OMMM 2025

Proceedings of the First Interdisciplinary Workshop on Observations of Misunderstood, Misguided and Malicious Use of Language Models

associated with

The 15th International Conference on Recent Advances in Natural Language Processing 2025

September 11th, 2025
Varna, Bulgaria

The First Interdisciplinary Workshop
Observations of Misunderstood, Misguided
and Malicious Use of Language Models
Associated with the International Conference
Recent Advances in Natural Language Processing 2025
PROCEEDINGS
Varna, Bulgaria
11th September 2025

ISBN 978-954-452-101-1

Designed by INCOMA Ltd.
Shoumen, BULGARIA

Message from the Program Chairs (TODO)

Welcome to the proceedings of the first edition of the Interdisciplinary Workshop on Observations of Misunderstood, Misguided and Malicious Use of Language Models: (OMMM 2025), hosted at the 15th Biennial Conference on Recent Advances in Natural Language Processing (RANLP 2025), in Varna, Bulgaria.

OMMM 2025 is a new endeavour with the purpose of drawing together communities studying the inappropriate and harmful uses of Large Language Models (LLMs). In particular, the organising comittee is made up of experts from natural language processing, human computer interaction and psychology. Through this venture we aim to share common perspectives on the capabilities, vulnerabilities and harmful applications of LLMs. Our aim is to foster a new community drawn from various disciplines within and beyond our own, which is focussed on the mitigation of potential harms from the ever increasing ubiquity of AI technology powered by LLMs.

The use of Large Language Models (LLMs) pervades scientific practices in multiple disciplines beyond the NLP/AI communities. Alongside benefits for productivity and discovery, widespread use often entails misuse due to misalignment of values, lack of knowledge, or, more rarely, malice. LLM misuse has the potential to cause real harm in a variety of settings. Through this workshop, we aim to gather researchers interested in identifying and mitigating inappropriate and harmful uses of LLMs. For the purposes of designing a programme and motivating submissions, we categorised the misuses of LLMs into three domains:

- **Misunderstood** usages: Misrepresentation, improper explanation, or opaqueness of LLMs. Including: The use of anthropomorphic language by or for LLMs; Attributions of consciousness to LLM agents; Interpretability of LLM outputs or decisions; and harms arising from overreliance or misplaced trust in LLMs.
- **Misguided** usages: Misapplication of LLMs where their utility is questionable or inappropriate. Including: underperformance and inappropriate applications; structural limitations and ethical considerations; and deployment without proper training or safeguards.
- **Malicious** usages: Use of LLMs for misinformation, plagiarism, and adversarial attacks. Including: Adversarial attacks, jailbreaking; Detection and watermarking of machine-generated content; Generation of misinformation or plagiarism; and bias mitigation and trust design.

This year, we received 13 submissions to the workshop. These submissions covered a variety of current topics of interest in line with the aims of the workshop. In particular, the organisers noted submissions on anthropmorphised descriptions of LLMs, including new datasets for identification of anthropmorphisation; Bias as applied to large language models and the downstream harmful effects; case studies including negative results where LLMs failed compared to traditional approaches; the detection of AI generated texts; and work on AI alignment.

All submissions were peer-reviewed by the members of the program committee which includes specialists drawn from NLP, Philosophy,, Psychology, AI Ethics, LLM Security, and Misinformation. The organisers provided a further meta-review for all submissions, summarising the outcomes and decision as well as offering additional feedback. Out of the 13 submissions to the workshop, 9 were accepted, 2 were rejected and a further 2 papers were accepted subject to improvements in line with reviewer feedback. Each PC member had no more than three assignments. The organisers were delighted to see so many papers submitted in line with the mission of the workshop, demonstrating the necessity of such an event and the nascent community surrounding it.

The workshop is held in-person, with online attendance for authors who were unable to attend. The program encompasses: An introductory session ran by the organisers covering the grand challenges of misunderstood, misguided and malicious use. The programme then consists of 3 sessions, one covering papers submitted that are relevant to each of the topics: 6 papers were presented in the first session on *Misunderstood Use*. 2 papers were presented in the second session on **Misguided Use**. Finally, 3 papers were presented in the third session on **Malicious Use**.

Each session was succeeded by a discussion session, culminating in a final discussion session to close the event. The organisers intend to use the results of the discussions to co-create with the participants a future publication on the grand challenges of LLM Misuse.

We would like to thank the members of the program committee for their timely help in reviewing the submissions and all the authors for submitting their papers to the workshop. We also thank the organisers of RANLP for hosting the workshop and their kind support in producing these proceedings. Additionally, our thanks go to those who maintain the ACL Anthology in which these proceedings appear.

OMMM Organizing Committee

Piotr Przybyła, Matthew Shardlow, Nanna Inie, Clara Colombatto

Organizing Committee

- Piotr Przybyła, Pompeu Fabra University and Institute of Computer Sciences, Polish Academy of Sciences
- Matthew Shardlow, Manchester Metropolitan University
- Nanna Inie, IT University of Copenhagen
- Clara Colombatto, University of Waterloo

Programme Committee

- Alina Wróblewska (Polish Academy of Sciences)
- Ashley Williams (Manchester Metropolitan University)
- Azadeh Mohammadi (University of Salford)
- Clara Colombatto (University of Waterloo)
- Dariusz Kalociński (Polish Academy of Sciences)
- Julia Struß (Fachhochschule Potsdam)
- Lev Tankelevitch (Microsoft Research)
- Leon Derczynski (NVIDIA)
- Marcos Zampieri (George Mason University)
- Matthew Shardlow (Manchester Metropolitan University)
- Nael B. Abu-Ghazaleh (University of California, Riverside)
- Nanna Inie (IT University of Copenhagen)
- Nhung T. H. Nguyen (Johnson & Johnson)
- Nishat Raihan (George Mason University)
- Oluwaseun Ajao (Manchester Metropolitan University)
- Peter Zukerman (University of Washington)
- Piotr Przybyła (Universitat Pompeu Fabra)
- Samuel Attwood (Manchester Metropolitan University)
- Sergiu Nisioi (University of Bucharest)
- Xia Cui (Manchester Metropolitan University)

Table of Contents

Bias in, Bias out: Annotation Bias in Multilingual Large Language Models

Xia Cui[1], Ziyi Huang[2], Naeemeh Adel[1]
[1]Manchester Metropolitan University, Manchester, UK.
{x.cui, n.adel}@mmu.ac.uk
[2]Hubei University, Wuhan, China.
ziyihuang@hubu.edu.cn

Abstract

Annotation bias in NLP datasets remains a major challenge for developing multilingual Large Language Models (LLMs), particularly in culturally diverse settings. Bias from task framing, annotator subjectivity, and cultural mismatches can distort model outputs and exacerbate social harms. We propose a comprehensive framework for understanding annotation bias, distinguishing among *instruction bias*, *annotator bias*, and *contextual and cultural bias*. We review detection methods (including inter-annotator agreement, model disagreement, and metadata analysis) and highlight emerging techniques such as multilingual model divergence and cultural inference. We further outline proactive and reactive mitigation strategies, including diverse annotator recruitment, iterative guideline refinement, and post-hoc model adjustments. Our contributions include: (1) a structured typology of annotation bias, (2) a comparative synthesis of detection metrics, (3) an ensemble-based bias mitigation approach adapted for multilingual settings, and (4) an ethical analysis of annotation processes. Together, these contributions aim to inform the design of more equitable annotation pipelines for LLMs.

1 Introduction

Large Language Models (LLMs) such as BERT (Devlin et al., 2019), T5 (Raffel et al., 2020), Llama (Touvron et al., 2023) and GPT-4 (Achiam et al., 2023) have transformed Natural Language Processing (NLP), achieving state-of-the-art performance across a wide range of tasks. Their success is largely attributed to pre-training on vast, unannotated corpora that enable them to learn powerful representations. However, aligning these models with human values and adapting them for high-stakes applications requires smaller, curated datasets annotated by humans.

This reliance introduces a critical vulnerability. Annotation bias, which refers to systematic distortions introduced during the labelling process, can severely affect model performance, fairness, and generalisation. It may arise from task framing, annotator subjectivity, or cultural mismatches, and its impact is particularly pronounced in multilingual and culturally heterogeneous contexts (Bender and Friedman, 2018; Plank, 2022).

The consequences of annotation bias are not hypothetical. For example, models trained to detect toxicity often misclassify African-American Vernacular English (AAVE) as offensive, due to cultural insensitivity in both annotation guidelines and annotator interpretation. Phrases such as "That's my nigga" which carry a supportive meaning in AAVE, are frequently labelled as hateful by annotators unfamiliar with the dialect (Sap et al., 2019). This highlights how linguistic and cultural assumptions embedded in the annotation process can lead to unjust model behaviour.

Such failures reflect a broader pattern. When biased annotations are used for training or fine-tuning, models tend to replicate and sometimes amplify these distortions, resulting in both representational harms and disparities in performance across demographic groups (Dodge et al., 2021; Sheng et al., 2019). Addressing these issues requires critical scrutiny of annotation workflows, with careful attention to cultural and contextual diversity.

In this paper, we examine the sources and consequences of annotation bias in multilingual LLMs. We propose a typology of annotation bias, encompassing instruction bias, annotator bias, and contextual or cultural bias. We review established and emerging detection methods, including inter-annotator agreement, model disagreement, and multilingual divergence. We adapt Weak Ensemble Learning (WEL) as a reactive mitigation strategy

Proceedings of the First Interdisciplinary Workshop on OMMM associated with RANLP 2025,
pages 1–16, Varna, Bulgaria, Sep 8-10, 2025.
https://doi.org/10.26615/978-954-452-101-1-001

and assess its effectiveness across multilingual and real-world datasets. Finally, we reflect on the ethical and labour implications of annotation work and suggest directions for building more inclusive and context-aware NLP pipelines.

2 Background and Motivation

Early annotation practices in NLP were shaped by linguistic theory and typically involved trained experts using detailed, rule-based guidelines. Datasets such as the Penn Treebank (Marcus et al., 1993) and FrameNet (Baker et al., 1998) exemplified this approach, producing consistent annotations at a small to moderate scale.

As NLP tasks expanded and model complexity increased, the field shifted toward large-scale annotation through crowdsourcing platforms (Snow et al., 2008a). This approach enabled the creation of widely used datasets like SNLI (Bowman et al., 2015) and MultiNLI (Williams et al., 2018), but introduced new concerns regarding annotation quality, consistency and subjectivity.

The rise of LLMs has further complicated annotation workflows. Today's datasets often combine expert review, crowdworker input, and semi-automated methods such as model-in-the-loop annotation (Pawar et al., 2025). Many target inherently subjective or ambiguous constructs, including helpfulness, safety, or moral alignment (Monarch, 2021; Uma et al., 2022). These tasks are particularly vulnerable to variation across annotators and contexts.

At the same time, NLP has increasingly embraced multilingual and multimodal benchmarks. Projects such as XTREME (Hu et al., 2020), AmericasNLI (Ebrahimi et al., 2022), TextVQA (Singh et al., 2019), and HowTo100M (Miech et al., 2019) highlight the challenges of applying traditional annotation schemas across languages, cultures, and modalities. Multimodal tasks introduce further complexity through temporality, affective signals, and cross-modal interpretation.

These trends have exposed a structural vulnerability: annotation bias. This includes not only individual annotator subjectivity, but also culturally conditioned assumptions, linguistic mismatches, and platform-mediated incentives (Blodgett et al., 2020; Plank, 2022). Annotation decisions made on small but influential datasets can propagate through model fine-tuning and evaluation, leading to downstream harms (Dodge et al., 2021).

In response, the field has developed ethical documentation frameworks such as *Data Statements* (Bender and Friedman, 2018) and *Datasheets for Datasets* (Gebru et al., 2021). These initiatives promote transparency by capturing the dataset's linguistic, demographic, and procedural context. They represent an important step toward recognising that high-quality, ethical NLP systems begin with well-understood and well-documented data.

3 Types of Annotation Bias

Annotation bias in NLP arises when human interpretations, cultural assumptions, or task formulations systematically distort labelled data. These biases can affect model learning, especially during fine-tuning and evaluation. In the context of LLMs, annotation bias often originates from multiple sources and compounds over the pipeline. Addressing it requires distinguishing among different types of bias and understanding how they interact.

We categorise annotation bias into three primary types based on its origin: **Instruction Bias** (Section 3.1), **Annotator Bias** (Section 3.2), and **Contextual and Cultural Bias** (Section 3.3). These types are not mutually exclusive. In many cases, a biased annotation reflects an interaction among all three. For example, culturally narrow task guidelines (instruction bias) given to a homogeneous annotator pool (annotator bias) tasked with labelling dialectal language (contextual bias) may produce systematically skewed data. Recognising this interplay is essential for designing effective detection and mitigation strategies (Bender and Friedman, 2018).

3.1 Instruction Bias

Instruction bias (Parmar et al., 2023) occurs when the design of an annotation task, including the prompt wording, labelling guidelines, or interface, embeds implicit assumptions that shape how annotators interpret or respond. These assumptions can systematically distort the resulting labels.

