running services relying on such models and it also has negative environmental consequences in form of increased CO₂ emissions. At the same time, as our results have already shown (see Section 2.6.1), smaller specialised models can still outperform large models with enough fine-tuning. As we recognize this challenge, we will continue to **research and assess the use of smaller models as well as efficient fine-tuning techniques aimed at lowering the environmental footprint** of the AI-CODE services.

6.2 Assessment of the potential of text generative AI for trusted information production

By the assessment of potential as well as threats of generative AI, we recognized that the **research and development of solutions for trusted information production is generally significantly lagging behind the efforts to use it for combating disinformation**. This may be due to various reasons. Firstly, media professionals may be reluctant to adopt such solutions due to official journalistic guidelines (some media organizations prohibit all use of generative AI in news production, see Section 3.1 for more details). Another reason may be that media professionals are not familiar with generative AI models and their actual capabilities (e.g., they are not aware of use cases when the AI models are supposed to work well and when not). The situation is even more complex because of the fact that some generative models, especially state-of-the-art LLMs, are capable to generate credible-looking content, even in situations where they lack the necessary knowledge and content may be fully or partially affected by hallucinations.

We recognize that **there is no study that would analyse how media professionals interpret trustfulness and credibility of generative AI outcomes**. An interesting and important question is how well-known human cognitive biases (e.g., confirmation bias or Dunning-Kruger effect) influence human interpretation of LLM outputs. By having such a knowledge, we may be able to implement AI-CODE services in order to prevent unintended over-reliance or misinterpretation of generative AI. To address this gap, our next planned step in this direction is to perform a **survey of human cognitive biases in the context of generative AI and its trustfulness based on the psychology literature**.

Despite the increasing capabilities of generative AI, we acknowledge that its adoption by media professionals to generate full news articles may still be considered too unreliable. Instead, we see its potential in more human-centric or human-in-the-loop settings, e.g., to support media professionals to improve or adjust their human-written texts for specific purposes. To this end, we **identified an interesting potential of generative AI for multilingual text paraphrasing and summarization** (Section 3.2). It can be applied in various use cases, such as summarizing the text of a whole news article to be shared on social media in the form of shorted social media post. In our future work, we are interested particularly in examining how such a potential can be used for **generating explanatory counternarratives for factual claims** from the fact-checking articles.

6.3 Assessment of AI to detect synthetic image disinformation

The assessment of AI-Generated / Synthetic Image Detection (SID) capabilities reveals an important gap between benchmark performance and real-world applications. While existing state-of-the-art models achieve impressive results on curated datasets, as illustrated in Section 4, their reliability degrades significantly when faced with uncontrolled, real-world conditions such as image compression, filtering, and cropping introduced by social media platforms. Our evaluation using in-the-wild datasets, with images collected from social media, confirms that such transformations impair detection robustness. This underlines the need for training pipelines that reflect the diversity typical of user-generated content. Furthermore, the differences in performance observed across different SID models highlight the role of architectural design, the composition of the used training datasets, and the importance of data augmentation strategies during training. Models with more advanced backbone architectures tend to generalize better, especially when combined with diverse and representative training data that capture the visual artifacts and noise present in real-world platform data. Similarly, augmentation strategies that simulate common platform transformations, such as recompression, resizing, cropping, colour filtering, etc., can improve model resilience by exposing it to the types of distortions it will encounter on images uploaded to digital platforms. **These results underscore the need for detection pipelines that do not target benchmark data in the lab but instead are designed to handle the content circulating online.** In practice, this means developing models that can maintain robust performance across various social media platforms, image types, and emerging generative techniques, ensuring detection systems remain effective as synthetic content generation methods evolve.