A common example appears in sentiment analysis, where annotators are often asked to classify texts as "positive", "negative", or "neutral". These categories overlook cultural and linguistic nuance, such as expressions of irony, ambivalence, or indirectness (Mohammad, 2016). The framing tends to reflect Western emotional norms that do not generalise across diverse populations (Huang and Yang,

2023). Similarly, toxicity detection tasks have been shown to mislabel minoritised dialects as offensive, in part due to annotation instructions that lack sociolinguistic sensitivity (Sap et al., 2019).

In the context of LLMs, instruction bias is further complicated by the use of prompts in place of formal annotation guidelines. *Zero-shot* (Wei et al., 2022) and *few-shot* prompting (Schick and Schütze, 2022) methods often replace expert-designed protocols. These prompts, though brief, function as implicit task instructions and strongly influence model behaviour. Minor changes in phrasing, such as asking "Is this inappropriate?" versus "Is this morally wrong?", can lead to significantly different model outputs, especially for subjective or value-laden tasks (Zhao et al., 2021; Schick and Schütze, 2022; He et al., 2024).

Moreover, prompts are frequently written by researchers or practitioners who come from specific cultural or disciplinary contexts. Their assumptions shape how tasks are framed and what kinds of answers are considered valid. For example, in mental health detection tasks, prompt templates often reflect Western norms of psychological distress. This reduces model performance on data from underrepresented linguistic or cultural groups (Parmar et al., 2023; Cui et al., 2024). Unlike traditional annotation guidelines, prompts are rarely revised or reviewed through participatory validation processes (Zamfirescu-Pereira et al., 2023; Cui et al., 2024).

3.2 Annotator Bias

Annotator bias arises from the individual or collective predispositions of those performing the labelling. These may include cognitive heuristics, beliefs, social norms, or demographic characteristics. Even when given identical instructions, annotators interpret data differently depending on their personal context.

Subjective annotation tasks such as toxicity detection, moral judgment, or hate speech classification are particularly susceptible to this type of bias (Sap et al., 2019; Liu et al., 2022; Plank, 2022). Aggregation techniques like majority voting can obscure these differences and suppress minority perspectives, especially when annotator diversity is limited (Aroyo and Welty, 2015; Shardlow, 2022).

The rise of crowdwork has intensified these challenges. Annotator pools often differ demographically from both the dataset's source community and its intended application domain (Eickhoff, 2018; Bender et al., 2021). As a result, annotations may misinterpret cultural cues, dialectal language, or context-specific emotional tone. Although such variation is not necessarily the result of carelessness, it can introduce systematic distortion, especially when disagreement is treated as noise rather than signal (Cabitza et al., 2023).

3.3 Contextual and Cultural Bias

Contextual and cultural bias occurs when task design and labelling decisions assume a particular worldview, linguistic norm, or social context. It becomes especially pronounced in multilingual and multimodal tasks, where language, meaning, and affective signals vary widely across cultures.

Annotation labels such as "polite", "supportive", or "offensive" often fail to translate cleanly across languages or communities (Ponti et al., 2020). Cultural norms shape how people interpret both language and non-verbal cues, including gestures and tone of voice (Barrett et al., 2019; Lukac et al., 2023). Without regionally grounded interpretation frameworks, annotators may mislabel visual or emotional content.

Additionally, most pretraining data is skewed toward English and Western sources. As of 2025, English accounts for nearly half of all indexed web content (Ani Petrosyan, 2025). This imbalance in data collection reinforces a corresponding bias in annotation practices.

Recent work has emphasised the importance of culturally grounded taxonomies and community consultation for annotation tasks involving identity, emotion, or morality (Blodgett et al., 2020; Hutchinson et al., 2020; Zhou et al., 2023). Without such grounding, models trained on annotated data risk reproducing narrow, non-representative worldviews.

4 Impact on Model Behaviour

Bias introduced during annotation does not remain confined to the dataset. It propagates into the models trained on that data and leads to measurable downstream harms. This phenomenon, often referred to as "bias in, bias out," is a central concern in machine learning. When annotation processes reflect cultural, social, or demographic distortions, models tend to reproduce those distortions, and in some cases, amplify them (Dodge et al., 2021).

One of the most well-documented consequences

is performance disparity across demographic groups. A model may perform well on aggregate metrics while underperforming on texts associated with certain identities, dialects, or cultural contexts. For example, commercial gender classification systems have shown much higher error rates for darker-skinned women. This discrepancy can be traced, in part, to unbalanced training data that lacked diverse and properly annotated examples (Buolamwini and Gebru, 2018). Similarly, recidivism prediction tools have displayed racially skewed false positive rates due to historical biases embedded in the labelled data (Dressel and Farid, 2018).

Beyond accuracy gaps, annotation bias also causes representational harm. These occur when models learn to reproduce social stereotypes or unfair associations. For instance, if training labels disproportionately associate "engineer" with men and "nurse" with women, the model may internalise and repeat these biases in downstream tasks such as text generation or summarisation (Sheng et al., 2019). In a similar way, toxicity detection models trained on biased annotations may misclassify expressions in African-American Vernacular English (AAVE) as hostile or inappropriate (Sap et al., 2019).

These harms can be formalised using established fairness metrics. **Demographic Parity** requires that the rate of positive predictions be equal across groups. **Equalised Odds** requires that true and false positive rates remain consistent regardless of group membership. Annotation bias undermines these goals. Returning to the AAVE example, if annotators are more likely to label AAVE expressions as toxic, a classifier trained on such data will exhibit a higher false positive rate for Black speakers. This violates Equalised Odds and leads to unfair penalties against specific communities (Dixon et al., 2018).

These examples demonstrate that annotation bias is not a peripheral issue. It directly contributes to systemic failures in LLMs that affect both technical performance and social impact. For this reason, examining and improving annotation practices is a foundational step toward fairer and more reliable NLP systems.

5 Case Studies and Empirical Evidence

To illustrate how annotation bias operates in practice, this section presents two case studies. The first addresses multilingual hate speech detection, where cultural definitions of offence lead to misalignment between training data and deployment contexts. The second focuses on multimodal emotion recognition, where non-verbal cues are interpreted differently across cultural frameworks. These cases highlight that bias often arises not from individual prejudice but from structural mismatches between annotation design and communicative diversity.

5.1 Case Study: Cross-Cultural Hate Speech Detection

Hate speech detection is highly sensitive to cultural context. What is considered offensive or harmful in one setting may be acceptable or even humorous in another. This presents a serious challenge for creating models intended to generalise across regions and languages.

Lee et al. (2023) evaluated monolingual hate speech classifiers across cultural contexts by applying models trained on English-language data from the United States to translated data from languages such as Korean and Arabic. The results showed a drop in F1 scores of up to 42% and a fourfold increase in false negatives. These failures stemmed not from technical flaws in the models themselves but from annotation biases embedded in the source datasets. Several factors contributed to this performance collapse:

- Cultural targets vary. Groups and individuals who are frequent targets of hate speech differ between cultures, meaning training data from one country may miss important examples from another.
- Sociocultural norms shape expression. Sarcasm, irony, and rhetorical devices have different meanings and social functions depending on the culture.
- Standards of offensiveness diverge. A statement considered hateful in one community may be seen as neutral or even acceptable in another, depending on social, political, or historical context.

This case demonstrates that hate speech is not a culturally neutral construct. Models built on datasets annotated within a single cultural context may fail when applied elsewhere, even if the language is translated. This failure is not only a limitation of model generalisation but also a direct consequence of annotation bias in the original data.

5.2 Case Study: Multimodal Emotion Recognition

Bias in multimodal datasets can be more difficult to detect but equally damaging. Emotion recognition tasks that use audio, video, or gesture data rely on the interpretation of non-verbal cues, which are deeply culturally embedded.

Gunes and Piccardi (2007) conducted a study where they investigated how physical gestures were interpreted across cultural contexts. They found that a single gesture could signal patience in Egypt, positivity in Greece, and confrontation in Italy. When such data are annotated by individuals unfamiliar with the cultural origin of the gesture, systematic mislabelling is likely.

Cultural variation also affects emoji and facial expression interpretation. Gao and VanderLaan (2020) showed that annotators from Western cultures rely more on mouth shapes to read emoji emotions, while those from Eastern cultures prioritise the eyes. These perceptual differences result in inconsistent annotations and affect model training when emojis are used as supervision signals.

These findings underscore the importance of culturally grounded annotation frameworks in multimodal NLP. Without them, datasets risk encoding a narrow view of human emotion and interaction, reducing the validity and generalisability of trained models.

6 Detecting Annotation Bias

Detecting annotation bias is a crucial step toward mitigating its impact on model training and evaluation. A variety of methods have been proposed to identify systematic patterns of bias in annotated datasets, each with different strengths and limitations. One common approach is to measure inter-annotator agreement (IAA), using metrics such as Cohen's κ (Smeeton, 1985). For N instances and M annotators, where $y_i^{(j)} \in \mathcal{Y}$ is annotator j's label for instance i, the agreement is defined as:

$$\kappa_{\text{Cohen}} = \frac{p_o - p_e}{1 - p_e}, \quad p_o = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}(y_i^{(1)} = y_i^{(2)}) \tag{1}$$

Here p_o is the observed agreement, i.e., the probability that both annotators assign the *same* label to a randomly selected item. $\mathbb{I}(\cdot)$ is the indicator function. The expected agreement by chance p_e is computed as:

$$p_e = \sum_{k \in \mathcal{Y}} P(y^{(1)} = k) \cdot P(y^{(2)} = k) \tag{2}$$

where $P(y^{(j)} = k)$ is the empirical probability of annotator j assigning label k. Fleiss' κ generalises this metric for multiple annotators (Fleiss et al., 2013):

$$\kappa_{\text{Fleiss}} = \frac{\bar{p} - \bar{p}_e}{1 - \bar{p}_e}, \bar{p} = \frac{1}{N}\sum_{i=1}^{N} \frac{\sum_{k=1}^{K} n_{ik}(n_{ik} - 1)}{M(M-1)} \tag{3}$$

where n_{ik} is the number of annotators assigning label k to instance i.

For settings with missing data or mixed label types, Krippendorff's α (Krippendorff, 2011) offers a more general reliability metric:

$$\alpha = 1 - \frac{D_o}{D_e} \tag{4}$$

where D_o is the observed disagreement (weighted across annotator pairs per item) and D_e is the expected disagreement under chance.

A complementary approach is to analyse *model disagreement*. When two models are trained on the same data, divergence in their predictions can reveal annotation ambiguity or bias (Geva et al., 2019). For two models f_1 and f_2, the disagreement rate (DR) is defined as:

$$\text{DR} = \frac{1}{|\mathcal{D}|}\sum_{x \in \mathcal{D}} \mathbb{I}\big(f_1(x) \neq f_2(x)\big) \tag{5}$$

Uma et al. (2022) extend this idea by comparing model predictions with human labels:

$$\Delta = \frac{1}{|\mathcal{D}|}\sum_{x \in \mathcal{D}} \big|f(x) - y_{\text{human}}(x)\big| \tag{6}$$

This metric helps identify inconsistencies between model behaviour and annotation patterns (Dsouza and Kovatchev, 2025).

Another lens on bias detection comes from **metadata analysis**. By examining annotator demographics, task context, and label distributions, researchers can uncover systematic bias (Sap et al., 2019). For an annotator group a, a demographic gap $G(a)$ can be computed as:

$$G(a) = \left| \frac{1}{|\mathcal{D}_a|}\sum_{x \in \mathcal{D}_a} y(x) - \frac{1}{|\mathcal{D}|}\sum_{x \in \mathcal{D}} y(x) \right| \tag{7}$$

Here, $\mathcal{D}_a$ denotes the subset of data annotated by group a, and $y(x)$ is the label assigned to instance x. A high $G(a)$ may signal systematic differences in annotation patterns between group a and the overall dataset, potentially reflecting underlying biases or cultural variation.

This gap measures how far the group's average labels deviate from the global average, which may indicate bias or representational disparity (Sap et al., 2019). Traditional metrics, however, may be less effective in multilingual and culturally diverse settings. In these cases, disagreement may reflect true variation rather than annotation error (Naous et al., 2024). To address this, new strategies are emerging. *multilingual model disagreement* compares the predictions of models fine-tuned in different languages on parallel corpora $\mathcal{D}_{l_1,l_2}$:

$$\mathrm{DR}(l_1, l_2) = \frac{1}{|\mathcal{D}_{l_1,l_2}|} \sum_{x \in \mathcal{D}_{l_1,l_2}} \mathbb{I}\big(f_{l_1}(x) \neq f_{l_2}(x)\big) \tag{8}$$

where f_{l_1} and f_{l_2} denote the models fine-tuned in languages l_1 and l_2, respectively.

Similarly, cultural inference techniques (Zhang et al., 2020b; Huang and Yang, 2023) use embeddings or sociolinguistic metadata to detect alignment between annotations and cultural backgrounds. One such indicator, Φ_{cultural} is calculated as the ℓ_2 distance between two groups:

$$\Phi_{\text{cultural}} = \| \phi(\mathcal{D}_a) - \phi(\mathcal{D}_{a'}) \|_2 \tag{9}$$

where $\phi(\cdot)$ maps a dataset to its cultural embedding space, and $\mathcal{D}_a$, $\mathcal{D}_{a'}$ denote datasets annotated by cultural groups a and a', respectively.

Together, these methods offer a toolkit for identifying annotation bias at different levels: label consistency, annotator disagreement, cultural framing, and model interpretation. In practice, combining quantitative metrics with qualitative analysis offers the best chance of uncovering and addressing complex forms of annotation bias.

7 Mitigation Strategies

Detecting annotation bias is only the first step toward creating fair and reliable NLP systems. Effective mitigation requires both proactive strategies, which aim to prevent bias during data collection, and reactive strategies, which address it after annotation or model training. This section outlines techniques across both categories, integrating recent formal approaches with practical best practices.

7.1 Proactive Strategies

Proactive strategies aim to reduce annotation bias at the source by redesigning annotation processes with awareness of potential pitfalls.

Diverse Annotator Pools To counter annotator bias, it is essential to recruit annotators from a broad range of demographic, cultural, and linguistic backgrounds (Bender et al., 2021; Paullada et al., 2021). A diverse pool can reveal meaningful disagreements and represent underreported perspectives (Aroyo and Welty, 2015). One way to quantify diversity is through the entropy of the demographic distribution:

$$H(A) = -\sum_{a \in \mathcal{A}} p(a) \log p(a) \tag{10}$$

where $\mathcal{A}$ is the set of annotator groups and $p(a)$ is the proportion of annotations from group a. A higher entropy score $H(A)$ indicates a more balanced and inclusive annotation pool.

Dynamic Annotation Guidelines To mitigate instruction bias, guidelines and prompts should be piloted, reviewed and refined iteratively. This feedback loop helps remove culturally specific assumptions and linguistic ambiguities (Parmar et al., 2023). In LLM-based settings, prompt engineering should be evaluated across cultural contexts to ensure validity (Zamfirescu-Pereira et al., 2023). One can formalise this iterative process by tracking the variance in annotator disagreement across iterations:

$$\sigma_t^2 = \frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} \mathrm{Var}\big(\{y_i^{(t)}(x)\}_{i=1}^n\big) \tag{11}$$

where $y_i^{(t)}(x)$ is the label from annotator i on item x during iteration t, with the goal that $\sigma_t^2 \to \min$ over t.

Culturally Grounded Taxonomies To address contextual and cultural bias, annotation schemes should be developed with culturally grounded taxonomies of emotion, politeness, morality, and related constructs (Blodgett et al., 2020; Hutchinson et al., 2020; Zhou et al., 2023). Engaging with communities or domain experts helps ensure that annotation labels are valid across languages and cultural settings (Ponti et al., 2020; Naous et al., 2024).

7.2 Reactive Strategies

Reactive strategies are applied after biases have entered the data or the model. They aim to mitigate downstream harms without necessarily revising the annotation process itself. One key challenge in post-hoc mitigation is handling inconsistencies introduced by annotator subjectivity and instruction bias, particularly when labels reflect divergent interpretations of subjective or culturally loaded concepts. Weighted ensemble methods can address this by leveraging multiple model perspectives to smooth over annotation noise, while still preserving minority viewpoints.

Post-hoc Model Adjustment Biases in trained models can sometimes be mitigated using post-hoc correction methods such as embedding debiasing or output regularisation. Kaneko et al. (2023) proposed modifying model outputs by subtracting a learned bias component:

$$f^{\text{debias}}(x) = f_\theta(x) - \lambda b(x) \tag{12}$$

where $f_\theta(x)$ is the original model output, $b(x)$ is a bias projection and λ controls debiasing strength.

Fine-tuning and In-context Debiasing Recent work has explored using targeted fine-tuning (Webster et al., 2021) or in-context prompting (Ganguli et al., 2023) to reshape model behaviour without altering the training data (Kaneko et al., 2025). In the fine-tuning case, model parameters θ are updated using reweighted or re-annotated dataset $\mathcal{D}^*$:

$$\min_\theta \mathbb{E}_{(x,y^*)\sim\mathcal{D}^*} \mathcal{L}(f_\theta(x), y^*) \tag{13}$$

In in-context learning, models are conditioned on carefully constructed prompts P that reduce bias:

$$f_\theta(y \mid x, P) \tag{14}$$

where P is designed to reduce the likelihood of biased completions while preserving task accuracy.

Multi-Objective Weighted Ensemble Learning Another reactive strategy leverages ensemble learning to mitigate annotation bias by explicitly modelling annotator disagreement (Geva et al., 2019). Given a dataset $\mathcal{D} = \{(x_i, \{y_i^{(j)}\}_{j=1}^M)\}_{i=1}^N$, Huang et al. (2025) proposed Weak Ensemble Learning (WEL), which samples one annotator label per instance to construct K label-variant datasets. Each trains a weak predictor f_{θ_k}, weighted by its held-out performance (e.g., F1, cross-entropy, Manhattan distance), with $\sum_{k=1}^K w_k = 1$. We extend WEL to a multilingual setting by applying the same label-sampling procedure across datasets in different languages using a shared multilingual model. Final predictions are computed as:

$$\hat{y}_i = \sum_{k=1}^{K} w_k \, f_{\theta_k}(x_i), \tag{15}$$

allowing the ensemble to capture annotator disagreement while leveraging multilingual representations from a single model.

We use `mBERT` (Devlin et al., 2019) as the base model. On the multi-source benchmark from the LeWiDi 2023 shared task (Leonardellli et al., 2023), WEL generally outperforms baselines using single-model CE loss (CE-only) (Uma et al., 2020) and majority-vote ensembles of top five annotators (Top-5-Ann) (Xu et al., 2024), achieving higher F1 and lower CE/MD scores. The only exception is *ArMIS*, where the very small annotator pool (three annotators) limits the effectiveness of random label sampling. As the primary focus of this paper is on the discussion of annotation bias in multilingual LLMs, we include the full experimental results in Appendix B.

8 Ethical and Practical Considerations

The discussion of annotation bias is incomplete without considering the ethical and practical realities of the annotation process itself. Creating high-quality labelled data is not only a technical challenge but also a form of labour that carries human and institutional consequences. These concerns are directly tied to the emergence and persistence of annotation bias because they influence how data are produced, who produces it, and under what conditions.

8.1 Annotator Wellbeing and Psychological Safety

One of the most pressing concerns involves the well-being of annotators, particularly those responsible for labelling harmful, toxic, or distressing content. Content moderation datasets, which are essential for training safety filters in LLMs, often expose annotators to a continuous stream of violent, hateful, or traumatic material. Research shows that prolonged exposure to such content can lead to severe psychological effects, including anxiety, depression, insomnia, and symptoms of post-traumatic stress disorder (PTSD) (Das et al., 2020).

This phenomenon is referred to as *vicarious trauma*, a condition in which individuals who are indirectly exposed to trauma begin to show symptoms similar to those of direct trauma survivors (Pearlman and Saakvitne, 1995). These effects are compounded by stressful work environments. Annotators often face tight deadlines and high task volumes, with limited autonomy or support systems (Spence et al., 2023). In many cases, stigma around mental health further prevents them from seeking help (Bergman and Rushton, 2023).

To mitigate these harms, researchers and data curators have a responsibility to implement safeguards. These may include access to mental health services, task rotation to reduce exposure to distressing material, and policies that allow annotators to opt out of specific assignments. Regular breaks, content warnings, and workplace cultures that promote psychological safety are also important steps toward ethical annotation pipelines (Spence et al., 2023).

8.2 Power Dynamics in Data Labour

Annotation work is often conducted through crowdworking platforms that rely on a globally distributed, low-cost labour force. These platforms are sometimes described as democratising access to work, but they often reflect significant power asymmetries between requesters and workers. Annotators frequently operate as anonymous contractors with no job security, limited bargaining power, and little visibility into how their work is used (Roberts, 2016). Compensation is usually task-based, which creates incentives to prioritise speed over accuracy.

This trade-off can result in lower-quality labels and increase the risk of bias in the final dataset (Snow et al., 2008b). Additionally, annotators rarely have channels for providing feedback about unclear instructions, ambiguous data, or annotation policies. As a result, a valuable feedback loop for improving annotation guidelines is often lost (Miceli and Posada, 2022).

These structural imbalances are not only ethical concerns; they also have technical implications. Poor working conditions can degrade data quality, obscure disagreement patterns, and exclude minority perspectives (Snow et al., 2008b). Creating fairer and more collaborative annotation systems, where annotators are treated as skilled contributors instead of disposable labour, can help ensure both ethical integrity and model reliability.

Ethical considerations must not be separated from methodological concerns. The conditions under which data are created shape their reliability, fairness, and downstream utility. Addressing annotation bias requires attention not only to technical design, but also to the social and economic contexts in which annotation work occurs.

9 Conclusion and Future Directions

Annotation bias remains a central challenge for multilingual and multimodal LLMs, shaping how models learn, generalise, and interact with diverse users. Mitigation requires both proactive measures (e.g., diverse annotators, refined guidelines) and reactive tools (e.g., bias detection, post-hoc adjustment), underpinned by ethical commitments to annotator well-being and fair labour.

Future work should prioritise community-driven annotation in marginalised contexts, culturally grounded benchmarks, and richer annotator metadata to improve fairness diagnostics, particularly in low-resource settings. LLMs themselves can assist as scalable annotation and bias-detection tools, but must be guided by real-world social and cultural contexts.

This paper contributes a typology of annotation bias, surveys detection methods across multilingual and cultural settings, and outlines mitigation strategies. We extend an ensemble-based method to multilingual settings to address label noise and inter-annotator disagreement, demonstrating its effectiveness on four socially sensitive tasks. Incorporating cultural awareness and accountability throughout the data pipeline will help NLP systems better reflect the diversity of human communication.

Acknowledgments

We thank the anonymous reviewers and program chairs for their insightful comments and constructive suggestions.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Ani Petrosyan. 2025. Most used languages online by share of websites 2025.

Lora Aroyo and Chris Welty. 2015. Truth is a lie: Crowd truth and the seven myths of human annotation. *AI Mag.*, 36(1):15–24.

Collin F Baker, Charles J Fillmore, and John B Lowe. 1998. The berkeley framenet project. In *COLING 1998 Volume 1: The 17th International Conference on Computational Linguistics*.

Lisa Feldman Barrett, Ralph Adolphs, Stacy Marsella, Aleix M. Martinez, and Seth D. Pollak. 2019. Emotional expressions reconsidered: Challenges to inferring emotion from human facial movements. *Psychological Science in the Public Interest*, 20(1):1–68. PMID: 31313636.

Emily M. Bender and Batya Friedman. 2018. Data statements for natural language processing: Toward mitigating system bias and enabling better science. *Transactions of the Association for Computational Linguistics*, 6:587–604.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 610–623, New York, NY, USA. Association for Computing Machinery.

Alanna Bergman and Cynda Hylton Rushton. 2023. Overcoming stigma: Asking for and receiving mental health support. *AACN advanced critical care*, 34(1):67–71.

Su Lin Blodgett, Solon Barocas, Hal Daumé III, and Hanna Wallach. 2020. Language (technology) is power: A critical survey of "bias" in NLP. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5454–5476, Online. Association for Computational Linguistics.

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

Joy Buolamwini and Timnit Gebru. 2018. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*, volume 81 of *Proceedings of Machine Learning Research*, pages 77–91. PMLR.

Federico Cabitza, Andrea Campagner, and Valerio Basile. 2023. Toward a perspectivist turn in ground truthing for predictive computing. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(6):6860–6868.

Xia Cui, Terry Hanley, Muj Choudhury, and Tingting Mu. 2024. Data-driven or dataless? detecting indicators of mental health difficulties and negative life events in financial resilience using prompt-based learning. In *2024 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

Anubrata Das, Brandon Dang, and Matthew Lease. 2020. Fast, accurate, and healthier: Interactive blurring helps moderators reduce exposure to harmful content. In *Proceedings of the AAAI conference on human computation and crowdsourcing*, volume 8, pages 33–42.

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.

Lucas Dixon, John Li, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. 2018. Measuring and mitigating unintended bias in text classification. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '18, page 67–73, New York, NY, USA. Association for Computing Machinery.

Jesse Dodge, Maarten Sap, Ana Marasović, William Agnew, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell, and Matt Gardner. 2021. Documenting large webtext corpora: A case study on the colossal clean crawled corpus. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1286–1305, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Julia Dressel and Hany Farid. 2018. The accuracy, fairness, and limits of predicting recidivism. *Science Advances*, 4(1):eaao5580.

Russel Dsouza and Venelin Kovatchev. 2025. Sources of disagreement in data for LLM instruction tuning. In *Proceedings of Context and Meaning: Navigating Disagreements in NLP Annotation*, pages 20–32, Abu Dhabi, UAE. International Committee on Computational Linguistics.

Abteen Ebrahimi, Manuel Mager, Arturo Oncevay, Vishrav Chaudhary, Luis Chiruzzo, Angela Fan, John Ortega, Ricardo Ramos, Annette Rios, Ivan Vladimir Meza Ruiz, Gustavo Giménez-Lugo, Elisabeth Mager, Graham Neubig, Alexis Palmer, Rolando Coto-Solano, Thang Vu, and Katharina Kann. 2022. AmericasNLI: Evaluating zero-shot natural language understanding of pretrained multilingual models in truly low-resource languages. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6279–6299, Dublin, Ireland. Association for Computational Linguistics.