Regarding emerging generative techniques, detectors trained on older generative models often fail to recognize content produced by newly released generation techniques. This highlights a fundamental limitation of current detection pipelines following the supervised learning paradigm. **They require continuous retraining or fine-tuning to remain effective.** However, this process is resource-intensive and forms a cat-and-mouse dynamic where new generative models outpace detection capabilities. Maintaining robust detectors demands an ongoing collection of fresh synthetic content, periodic retraining or adaptation of models, and, crucially, systematic evaluation on new data to detect potential distribution shifts. **Without this continuous monitoring and model maintenance, detectors risk becoming obsolete, leading to undetected synthetic content and eroding trust in the system.** These challenges motivate the exploration of training-free approaches in recent works, such as anomaly detection, which seek to model the distribution of real images and flag deviations without depending on the specifics of known generative models. These methods offer a more scalable and future-proof alternative, but they too require rigorous benchmarking to ensure their effectiveness in the wild.

Another limitation of current Synthetic Image Detection (SID) models is their inability to accurately localize tampered regions, especially in cases where the images are fully generated. This limits the usefulness of detection results for disinformation analysis, where identifying specific manipulations is often crucial. More broadly, there remains a significant gap between synthetic content detection and the broader goal of combating disinformation. **Simply detecting that an image is AI-generated is not sufficient, as synthetic content is increasingly used in benign contexts such as art, humour, or entertainment.** What is missing is the ability to interpret SID outputs within a meaningful context, distinguishing potentially misleading and/or harmful uses from harmless ones. To bridge this gap, detection systems must provide explainable outputs and integrate contextual information that helps users assess intent and impact. Additionally, allowing configurable decision thresholds, for example, tuning for lower false positives or higher recall, enables stakeholders such as fact-checkers and media professionals to adapt the system to their specific needs. **Ultimately, SID models must evolve from isolated classifiers to context-aware tools that can support human judgment in disinformation scenarios.**

6.4 Risk assessment of generative AI

The widespread availability and growing accessibility of image generative AI models present considerable risks in the context of disinformation. Our systematic analysis of 70 prominent generative models in Section 5 (from generative families such as GANs and Diffusion models) reveals that many are open-source, well-

documented, and easily deployable through platforms like GitHub and Hugging Face. Frameworks such as the Diffusers library further simplify their use, reducing the barrier to entry for non-experts. **With minimal technical knowledge and guidance from tools like ChatGPT, users can produce synthetic images within minutes.** Moreover, while high-end hardware accelerates generation, even mid-budget consumer GPUs are sufficient to run many of these models locally. Although the generation speed may be relatively slow, it is still fast enough to create a substantial volume of convincing fake content over time. Regarding the quality of the generated content, while many models generate low-fidelity outputs with noticeable artifacts, several other **models, such as CrystalClearXL and RealVisXL, produce highly realistic images, making them particularly concerning for disinformation campaigns.** Importantly, these models can be used not only for image synthesis but also for enhancement tasks such as super-resolution and text-to-image generation, which can be exploited to create visual narratives with minimal effort.

A key insight from this assessment is that the quality metrics commonly used in generative model research (such as Fréchet Inception Distance) are often insufficient for real-world risk evaluation. These metrics frequently prioritize aspects such as how well a model follows input instructions (e.g., in text-to-image generation) or how closely it replicates the composition of a reference input image. But they often overlook the perceived realism and subtle visual inconsistencies that are critical in disinformation contexts. As a result, models that produce clearly synthetic or distorted images may still achieve high metric scores. Conversely, models that improvise more creatively, generating highly photorealistic images, may receive lower scores despite posing a greater risk of visual deception. This misalignment reveals a **gap in current evaluation practices and underscores the need for new metrics that better reflect the human perception of realism and the potential for misuse in disinformation scenarios.** Addressing these concerns requires the development of more nuanced evaluation metrics, deeper understanding of how image complexity impacts realism.

Moreover, while most existing analyses focus on realism, the potential of generative models to alter rather than merely synthesize content introduces a more nuanced risk. Subtle manipulations of existing visuals, such as removing individuals from scenes or modifying expressions could profoundly shape public perception without triggering an authenticity alarm. In addition, these images will have high quality and realism as most of the content will be pristine. Therefore, future work needs to target how generative models may be repurposed to produce persuasive but misleading imagery.