Carsten Eickhoff. 2018. Cognitive biases in crowdsourcing. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, WSDM '18, page 162–170, New York, NY, USA. Association for Computing Machinery.

Joseph L Fleiss, Bruce Levin, and Myunghee Cho Paik. 2013. *Statistical methods for rates and proportions*. john wiley & sons.

Deep Ganguli, Amanda Askell, Nicholas Schiefer, Thomas I. Liao, Kamilė Lukošiūtė, Anna Chen, Anna Goldie, Azalia Mirhoseini, Catherine Olsson, Danny Hernandez, Dawn Drain, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jackson Kernion, Jamie Kerr, Jared Mueller, Joshua Landau, Kamal Ndousse, Karina Nguyen, Liane Lovitt, Michael Sellitto, Nelson Elhage, Noemi Mercado, Nova DasSarma, Oliver Rausch, Robert Lasenby, Robin Larson, Sam Ringer, Sandipan Kundu, Saurav Kadavath, Scott Johnston, Shauna Kravec, Sheer El Showk, Tamera Lanham, Timothy Telleen-Lawton, Tom Henighan, Tristan Hume, Yuntao Bai, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, Christopher Olah, Jack Clark, Samuel R. Bowman, and Jared Kaplan. 2023. The capacity for moral self-correction in large language models.

Boting Gao and Doug P VanderLaan. 2020. Cultural influences on perceptions of emotions depicted in emojis. *Cyberpsychology, Behavior, and Social Networking*, 23(8):567–570.

Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. 2021. Datasheets for datasets. *Commun. ACM*, 64(12):86–92.

Mor Geva, Yoav Goldberg, and Jonathan Berant. 2019. Are we modeling the task or the annotator? an investigation of annotator bias in natural language understanding datasets. 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 1161–1166, Hong Kong, China. Association for Computational Linguistics.

Hatice Gunes and Massimo Piccardi. 2007. Bi-modal emotion recognition from expressive face and body gestures. *Journal of Network and Computer Applications*, 30(4):1334–1345.

Kang He, Yinghan Long, and Kaushik Roy. 2024. Prompt-based bias calibration for better zero/few-shot learning of language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 12673–12691, Miami, Florida, USA. Association for Computational Linguistics.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. Xtreme: a massively multilingual multi-task benchmark for evaluating cross-lingual generalization. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org.

Jing Huang and Diyi Yang. 2023. Culturally aware natural language inference. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7591–7609, Singapore. Association for Computational Linguistics.

Ziyi Huang, Nishanthi Rupika Abeynayake, and Xia Cui. 2025. Weak ensemble learning from multiple annotators for subjective text classification. In *Proceedings of the 4th Workshop on Perspectivist Approaches to NLP (NLPerspectives) @ EMNLP 2025*, Suzhou, China. Association for Computational Linguistics.

Ben Hutchinson, Vinodkumar Prabhakaran, Emily Denton, Kellie Webster, Yu Zhong, and Stephen Denuyl. 2020. Social biases in NLP models as barriers for persons with disabilities. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5491–5501, Online. Association for Computational Linguistics.

Masahiro Kaneko, Danushka Bollegala, and Timothy Baldwin. 2025. The gaps between fine tuning and in-context learning in bias evaluation and debiasing. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 2758–2764, Abu Dhabi, UAE. Association for Computational Linguistics.

Masahiro Kaneko, Danushka Bollegala, and Naoaki Okazaki. 2023. The impact of debiasing on the performance of language models in downstream tasks is underestimated. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 29–36, Nusa Dua, Bali. Association for Computational Linguistics.

Klaus Krippendorff. 2011. Computing krippendorff's alpha-reliability.

Nayeon Lee, Chani Jung, and Alice Oh. 2023. Hate speech classifiers are culturally insensitive. In *Proceedings of the First Workshop on Cross-Cultural Considerations in NLP (C3NLP)*, pages 35–46, Dubrovnik, Croatia. Association for Computational Linguistics.

Elisa Leonardellli, Gavin Abercrombie, Dina Almanea, Valerio Basile, Tommaso Fornaciari, Barbara Plank, Massimo Poesio, Verena Rieser, and Alexandra Uma. 2023. SemEval-2023 Task 11: Learning With Disagreements (LeWiDi). In *Proceedings of the 17th International Workshop on Semantic Evaluation*, Toronto, Canada. Association for Computational Linguistics.

Haochen Liu, Joseph Thekinen, Sinem Mollaoglu, Da Tang, Ji Yang, Youlong Cheng, Hui Liu, and Jiliang Tang. 2022. Toward annotator group bias in crowdsourcing. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1797–1806,

Dublin, Ireland. Association for Computational Linguistics.

Martin Lukac, Gulnaz Zhambulova, Kamila Abdiyeva, and Michael Lewis. 2023. Study on emotion recognition bias in different regional groups. *Scientific Reports*, 13(1):8414.

Mitch Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330.

Milagros Miceli and Julian Posada. 2022. The data-production dispositif. *Proceedings of the ACM on human-computer interaction*, 6(CSCW2):1–37.

Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. 2019. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Saif M. Mohammad. 2016. 9 - sentiment analysis: Detecting valence, emotions, and other affectual states from text. In Herbert L. Meiselman, editor, *Emotion Measurement*, pages 201–237. Woodhead Publishing.

Robert Munro Monarch. 2021. *Human-in-the-Loop Machine Learning: Active learning and annotation for human-centered AI*. Simon and Schuster.

Tarek Naous, Michael J Ryan, Alan Ritter, and Wei Xu. 2024. Having beer after prayer? measuring cultural bias in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16366–16393, Bangkok, Thailand. Association for Computational Linguistics.

Mihir Parmar, Swaroop Mishra, Mor Geva, and Chitta Baral. 2023. Don't blame the annotator: Bias already starts in the annotation instructions. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1779–1789, Dubrovnik, Croatia. Association for Computational Linguistics.

Amandalynne Paullada, Inioluwa Deborah Raji, Emily M. Bender, Emily Denton, and Alex Hanna. 2021. Data and its (dis)contents: A survey of dataset development and use in machine learning research. *Patterns*, 2(11):100336.

Siddhesh Pawar, Junyeong Park, Jiho Jin, Arnav Arora, Junho Myung, Srishti Yadav, Faiz Ghifari Haznitrama, Inhwa Song, Alice Oh, and Isabelle Augenstein. 2025. Survey of cultural awareness in language models: Text and beyond. *Computational Linguistics*, pages 1–96.

Laurie Anne Pearlman and Karen W Saakvitne. 1995. *Trauma and the therapist: Countertransference and vicarious traumatization in psychotherapy with incest survivors.* WW Norton & Company.

Barbara Plank. 2022. The "problem" of human label variation: On ground truth in data, modeling and evaluation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Abu Dhabi. Association for Computational Linguistics.

Edoardo Maria Ponti, Goran Glavaš, Olga Majewska, Qianchu Liu, Ivan Vulić, and Anna Korhonen. 2020. XCOPA: A multilingual dataset for causal commonsense reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2362–2376, Online. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Sarah T. Roberts. 2016. *The Intersectional Internet: Race, Sex, Class and Culture Online*, chapter Commercial content moderation: Digital labourers'dirty work. Peter Lang Publishing.

Maarten Sap, Dallas Card, Saadia Gabriel, Yejin Choi, and Noah A. Smith. 2019. The risk of racial bias in hate speech detection. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1668–1678, Florence, Italy. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2022. True few-shot learning with Prompts—A real-world perspective. *Transactions of the Association for Computational Linguistics*, 10:716–731.

Matthew Shardlow. 2022. Agree to disagree: Exploring subjectivity in lexical complexity. In *Proceedings of the 2nd Workshop on Tools and Resources to Empower People with REAding DIfficulties (READI) within the 13th Language Resources and Evaluation Conference*, pages 9–16, Marseille, France. European Language Resources Association.

Emily Sheng, Kai-Wei Chang, Premkumar Natarajan, and Nanyun Peng. 2019. The woman worked as a babysitter: On biases in language generation. 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 3407–3412, Hong Kong, China. Association for Computational Linguistics.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. 2019. Towards vqa models that can read. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Nigel C Smeeton. 1985. Early history of the kappa statistic. *Biometrics*, 41:795.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Ng. 2008a. Cheap and fast – but is it good? evaluating non-expert annotations for natural language tasks. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 254–263, Honolulu, Hawaii. Association for Computational Linguistics.

Rion Snow, Brendan O'connor, Dan Jurafsky, and Andrew Y Ng. 2008b. Cheap and fast–but is it good? evaluating non-expert annotations for natural language tasks. In *Proceedings of the 2008 conference on empirical methods in natural language processing*, pages 254–263.

Ruth Spence, Antonia Bifulco, Paula Bradbury, Elena Martellozzo, and Jeffrey DeMarco. 2023. The psychological impacts of content moderation on content moderators: A qualitative study. *Cyberpsychology: Journal of Psychosocial Research on Cyberspace*, 17(4).

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Alexandra Uma, Tommaso Fornaciari, Dirk Hovy, Silviu Paun, Barbara Plank, and Massimo Poesio. 2020. A case for soft loss functions. *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, 8(1):173–177.

Alexandra N. Uma, Tommaso Fornaciari, Dirk Hovy, Silviu Paun, Barbara Plank, and Massimo Poesio. 2022. Learning from disagreement: A survey. volume 72, page 1385–1470, El Segundo, CA, USA. AI Access Foundation.

Kellie Webster, Xuezhi Wang, Ian Tenney, Alex Beutel, Emily Pitler, Ellie Pavlick, Jilin Chen, Ed Chi, and Slav Petrov. 2021. Measuring and reducing gendered correlations in pre-trained models.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. Finetuned language models are zero-shot learners. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

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

Jin Xu, Mariët Theune, and Daniel Braun. 2024. Leveraging annotator disagreement for text classification. In *Proceedings of the 7th International Conference on Natural Language and Speech Processing (ICNLSP 2024)*, pages 1–10, Trento. Association for Computational Linguistics.

J Diego Zamfirescu-Pereira, Richmond Y Wong, Bjoern Hartmann, and Qian Yang. 2023. Why johnny can't prompt: how non-ai experts try (and fail) to design llm prompts. In *Proceedings of the 2023 CHI conference on human factors in computing systems*, pages 1–21.

Guanhua Zhang, Bing Bai, Junqi Zhang, Kun Bai, Conghui Zhu, and Tiejun Zhao. 2020a. Demographics should not be the reason of toxicity: Mitigating discrimination in text classifications with instance weighting. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4134–4145, Online. Association for Computational Linguistics.

Haiping Zhang, Xingxing Zhou, Guoan Tang, Genlin Ji, Xueying Zhang, and Liyang Xiong. 2020b. Inference method for cultural diffusion patterns using a field model. *Transactions in GIS*, 24(6):1578–1601.

Tony Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *International Conference on Machine Learning*.

Yi Zhou, Jose Camacho-Collados, and Danushka Bollegala. 2023. A predictive factor analysis of social biases and task-performance in pretrained masked language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11082–11100, Singapore. Association for Computational Linguistics.

Supplementary Materials

A Relationships Among Bias Types, Detection, and Mitigation

This supplementary section details the observed relationships between annotation bias types and their associated detection and mitigation approaches, as illustrated in Figures 1 and 2. These relationships emerged from our analysis of current literature and empirical findings, rather than constituting a prescriptive framework.

A.1 Relationships Between Bias Types

Our analysis identifies three primary bias types that frequently interact in annotation processes: (1) **Instruction Bias**: Arising from task design choices, guidelines, and prompt formulations; (2) **Annotator Bias**: Stemming from individual predispositions and demographic characteristics; (3) **Contextual & Cultural Bias**: Emerging from cultural mismatches and linguistic norms. These bias types often co-occur and compound each other, particularly in multilingual settings where cultural context influences both task interpretation and annotation execution.

A.2 Correlations Between Detection and Mitigation Approaches

Figure 2 illustrates correlations observed between specific bias types and effective handling strategies.

Instruction bias correlations include detection through inter-annotator agreement metrics (Krippendorff, 2011) and model disagreement analysis (Geva et al., 2019), with mitigation through guideline refinement (Parmar et al., 2023) and in-context debiasing (Ganguli et al., 2023).

Annotator bias correlations involve detection via metadata analysis (Sap et al., 2019) with mitigation through diverse annotator recruitment (Bender and Friedman, 2018) and weak ensemble learning (Huang et al., 2025).

Contextual and cultural bias correlations include detection via multilingual divergence analysis (Huang and Yang, 2023) and cultural inference methods (Zhang et al., 2020a), with mitigation through culturally grounded taxonomies (Ponti et al., 2020) and post-hoc adjustments (Kaneko et al., 2023).

A.3 Emergent Cross-Connections

Our analysis reveals several emergent cross-connections where detection methods inform mitigation strategies: (1) Inter-annotator disagreement metrics often correlate with both instruction and annotator bias, suggesting applications for ensemble-based mitigation; (2) Cultural inference methods show relationships with both bias detection and the development of culturally-aware taxonomies; (3) Metadata analysis frequently informs both bias identification and targeted mitigation through annotator diversity initiatives. These relationships suggest that effective bias handling may benefit from considering detection methods not only as diagnostic tools but also as informants for mitigation strategy selection. However, these correlations should be interpreted as observed relationships rather than definitive prescriptions, as contextual factors may alter their applicability in specific settings.

B Benchmark Comparison using WEL on Multilingual LLMs

B.1 Data

We assess Weak Ensemble Learning (WEL) on the LeWiDi 2023 shared task datasets (Leonardelli et al., 2023), which are designed to evaluate generalisation across languages and domains. The benchmark consists of four corpora that vary in language, genre, and annotation protocol.

Three corpora (*ArMIS*, *HS-Brexit*, *MD-Agreement*) contain social media posts from *X*[1]. **ArMIS** comprises Arabic posts annotated for misogyny. **HS-Brexit** contains English posts labelled for Brexit-related hate speech. **MD-Agreement** consists of English posts annotated for offensiveness across multiple domains (e.g., BLM, elections, COVID-19); we disregard domain metadata and treat them uniformly.

The fourth corpus, *ConvAbuse*, contains English dialogues between users and conversational agents. Utterances are rated on a 5-point abuse scale ranging from -3 (highly abusive) to 1 (non-abusive). Following prior work, we reduce this to a binary classification task: *offensive* (< 0) vs. *non-offensive* (≥ 0). Multi-turn dialogues are flattened into single text sequences.

All datasets undergo standard preprocessing, including the removal of HTML tags, URLs, user mentions, punctuation, digits, non-ASCII characters, and redundant whitespace. Table 1 provides a summary of dataset statistics, including split sizes, annotator ranges, and total annotator counts.

[1] https://x.com/

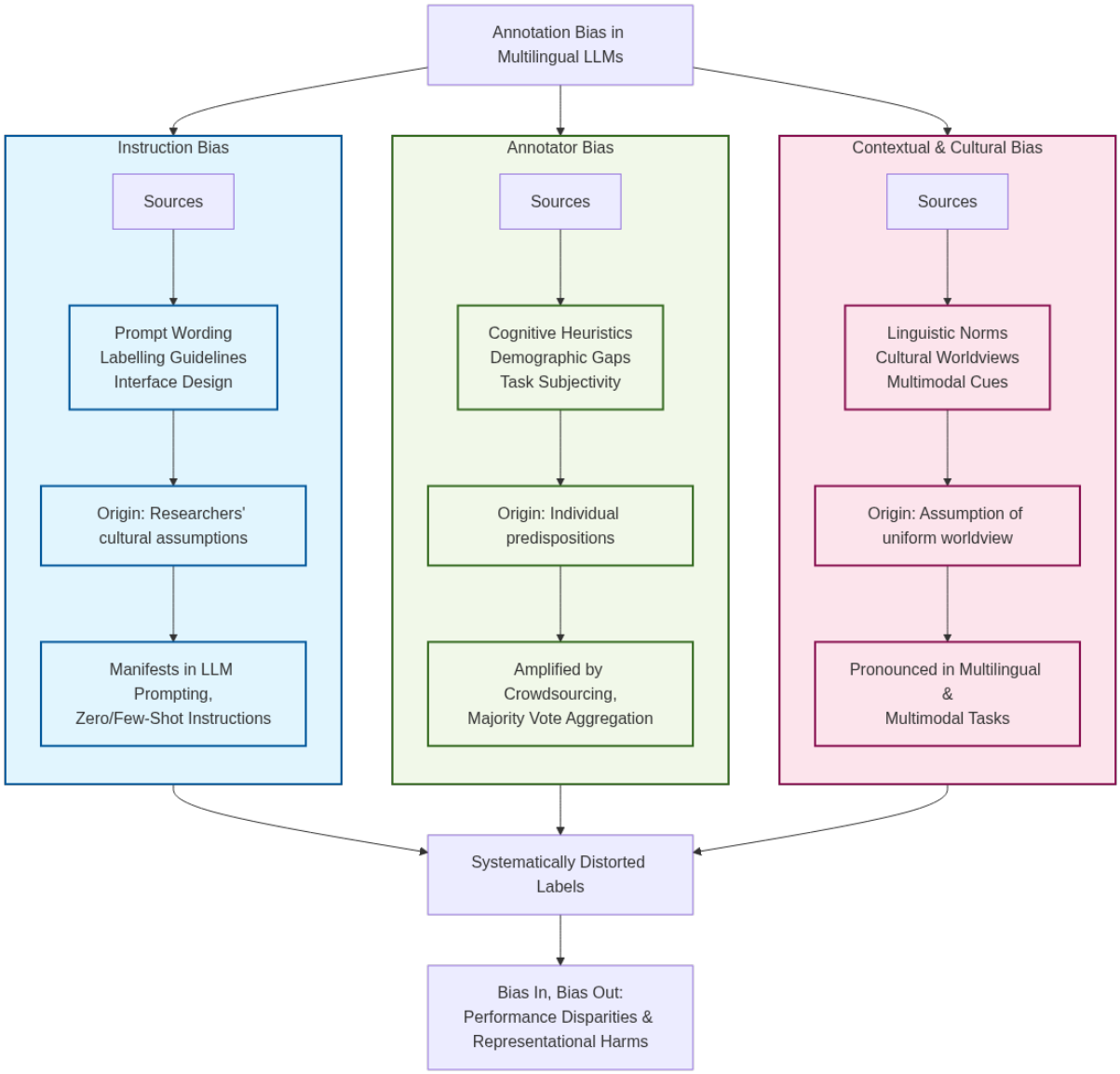

Figure 1: Taxonomy of annotation bias types observed in multilingual LLMs, showing three primary categories with distinct colouring: Instruction Bias, Annotator Bias, and Contextual & Cultural Bias.

B.2 Base LLM

To enable cross-linguistic generalisation in our ensemble-based framework, we adopt the multilingual BERT (`mBERT`) model (Devlin et al., 2019), more specifically, `bert-base-multilingual-uncased`[2], as the shared encoder for all weak learners in the WEL framework. This transformer-based model is pre-trained on 104 languages using a masked language modelling objective and retains casing information, making it well-suited for tasks with mixed scripts and morphologically rich languages.

In our setup, each weak predictor f_{θ_k} in the ensemble is instantiated by fine-tuning a separate copy of the multilingual BERT model on a label-variant dataset constructed via random sampling from annotator labels (as described in Section 7). Despite training on datasets in different languages and domains, all predictors share the same multilingual backbone, allowing for consistent representation across languages while preserving the benefits of ensemble diversity. This choice enables us to evaluate the robustness of WEL in a multilingual, multi-dataset context without requiring separate architectures per language.

[2] https://huggingface.co/google-bert/bert-base-multilingual-uncased

Table 1: Data statistics and metadata for the four textual datasets. #Train, #Dev, and #Test denote the number of instances in the training, development, and test splits. #TotalAnn indicates the total number of annotators, while #Ann represents the minimum and maximum number of annotators per instance. Contribution, Diversity, Language, and Genre provide further dataset details.

Dataset	#Train	#Dev	#Test	#TotalAnn	#Ann	Contribution	Diversity	Language	Genre
ArMIS	657	141	145	3	3	Fixed Annotators	Low	Arabic	Short Text
ConvAbuse	2398	812	840	8	2-7	Mixed Annotators	Low	English	Conversation
HS-Brexit	784	168	168	6	6	Fixed Annotators	Low	English	Short Text
MD-Agreement	6592	1104	3057	670	5	Mixed Annotators	High	English	Short Text

B.3 Results

Table 2 compares three models (CE-only (Uma et al., 2020), Top-5-Annotators (Xu et al., 2024), and WEL) across four datasets using F1 (higher better), cross-entropy (CE), and Manhattan distance (MD) (lower better). We perform a grid search over loss term coefficients in the objective function, each sampled from the range $[0, 0.001, 0.01, 0.1, 1]$, resulting in 1,295 unique combinations per dataset (excluding 0s for all). WEL consistently achieved the highest F1 scores and best calibration metrics (CE/MD) across three of four datasets, demonstrating its robustness for uncertainty-aware NLP.

Key observations emerge: (1) Performance varies substantially by domain, with *ConvAbuse* showing highest F1 scores but also extreme MD values for CE-only (4.81 vs. $<$1.0 elsewhere), indicating prediction instability that WEL addresses; (2) WEL's advantage in calibration metrics (CE/MD) exceeds its F1 improvements, highlighting its particular strength in uncertainty estimation; (3) Statistically significant improvements ($p < 0.05$) on *HS-Brexit* and *MD-Agreement* demonstrate WEL's robustness for hate speech and agreement tasks; (4) The *ArMIS* exception, where minimal gains occurred with only three annotators, establishes a practical boundary condition: WEL requires sufficient annotator diversity (likely $>$3) for effective ensemble learning. These results position WEL as particularly valuable for applications requiring reliable confidence estimates, while clearly defining its limitations in low-diversity annotation settings.

Table 2: Performance comparison across datasets and models. Best values are highlighted (F1: higher better; CE/MD: lower better). * indicates $p < 0.05$ significance.

Dataset	Metric	CE-only	Top-5-Ann	**WEL**
ArMIS	F1	0.6482	0.6552	0.6483
	CE	0.7019	0.6502	0.6596
	MD	0.7001	0.6443	0.6609
ConvAbuse	F1	0.8362	0.9310	0.9405*
	CE	0.9671	0.5651	0.5577*
	MD	4.8068	0.1648	0.1709
HS-Brexit	F1	0.7917	0.8929*	0.9167*
	CE	0.7652	0.6154*	0.5889*
	MD	0.7985	0.2394*	0.2585*
MD-Agreement	F1	0.7880	0.7808*	0.8214*
	CE	0.9948	0.6629*	0.6245*
	MD	1.7574	0.3995*	0.3632*

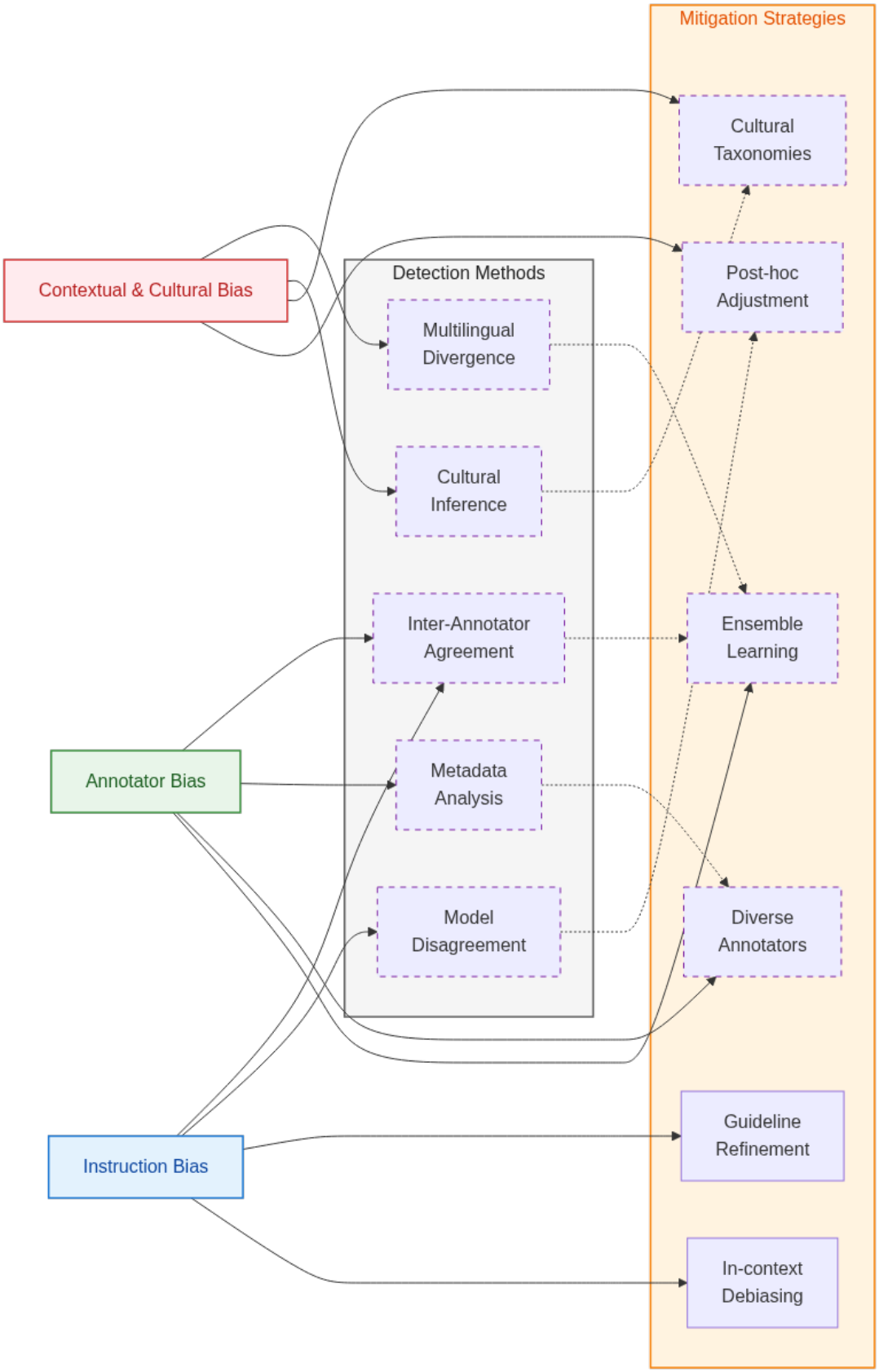

Figure 2: Relationships between annotation bias sources, detection methods, and mitigation strategies. Solid lines indicate primary correlations; dashed lines (purple) show secondary cross-connections.