7 Conclusions

In this deliverable, we have summarized the research performed in tasks *T2.2 Generative AI methods for text generation* and *T2.3 Trustworthy AI methods for synthetic media detection*. We assessed the state-of-the-art generative AI models and approaches for both modalities and their current positive as well as negative uses.

The main contributions and conclusions of our work are as follows:

- **Generation of disinformation (Section 2.2):** Following and building upon previous research works, we have shown that although state-of-the-art LLMs can be used to generate disinformation that can be mistaken for human-written in case of high-resource languages, their performance in producing error-free convincing text is generally worse for low-resource languages such as Greek. On the other hand, we successfully demonstrated that LLMs can be used to judge generated texts (e.g., on their grammatical structure or coherence) which can alleviate the need for human manual work.
- **Detection of machine-generated text (Section 2.3):** We have shown the benefits of multilingual fine-tuning of multilingual detectors as well as that the adversarial retraining using a simple data augmentation by obfuscated texts can significantly increase the adversarial robustness of detectors. In addition, we have provided new datasets of human-written and machine-generated texts and tested the competitiveness of our solution (based on fine-tuned LLMs) in a shared task against other competing solutions.

- **Logical fallacy detection (Section 2.4):** We provided two new annotated datasets for logical fallacy detection (English for the news domain and Italian for the social media domain) and experimented with several detection approaches. Currently, smaller specialized (fine-tuned) models outperform larger models (LLMs) in zero-shot settings. On the other hand, large models can also be fine-tuned (in a parameter-efficient way), which, as our experiments show, improves their performance on the task.
- **Check-worthy claim detection and matching (Section 2.5):** We have assessed the check-worthiness detection in both the domain of news articles (in English) as well as social media (in Italian). We experimented with both smaller BERT-like models focusing on multi-task learning as well as with parameter-efficient fine-tuning of an LLM-based detection method. We found both approaches to improve over the baseline performance.
- **Low-resource text classification (Section 2.6):** Based on the results of our experiments in this part (but also in line with the results shown in Sections 2.4 and 2.5), we can conclude that fine-tuned smaller specialized models can still outperform general large models used in zero-shot or even few-shot settings if there is enough data available for fine-tuning. In low-resource settings, there are several strategies which can be used to select samples for few-shot (in-context) learning. Dataset augmentation can also help to provide more data, although the applicability of this approach depends on the task at hand (e.g., it is not usable for training detectors of machine-generated text).
- **Verdict production for fact checking (Section 3.2):** The results demonstrated that our current LLM-based approach outperformed its predecessor across multiple languages, with precise, multi-layered prompt engineering significantly enhancing the accuracy and coherence of the outputs.
- **Paraphrasing and the notion of authorship (Section 3.3):** Our findings suggest that authorship, when using LLMs for paraphrasing of human-written text, should be task-dependent and it should be evaluated differently in situations where the content is more important as in situations where the style is more important (since LLM-paraphrased texts tend to drift away from the style of the original toward the style of the used LLM); another aspect to consider is the portion of the generated text.
- **Detection of synthetic image disinformation (Section 4):** The analysis demonstrated that synthetic image detection remains an unresolved challenge, with no detection method emerging as clearly superior due to the complexity and variability of generative models. Additionally, the impact of image transformations, such as those commonly applied on social media platforms, further complicates the assessment of detector performance. Further benchmarking with newer methods and more challenging datasets is required.
- **Risks of image generative AI (Section 5):** The widespread accessibility of generative models, facilitated by pre-trained models on platforms like Hugging Face and user-friendly frameworks like Diffusers, has lowered the barrier for non-experts to generate synthetic images, increasing the risk of misuse in disinformation. While most models produce low-quality images with distortions, some generate highly realistic images, posing a greater threat. Current quality metrics are inadequate for assessing realism and are influenced by image complexity, underscoring the need for more robust evaluation methods. Future research should explore the potential of these models to intentionally alter images for disinformation, enabling a deeper understanding of their misuse in crafting misleading content.