Freeze and Reveal: Exposing Modality Bias in Vision-Language Models

Vivek Hruday Kavuri, Vysishtya Karanam, Venkata Jahnavi Venkamsetty,
Kriti Madumadukala, Lakshmipathi Balaji Darur, Ponnurangam Kumaraguru
IIIT Hyderabad
{kavuri.hruday, lakshmipathi.balaji}@research.iiit.ac.in,
{vysishtya.karanam, venkata.venkamsetty, kriti.madumadukala}
@students.iiit.ac.in, pk.guru@iiit.ac.in

Abstract

Vision-Language Models (VLMs) achieve impressive multimodal performance but often inherit gender biases from their training data. This bias might be coming from both the vision and text modalities. In this work, we dissect the contributions of vision and text backbones to these biases by applying targeted debiasing—Counterfactual Data Augmentation (CDA) and Task Vector methods. Inspired by data-efficient approaches in hate speech classification, we introduce a novel metric, *Degree of Stereotypicality* (DoS), and a corresponding debiasing method, *Data Augmentation Using DoS* (DAUDoS), to reduce bias with minimal computational cost. We curate a gender-annotated dataset and evaluate all methods on the VisoGender benchmark to quantify improvements and identify the dominant source of bias. Our results show that CDA reduces the gender gap by 6% and DAUDoS by 3% but using only one-third the data. Both methods also improve the model's ability to correctly identify gender in images by 3%, with DAUDoS achieving this improvement using only almost one-third of training data. From our experiments, we observed that CLIP's vision encoder is more biased whereas PaliGemma2's text encoder is more biased. By identifying whether the bias stems more from the vision or text encoders, our work enables more targeted and effective bias mitigation strategies in future multi-modal systems. We release our code public at https://github.com/vivekhruday05/VLM_bias

1 Introduction

The integration of visual and textual modalities in VLMs has led to remarkable advances in multimodal AI (Radford et al., 2021; Steiner et al., 2024; Li et al., 2022, 2023; Achiam et al., 2023; Team et al., 2023). VLMs have demonstrated exceptional capabilities across various tasks, including image retrieval (Xue et al., 2022; Bai et al., 2023), captioning (Li et al., 2022, 2023; Liu et al., 2024; Steiner et al., 2024). However these models often inherit gender biases present in their training data (Su et al., 2019) thus making them not suitable/reliable for real world deployment. Such biases also arise from stereotypical representations in both text and images, resulting in skewed perceptions that can propagate through downstream tasks.

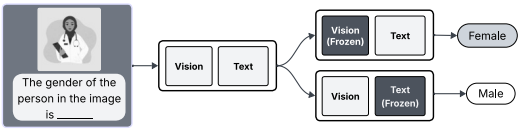

Figure 1: Different modalities posses different level of bias. We aim to show which one exhibits more bias.

In this work, we address these challenges by applying targeted debiasing techniques for both modalities. Specifically for a given VLM we debias a particular modality sub-module on a curated dataset and evaluate it for gender bias using VisoGender (Hall et al., 2023) to determine impact of each modality on gender bias. For this purpose we use the CelebA-Dialog dataset (Jiang et al., 2021) and curate the samples from the same. We annotate the data for gender based on the pronouns used in the caption and stereotypicality based on the statistical distribution of the data and insights from previous works (Fitousi, 2021; Muthukumar et al., 2018). To determine if a particular modality has higher influence in model's bias we evaluate it across multiple methods on our dataset. (i) We use CDA (Wu and Dredze, 2020; Webster et al., 2021; Zmigrod et al., 2019) a technique that mitigates bias by incorporating counterfactual data into the training process. (ii) We adapt Task Vector Unlearning (Dige et al., 2024; Ilharco et al., 2023;

Proceedings of the First Interdisciplinary Workshop on OMMM associated with RANLP 2025,
pages 17–26, Varna, Bulgaria, Sep 8-10, 2025.
https://doi.org/10.26615/978-954-452-101-1-002

Zhang et al., 2023) for debiasing. (iii) We propose a data-efficient debiasing approach, DAUDoS. We propose and do this for both CLIP-like similarity score based models and captioning type models and evaluate them across different methods. We consistently observe across multiple methods that CLIP's vision encoder is more biased compared to text encoder and in case of PaliGemma2, it's text encoder is more biased when compared to vision encoder.

In summary our key contributions are as follows:

- We propose a modality-targeted debiasing framework that applies CDA and Task-Vector methods separately to vision and text encoders to pinpoint each modality's bias.
- We curate a gender-annotated dataset for this analysis and evaluate our debiasing methods using the VisoGender benchmark.
- We propose DoS and introduce DAUDoS, lightweight debiasing methods that reduce gender bias on VisoGender with minimal overhead.

2 Related work

Bias in VLMs. VLMs such as CLIP and PaliGemma-2 have significantly advanced multimodal AI by integrating textual and visual modalities, enabling strong performance across diverse tasks. However, concerns have emerged regarding their tendency to inherit biases (Abdollahi et al., 2024; Darur et al., 2024; Xiao et al., 2024; Wolfe et al., 2023) present in training data, particularly gender bias. This bias can stem from both text and image components, as language models trained on large-scale Internet corpora frequently encode societal stereotypes, while image datasets may reinforce skewed gender representations by over representing specific demographics in certain professions, emotions, or activities. The interaction between these modalities further complicates the propagation of bias, making it crucial to determine whether textual or visual elements contribute more significantly to gender bias in VLMs. Previous works such as (Weng et al., 2024) focus on causal mediation to trace and mitigate gender bias in GLIP, showing image features contribute most and proposing input-level blurring to reduce bias. There are also works such as (Srinivasan and Bisk, 2022) which deal with bias measurement to multimodal models, revealing compounded intra and cross-modal stereotypes in VL-BERT. In contrast to these, our work targets a particular modality to find out which of the modalities contribute to a greater gender bias and whether they differ across different models and methods.

Bias Evaluation. Several studies have attempted to quantify and mitigate bias in AI models. Prior work has shown that word embeddings encode and perpetuate gender stereotypes in language representations (Zhao et al., 2019), that multimodal models like CLIP amplify both gender and racial biases in their image-to-text mappings (Steed and Caliskan, 2021). There are also existing real-world benchmarks which measure societal biases in generative models, emphasizing the need for robust evaluation frameworks (Gehman et al., 2020). Debiasing techniques focused on text prompts in multimodal models, indicating that interventions at the textual level can reduce bias to some extent but may not fully address the issue in vision-language interactions (Moreira et al., 2024).

Debiasing Techniques. To mitigate gender bias, researchers have proposed several debiasing techniques, including CDA and Task Vector methods. CDA works by synthetically generating counterfactual training data by swapping gendered terms (e.g., replacing "he" with "she"), thereby balancing gender representation in textual inputs (Zmigrod et al., 2020) and Task Vector (Ilharco et al., 2023) is an unlearning method which has it's roots originated from unlearing literature but also used in bias mitigation (Dige et al., 2024). While effective in NLP models, its application to VLMs remains underexplored.

Data-Efficient Debiasing. Training on all counterfactual examples can be computationally expensive and time-consuming. To address this, prior works (Nejadgholi et al., 2022; Garg et al., 2025) propose approaches for improving generalization in hate speech classification while relying on fewer annotated examples. These methods leverage Concept Activation Vectors (CAVs) and introduce a novel metric, the *Degree of Explicitness*, which quantifies the explicit nature of hateful content. By assigning explicitness scores to samples, they selectively fine-tune models on a curated subset of training instances, thereby enhancing efficiency without compromising performance. Inspired by these advances in NLP, we extend these ideas to the multi-modal setting and propose a novel metric

termed the *Degree of Stereotypicality* (DoS), which quantifies how strongly a sample exhibits stereotypical associations. Building on this, we introduce a data-efficient bias mitigation strategy called DuaDOS, which enables targeted augmentation based on stereotypicality scores. This approach reduces computational overhead while maintaining or improving model fairness and robustness in multimodal AI systems.

3 Dataset

We use the CelebA-Dialog dataset (Jiang et al., 2021) and curate the samples from the same. This dataset contains structured annotations describing different facial attributes of celebrities and ratings of each of the attributes on a scale of 0 to 5. The captions also include gender-specific pronouns such as *she*, *her*, *he*, *him*, etc., indicating the possibility of an implicit gender labeling task. Since, we require gender for each of the data point, both for applying our methods and evaluation, we annotate the gender and describe the process in the following subsections. We also need whether a data-point is stereotypical or anti-stereotypical, so that we can use for CDA. Hence, we also annotate that attribute and describe the process in the following subsections. An example of how initial data looks like is shown in Table 1.

Table 1: Examples of raw dataset samples with annotations. Each image is associated with both attribute-wise and overall captions, along with a numeric rating vector indicating the prominence of each attribute (e.g., bangs, eyeglasses, beard, smile, age) in order.

Image	
Bangs	He has no bangs at all. *Rating: 0*
Eyeglasses	There are no eyeglasses on the face. *Rating: 0*
Beard	This gentleman doesn't have any beard at all. *Rating: 0*
Smiling	This gentleman looks serious with no smile on his face. *Rating: 0*
Age	This person looks very old. *Rating: 5*
Overall Caption	This man in his eighties has no mustache, no fringe, and no smile. He is not wearing any eyeglasses.

3.1 Data Pre-processing and Annotation

First, we require gender labels for every data point. To achieve this, we employ a rule-based automatic labeler. Specifically, we search for gender-related terms or pronouns such as *his/her*, *he/she*, *gentleman/lady*, and *male/female*. Based on the presence of these words, we classify the data point as male or female. If none of these words appears, the annotator assigns the label *unknown*. This approach results in only 40 data points labeled as *unknown*, which is negligible compared to the size of the dataset, allowing us to prune them.

Next, we annotate the data points for stereotype classification. The dataset includes a rating from 0 to 5 for each data point across attributes {*Bangs, Smiling, No Beard, Young, Eye Glasses*}. Based on these ratings and predefined thresholds for stereotypical male and female characteristics, we label data points as either *stereotypical* or *anti-stereotypical*. These thresholds are determined by referring to prior publications and statistical insights from the dataset (Fitousi, 2021; Muthukumar et al., 2018). An example of a data point after the annotation is shown in Table 2.

4 Methodology

Our main objective is to determine which modality—vision or text—contributes more to gender bias in our selected models. To achieve this, as shown in the Figure 2, we independently debias the encoder for each modality while keeping the rest of the model frozen, and then assess the overall bias using our evaluation metrics. The modality that, when debiased separately, leads to a greater reduction in bias is considered to be inherently more biased.

This approach allows us to isolate the bias contributions of each encoder and provides insights

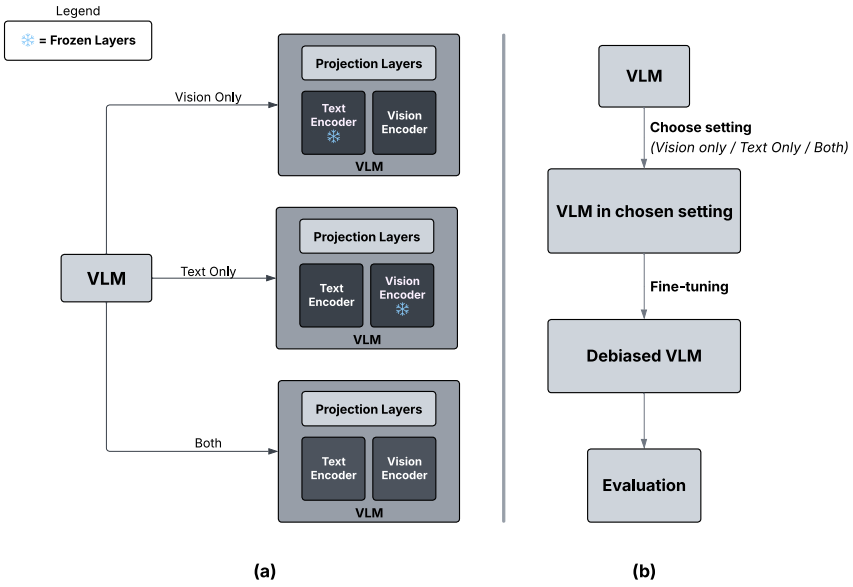

Figure 2: (a) Shows different layers that will be frozen in different settings we experiment in. (b) Shows an overall pipeline of our architecture. "Choose setting" means choosing a setting from one of the settings shown in (a).

Table 2: Data sample after preprocessing. Gender and stereotype labels are added based on rule-based and attribute rating analysis, respectively. Remaining attributes such as the ratings and individual captions are discarded.

Image	
Gender	Female
Stereotypical	False
Overall Caption	She has no smile and no bangs.This is a young child who has no eyeglasses.

into which modality is a more significant source of bias in the integrated VLM. To achieve this, we use pre-existing debiasing methods that debias the whole model to independently debias the encoder for each modality while keeping the rest of the model frozen. The debiasing methods we use are CDA and Weighted Task Vector.