Our remaining work in WP2 will be carried out as a part of (already running) task *T2.4 AI-based credibility and trust assessment and disinformation detection tools*. The goal of this task is to critically evaluate and coalesce the findings and outcomes from T2.2, T2.3 presented in this deliverable with the findings from task *T2.1 Decentralised and immersive social media* presented in the deliverable D2.1 together with the current capabilities of the state-of-the-art solutions on both, credibility and trust assessment as well as disinformation detection. Besides their applicability for conventional media, a special attention is also given to investigate how the existing tools can be adapted to the next-generation social media and AI generated content; together with identification of any necessary adaptation changes that will serve as a valuable input for development of services envisioned in the AI-CODE project.

More specifically, as a part of our remaining work, **we will build on the current outcomes and findings of**

tasks T2.2 and T2.3, covered by this deliverable, as follows:

- In machine-generated text detection, we aim to further evaluate multi-domain and cross-domain performance of detection methods and to increase generalizability of their performance (their robustness) also towards emerging next generation social media. All these results will be used to inform the design of a component that will be integrated into the planned AI-CODE services (especially in context of *Trustability and credibility assessment for content and sources*).
- In the case of logical fallacy detection and check-worthy claim detection, we aim to explore possible extensions of the experiments (e.g., for other languages or combining the approaches). Since both tasks are highly relevant in context of *Trustability and credibility assessment for content and sources* as well as of *Personal companion for understanding disinformation for media professionals*, their feasibility and technological readiness will be assessed and specific recommendations for AI-CODE services will be made.
- In verdict production for fact checking, to achieve optimal results, we plan to incorporate fine-tuning techniques in the next stages of the project. Additional models may also be considered for further evaluation and refinement. These findings and outcomes will directly feed into *Counter-narrative Recommendation* module planned as a part of *Generative AI Interactive Coaching Service & Dynamic Simulator for Media Professionals*.
- We plan to extend the approach outlined in Section 4 to video content. While this study focused on benchmarking synthetic image detection, in the next phase, we aim to broaden the benchmarking framework to include fake or manipulated videos. This extension will particularly emphasize deepfake videos, with a focus on face manipulation. By incorporating videos into our analysis, we hope to provide a comprehensive evaluation framework that addresses the unique challenges of detecting synthetic content across both static and dynamic media. This will feed into the planned *Media Asset Annotation and Management* service.
- Synthetic image detection, MGT detection system and eventually synthetic and manipulated video detection will be leveraged for other services within AI-CODE, namely the *Trust and Credibility Assessment Service*. This service will integrate synthetic content detection tools, developed based on the work presented in this deliverable, as supplementary signals for credibility evaluation.
- Based on the analysis of the current state of generative models and the performance of state-of-the-art detection models related to synthetic content, claim detection, fallacy detection and all the other tasks presented in this document, we aim to further refine the requirements for a *Transparency* service in model cards. Our focus will include providing detailed performance metrics and addressing ethical considerations, such as measurements related to bias and fairness, to ensure a comprehensive and responsible approach.

Work carried out in WP2 flows into the requirements process that is being carried out in WP3. Specifically, the requirements process brings together end-user needs and the technical ambitions of the AI-CODE project. The requirements process defines how novel, easy-to-use services shall be designed and developed to take advantage of new technological developments, while addressing the needs of the users and the intended context of use within the scope of the AI-CODE project. The findings reported in this deliverable, and the remainder of the work carried out in WP2, make an important contribution to the alignment of the technical potential of Generative AI and with the needs of AI-CODE stakeholders. WP2 helps to ensure that AI-CODE services are incorporating state-of-the-art technological innovations that address the real needs of media professionals. To further reinforce this goal, in the upcoming deliverable D2.3, we will reflect on the identified challenges and open problems discussed in Section 6 and provide actionable technological recommendations intended to direct development of AI-CODE services during the rest of the AI-CODE project implementation.