4.1 Counter Factual Data Augmentation

As discussed in (Wu and Dredze, 2020; Webster et al., 2021; Zmigrod et al., 2019), Counterfactual Data Augmentation (CDA) is a technique that mitigates biases by incorporating counterfactual data into the training process. In this approach, the model is fine-tuned on augmented data that challenges stereotypical associations, which helps to attenuate biased representations.

We define counterfactual data as examples that contradict prevailing stereotypes. By augmenting these anti-stereotypical examples, we hypothesize that the model will better recognize and handle non-stereotypical patterns, thus reducing inherent biases. Given that our methodology requires pre-existing debiasing mechanisms to independently address biases in the model's multimodal encoders, CDA is integrated as one of the experimental settings in our study.

4.2 Task Vector

As discussed in (Dige et al., 2024; Ilharco et al., 2023; Zhang et al., 2023), the Task Vector is derived by subtracting the weights of a base model from those of a model fine-tuned on a specific task. To enhance flexibility in debiasing strength, we introduce a *weighted Task Vector method*, controlled by two hyperparameters: α and `blend`. Specifically, we adjust the original weights using:

$$W_{\text{debiased}} = W_{\text{original}} - ((1 - \texttt{blend}) \cdot \alpha) \cdot \Delta W_{\text{task}} \quad (1)$$

Here, α controls the overall intensity of debiasing, while $\texttt{blend} \in [0, 1]$ interpolates between the original and fully debiased model. A higher $\texttt{blend}$ retains more of the original model's behavior, while a lower value emphasizes debiasing more strongly.

To identify optimal hyperparameters, we perform a random search over $\alpha \in [0.1, 1.0]$ and $\texttt{blend} \in [0.0, 1.0]$, guided by a loss that balances accuracy and fairness:

$$\mathcal{L} = -\text{RA}_{\text{avg}} + \lambda_{\text{gap}} \cdot \text{GenderGap} \tag{2}$$

where RA_{avg} is the average resolution accuracy across male and female identities, and $\text{GenderGap} = |\text{RA}_m - \text{RA}_f|$ penalizes disparity. This formulation promotes both high performance and equitable behavior by controlling for bias introduced during fine-tuning.

4.3 Data Augmentation Using DoS (DAUDoS)

In this section, we introduce DAUDoS, a targeted debiasing strategy that leverages the stereotypicality of samples to perform efficient fine-tuning. The overall process is illustrated in Figure 3.

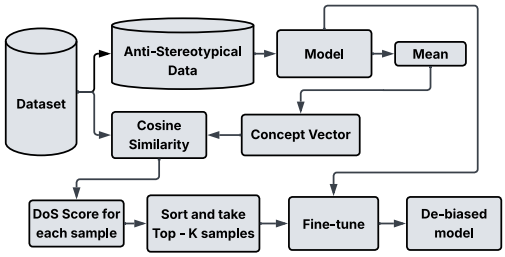

Figure 3: Depicting the method Data Augmentation Using DoS (DAUDoS). We first compute a concept vector from anti-stereotypical samples. Then, each dataset sample is scored based on its similarity to this vector, giving its Degree of Stereotypicality (DoS). The most stereotypical samples (more similarity with concept vector or score nearer to 1) are selected for fine-tuning, allowing targeted debiasing with minimal data.

The key idea behind DAUDoS is to assign a *Degree of Stereotypicality* (DoS) score to each sample in the dataset. To do so, we begin by constructing a small set of anti-stereotypical samples. These are fed into a pre-trained model to obtain embeddings, from which we compute a *Concept Activation Vector (CAV)*. Formally, if $\{\mathbf{z}_i\}_{i=1}^n$ are the model embeddings of the anti-stereotypical samples, the concept vector $\mathbf{v}_{\text{CAV}}$ is computed as their mean:

$$\mathbf{v}_{\text{CAV}} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{z}_i. \tag{3}$$

Next, for each input sample x, we obtain its model embedding $\mathbf{z}_x$ and compute its cosine similarity with $\mathbf{v}_{\text{CAV}}$:

$$\text{DoS}(x) = \cos(\mathbf{z}_x, \mathbf{v}_{\text{CAV}}). \tag{4}$$

This DoS score captures how closely the sample aligns with the concept of anti-stereotypicality: higher scores indicate lower stereotypicality, and vice versa.

Once scores are assigned, we sort all training samples by their DoS values and select the top-K most stereotypical samples for fine-tuning. This allows us to focus training on the subset of data that contributes most to bias, thereby making the process compute-efficient. These selected samples are used to fine-tune the model, leading to a debiased version as shown in Figure 3.

By guiding the data augmentation process with DoS, DAUDoS minimizes training cost while retaining effectiveness in bias mitigation across modalities.

5 Experiments

For CDA we use the anti-stereotypical examples from the dataset we annotated and fine-tune *openai/clip-vit-base-patch32*. Then for task vector, we used the stereotypical data to finetune the model and obtain task vector. In DAUDoS, we selected the samples based on the scores irrespective of what the label of the sample is (whether it is stereotypical or anti-stereotypical). We do these methods as discussed previously, in 4 different settings, namely:

Text only. In this setting, we freeze all the modules in a model except for the text encoder and projection layers related to text modality. There by only modifying the weights corresponding to the text encoder in the back propagation.

Vision Only. In this setting, we freeze all the modules in a model except for the vision encoder and projection layers related to vision modality. There by only modifying the weights corresponding to the vision encoder in the back propagation.

We use Nvidia Geforce 2080 Ti for finetuning the models on the anti-stereotypical data. We describe the evaluation pipeline and the results in the upcoming sections.

6 Results

To quantify gender bias in VLMs, as proposed in (Darur et al., 2024), we employ **Resolution Accuracy (RA)** as our primary metric. RA measures the classification performance for male (RA_m) and female (RA_f) labels by evaluating how accurately the model assigns gendered labels to images. We define the **Average Resolution Accuracy** (RA_{avg}) as the mean accuracy across male and female classifications:

$$RA_{avg} = \frac{RA_m + RA_f}{2} \tag{5}$$

Additionally, we compute the **Gender Gap (GG)** to quantify bias intensity by measuring the difference in resolution accuracy between male and female classifications:

$$GG = |RA_m - RA_f| \tag{6}$$

A higher GG indicates stronger gender bias, whereas a lower GG suggests more balanced performance across genders.

Our evaluation considers model logits and their corresponding gender preferences on the Visogender benchmark (Hall et al., 2023) in two settings: **Occupation-Object (OO)** and **Occupation-Participant (OP)**.

In the **OO** setting, each instance involves a single individual paired with an occupational cue; the model is tasked with assigning the correct gender label based solely on the visual representation and the occupational context. Conversely, the **OP** setting presents a more complex scenario in which each sample includes two individuals with different roles, requiring the model to simultaneously predict the gender of multiple participants. This dual framework enables us to assess the model's ability to handle both isolated and relational gender cues, thereby providing a comprehensive view of its fairness in gender classification.

After obtaining the gender preference scores and using the true labels of the dataset, we compute RA_{avg} and GG for various debiasing configurations. In the following subsections, we report the results for the CLIP and Paligemma2 models.

6.1 CLIP Results

Table 3 summarizes the performance of CLIP under different debiasing configurations. In the OO experiments, the *Raw Clip* baseline achieves an RA_{avg} of 0.94 and a moderate GG of 0.06. Debiasing the text encoder alone (text only) has almost same RA_{avg} 0.94 and decreases GG to 0.052. Notably, when the vision encoder is debiased (vision only), CLIP achieves an RA_{avg} of 0.96 with the gender gap completely eliminated ($GG = 0.0000$). A configuration where both encoders are left trainable (both) mirrors the outcome same as that of the case when the vision modality is debiased.

In the OP experiments (right columns of Table 3), the Raw CLIP model demonstrates a much lower accuracy compared to OO setting with RA_{avg} 0.56 and a high GG of 0.30. Debiasing the text encoder (text only) improves RA_{avg} to 0.57 and reduces GG to 0.17. Further improvement occurs when the vision encoder is debiased (vision only), yielding $RA_{avg} = 0.58$ and $GG = 0.08$. Finally, allowing both encoders to update (both) provides the highest RA_{avg} (0.63) with the lowest observed GG (0.06).

Figure 4 and Figure 5 display a plot of GG across the different debiasing configurations for

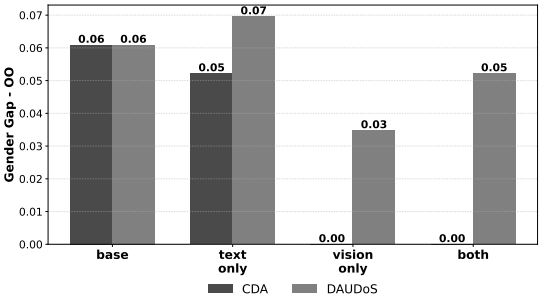

Figure 4: GG scores for **OO** setting in CLIP across debiasing configurations. Vision debiasing yields the least bias (GG = 0.0 by CDA, 0.03 by DAUDoS), similar to full model debiasing (GG = 0.0 by CDA, 0.05 by DAUDoS), indicating greater bias in the vision modality.

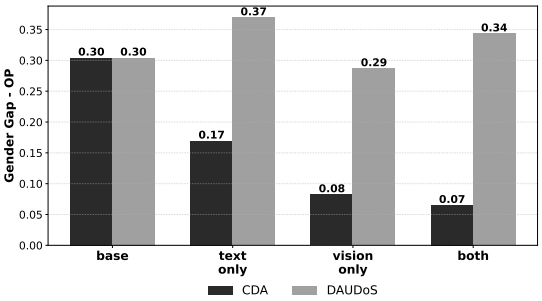

Figure 5: GG scores for **OP** setting in CLIP across debiasing configurations. Vision debiasing shows lowest bias (GG = 0.08 by CDA, 0.27 by DAUDoS), close to full model debiasing (GG = 0.07 by CDA, 0.34 by DAUDoS), again suggesting higher bias in the vision modality.

Table 3: Modality-targeted debiasing in CLIP under OO and OP settings. High RA implies better performance, low GG implies less bias. Debiasing the vision encoder in CLIP (Vision Only) achieves the highest RA_{avg} (0.97) with $GG = 0.00$, indicating vision contributes most bias.

	CDA							
Freeze Type	RA_m	RA_f	RA_{avg}	**GG**	RA_m	RA_f	RA_{avg}	**GG**
	OO				**OP**			
Raw Clip	0.91	0.97	0.94	0.06	0.41	0.65	0.56	0.30
Text Only	0.91	**0.97**	0.94	0.05	0.48	0.65	0.57	0.17
Vision Only	**0.97**	**0.97**	**0.97**	**0.00**	*0.54*	0.62	*0.58*	*0.08*
Both	**0.97**	**0.97**	**0.97**	**0.00**	**0.60**	*0.66*	**0.63**	**0.07**
Task Vector (α = 0.56, blend = 0.78)								
Text Only	*0.17*	**0.75**	**0.46**	0.57	0.10	*0.02*	0.06	**0.08**
Vision Only	**0.63**	0.23	*0.43*	*0.39*	**0.56**	**0.22**	*0.39*	0.33
Both	0.07	*0.26*	0.17	**0.19**	*0.30*	0.01	0.15	*0.29*
DAUDoS								
Text Only	0.91	**0.98**	*0.95*	0.07	0.38	*0.75*	0.57	0.37
Vision Only	**0.94**	*0.97*	**0.96**	**0.03**	**0.46**	0.74	*0.60*	**0.29**
Both	*0.93*	**0.98**	**0.96**	*0.05*	*0.44*	**0.78**	**0.61**	0.34

Table 4: Modality-targeted debiasing in PaliGemma2 under OO and OP settings. High RA implies better performance, low GG implies less bias. Debiasing the text encoder in PaliGemma2 (text only) yields $RA_{\text{avg}} = 0.99$ with $GG = 0.01$, showing text is the primary bias source.

	CDA							
Freeze Type	RA_f	RA_m	RA_{avg}	**GG**	RA_f	RA_m	RA_{avg}	**GG**
	OO				**OP**			
Raw Paligemma	0.79	0.46	0.63	0.33	0.90	0.45	0.68	0.45
Text Only	**0.99**	**0.98**	**0.99**	**0.01**	0.72	0.78	0.75	*0.07*
Vision Only	0.42	0.39	0.40	*0.03*	0.65	0.47	0.56	0.18
Both	0.98	0.97	0.97	**0.01**	**0.76**	**0.86**	**0.81**	0.10
DAUDoS								
Text Only	*0.90*	**0.99**	*0.94*	*0.09*	**0.65**	*0.87*	**0.76**	*0.23*
Vision Only	0.48	0.67	0.57	0.19	0.50	0.80	0.65	0.30
Both	**0.93**	**0.99**	**0.96**	**0.06**	0.52	**0.91**	*0.72*	0.39

CLIP, clearly illustrating that interventions aimed at debiasing the vision encoder (vision only setting) are particularly effective in lowering the gender gap. Hence, the more biased encoder in CLIP is vision encoder. We can observe this result consistently across methods.

6.2 Paligemma2 Results

Table 4 shows the performance of the Paligemma2 model under similar conditions. In the CDA experiments, configurations such as "text only" and "both" achieve very high RA_{avg} (approximately 0.97–0.99) while maintaining a very low gender gap (e.g., GG=0.01 for text only). For the DAUDoS setting, while the RA_{avg} remains high (around 0.94–0.96), it is important to note that these results were obtained using only one-third of the dataset. This aligns with our objective of achieving competitive performance using minimal data—demonstrating that selective sampling is both efficient and effective. Using the entire dataset would defeat the purpose of our sorting and data reduction strategy.

In the OP experiments (right columns of Table 4), the Raw model demonstrates similar accuracy compared to OO setting with RA_{avg} 0.68 and a high GG of 0.45. Debiasing the text encoder (text only) improves RA_{avg} to 0.75 and reduces GG to 0.07. But, notably no further improvement occurs when the vision encoder is debiased (vision only), yielding $RA_{avg} = 0.56$ and $GG = 0.18$. Finally, allowing both encoders to update (both) provides the highest RA_{avg} (0.81) but the Gender Gap GG of (0.06) is still higher than the gender gap observed in case of text only setting.

Figure 6 and Figure 7 provide a plot of GG for

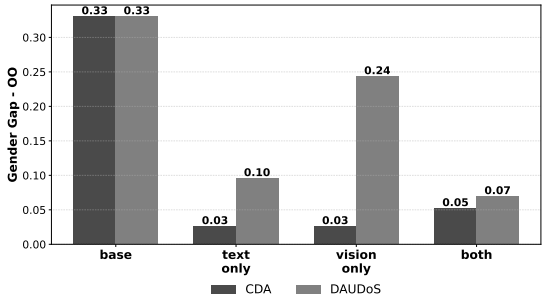

Figure 6: GG scores for **OO** setting across debiasing configurations for Paligemma2. Text debiasing yields lowest bias (GG = 0.03 by CDA, 0.10 by DAUDoS), similar to full model debiasing (GG = 0.05 by CDA, 0.07 by DAUDoS), suggesting higher bias in text modality.

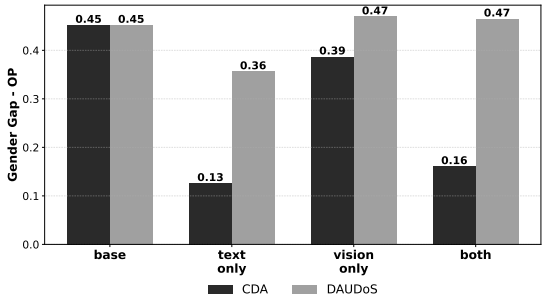

Figure 7: GG scores for **OP** setting across debiasing configurations for Paligemma2. Text debiasing gives lowest bias (GG = 0.13 by CDA, 0.36 by DAUDoS), close to full model debiasing (GG = 0.16 by CDA, 0.47 by DAUDoS), again pointing to text as the more biased modality.

the Paligemma2 model, reinforcing the trend that debiasing the text modality (Text Only) is particularly effective in reducing gender bias. Hence the more biased modality in PaliGemma2 is the text modality. We can observe this result consistently across methods.

7 Discussion

Our study investigates gender bias in VLMs by independently debiasing the text and vision encoders using methods like CDA and Task Vectors. Experiments on the CelebA-Dialogue dataset and evaluations with the VisoGender benchmark reveal that targeting individual modalities is more effective than intervening at the model level. In CLIP, debiasing the vision encoder yields lower gender gaps with minimal impact on accuracy—likely due to the balanced parameter sizes across modalities. In contrast, PaliGemma2's larger text encoder (2.5B parameters vs. 0.5B for vision) makes debiasing the text modality more impactful.

The findings also underscore that modality-specific debiasing leads to better bias mitigation than strategies applied post-encoder, such as projection layer adjustments, which only offer limited improvements. Our proposed DAUDoS method further supports this trend, demonstrating the generalizability of our approach across models and settings.

To conclude, we conduct experiments on the CelebA-Dialogue dataset and evaluate the outcomes using the VisoGender benchmark. Results consistently reveal that targeted debiasing of individual encoders mitigates gender bias more effectively while preserving overall model performance. By demonstrating that targeted interventions reduce gender bias while preserving performance, our work contributes practical insights for building fairer vision-language systems.

Limitations

Despite these contributions, our study has limitations. First, the use of binary gender annotations excludes non-binary and LGBTQ+ identities, restricting the inclusiveness of our evaluation. Second, our focus is limited to gender bias and does not consider intersectional biases, such as those related to race or age.

Future Work

In future work, we plan to broaden the scope of our analysis to address intersectional biases, such as those involving race, age, and skin tone, which may interact with gender in complex ways. This would allow for a more nuanced understanding of model fairness across diverse identities. Additionally, investigating the temporal and contextual dynamics of bias—such as how models adapt to evolving cultural norms or contextual cues can offer deeper insights into the stability and robustness of debiasing methods.

Another important direction is exploring bias mitigation strategies during the pretraining phase, rather than only through fine-tuning, to assess whether early interventions result in more systemic improvements. Finally, we plan to test our methods in real-world deployment scenarios such as image captioning, content moderation, and recommendation systems, to evaluate both fairness and utility in applied settings.

References

Ali Abdollahi, Mahdi Ghaznavi, Mohammad Reza Karimi Nejad, Arash Mari Oriyad, Reza Abbasi, Ali Salesi, Melika Behjati, Mohammad Hossein Rohban, and Mahdieh Soleymani Baghshah. 2024. *GABInsight: Exploring Gender-Activity Binding Bias in Vision-Language Models*. IOS Press.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Yang Bai, Xinxing Xu, Yong Liu, Salman Khan, Fahad Khan, Wangmeng Zuo, Rick Siow Mong Goh, and Chun-Mei Feng. 2023. Sentence-level prompts benefit composed image retrieval. *arXiv preprint arXiv:2310.05473*.

L. Darur, S.K. Gouravarapu, S. Goel, and P. Kumaraguru. 2024. Improving bias metrics in vision-language models by addressing inherent model disabilities. In *Workshop on Algorithmic Fairness through the Lens of Metrics and Evaluation, NeurIPS 2024*.

Omkar Dige, Diljot Arneja, Tsz Fung Yau, Qixuan Zhang, Mohammad Bolandraftar, Xiaodan Zhu, and Faiza Khan Khattak. 2024. Can machine unlearning reduce social bias in language models? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 954–969, Miami, Florida, US. Association for Computational Linguistics.

Daniel Fitousi. 2021. Stereotypical processing of emotional faces: Perceptual and decisional components. *Frontiers in Psychology*, 12.

Samarth Garg, Vivek Hruday Kavuri, Gargi Shroff, and Rahul Mishra. 2025. Ktcr: Improving implicit hate detection with knowledge transfer driven concept refinement.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. 2020. Realtoxicityprompts: Evaluating neural toxic degeneration in language models.

Siobhan Mackenzie Hall, Fernanda Gonçalves Abrantes, Hanwen Zhu, Grace Sodunke, Aleksandar Shtedritski, and Hannah Rose Kirk. 2023. Visogender: A dataset for benchmarking gender bias in image-text pronoun resolution.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2023. Editing models with task arithmetic.

Yuming Jiang, Ziqi Huang, Xingang Pan, Chen Change Loy, and Ziwei Liu. 2021. Talk-to-edit: Fine-grained facial editing via dialog. In *Proceedings of International Conference on Computer Vision (ICCV)*.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pages 12888–12900. PMLR.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. 2024. Llava-next: Improved reasoning, ocr, and world knowledge.

Diego A. B. Moreira, Alef Iury Ferreira, Jhessica Silva, Gabriel Oliveira dos Santos, Luiz Pereira, João Medrado Gondim, Gustavo Bonil, Helena Maia, Nádia da Silva, Simone Tiemi Hashiguti, Jefersson A. dos Santos, Helio Pedrini, and Sandra Avila. 2024. Fairpivara: Reducing and assessing biases in clip-based multimodal models.

Vidya Muthukumar, Tejaswini Pedapati, Nalini Ratha, Prasanna Sattigeri, Chai-Wah Wu, Brian Kingsbury, Abhishek Kumar, Samuel Thomas, Aleksandra Mojsilovic, and Kush R. Varshney. 2018. Understanding unequal gender classification accuracy from face images.

Isar Nejadgholi, Kathleen Fraser, and Svetlana Kiritchenko. 2022. Improving generalizability in implicitly abusive language detection with concept activation vectors. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5517–5529, Dublin, Ireland. Association for Computational Linguistics.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision.

Tejas Srinivasan and Yonatan Bisk. 2022. Worst of both worlds: Biases compound in pre-trained vision-and-language models.

Ryan Steed and Aylin Caliskan. 2021. Image representations learned with unsupervised pre-training contain human-like biases. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 701–713. ACM.

Andreas Steiner, André Susano Pinto, Michael Tschannen, Daniel Keysers, Xiao Wang, Yonatan Bitton, Alexey Gritsenko, Matthias Minderer, Anthony Sherbondy, Shangbang Long, Siyang Qin, Reeve Ingle, Emanuele Bugliarello, Sahar Kazemzadeh, Thomas Mesnard, Ibrahim Alabdulmohsin, Lucas Beyer, and Xiaohua Zhai. 2024. Paligemma 2: A family of versatile vlms for transfer.

Hui Su, Xiaoyu Shen, Rongzhi Zhang, Fei Sun, Pengwei Hu, Cheng Niu, and Jie Zhou. 2019. Improving multi-turn dialogue modelling with utterance ReWriter. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 22–31, Florence, Italy. Association for Computational Linguistics.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Kellie Webster, Xuezhi Wang, Ian Tenney, Alex Beutel, Emily Pitler, Ellie Pavlick, Jilin Chen, Ed Chi, and Slav Petrov. 2021. Measuring and reducing gendered correlations in pre-trained models.

Zhaotian Weng, Zijun Gao, Jerone Andrews, and Jieyu Zhao. 2024. Images speak louder than words: Understanding and mitigating bias in vision-language model from a causal mediation perspective.

Robert Wolfe, Yiwei Yang, Bill Howe, and Aylin Caliskan. 2023. Contrastive language-vision ai models pretrained on web-scraped multimodal data exhibit sexual objectification bias. In *2023 ACM Conference on Fairness Accountability and Transparency*, FAccT '23, page 1174–1185. ACM.

Shijie Wu and Mark Dredze. 2020. Are all languages created equal in multilingual BERT? In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 120–130, Online. Association for Computational Linguistics.

Yisong Xiao, Aishan Liu, QianJia Cheng, Zhenfei Yin, Siyuan Liang, Jiapeng Li, Jing Shao, Xianglong Liu, and Dacheng Tao. 2024. Genderbias-*VL*: Benchmarking gender bias in vision language models via counterfactual probing.

Hongwei Xue, Yuchong Sun, Bei Liu, Jianlong Fu, Ruihua Song, Houqiang Li, and Jiebo Luo. 2022. Clip-vip: Adapting pre-trained image-text model to video-language representation alignment. *arXiv preprint arXiv:2209.06430*.

Jinghan Zhang, Shiqi Chen, Junteng Liu, and Junxian He. 2023. Composing parameter-efficient modules with arithmetic operations.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Ryan Cotterell, Vicente Ordonez, and Kai-Wei Chang. 2019. Gender bias in contextualized word embeddings.

Ran Zmigrod, Sabrina J. Mielke, Hanna Wallach, and Ryan Cotterell. 2019. Counterfactual data augmentation for mitigating gender stereotypes in languages with rich morphology. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1651–1661, Florence, Italy. Association for Computational Linguistics.

Ran Zmigrod, Sabrina J. Mielke, Hanna Wallach, and Ryan Cotterell. 2020. Counterfactual data augmentation for mitigating gender stereotypes in languages with rich morphology.

AnthroSet: a Challenge Dataset for Anthropomorphic Language Detection

Dorielle Lonke[1] **Jelke Bloem**[1] **Pia Sommerauer**[2]
[1]University of Amsterdam, Institute for Logic, Language and Computation
[2]Vrije Universiteit Amsterdam, Computational Linguistics and Text Mining Lab
dorielle.lonke@student.uva.nl j.bloem@uva.nl pia.sommerauer@vu.nl

Abstract

This paper addresses the challenge of detecting anthropomorphic language in AI research. We introduce AnthroSet, a novel dataset of 600 manually annotated utterances covering various linguistic structures. Through the evaluation of two current approaches for anthropomorphism and atypical animacy detection, we highlight the limitations of a masked language model approach, arising from masking constraints as well as increasingly anthropomorphizing AI-related terminology. Our findings underscore the need for more targeted methods and a robust definition of anthropomorphism.

1 Introduction

With the evolving popularity and applications of AI systems, the terms used to describe their functionalities have become increasingly anthropomorphizing (Floridi and Nobre, 2024). The tendency to attribute human-like capabilities and properties to AI systems involves various topics of interest, including cognitive and psychological analyses (Waytz et al., 2010; Hofstadter, 1995), ethical considerations and accountability (Salles et al., 2020), and undue AI hype (Placani, 2024; Barrow, 2024).

While the topic of anthropomorphism in AI is widely discussed, there is not one clear definition of what it entails. Efforts to describe anthropomorphic language focus on AI output rather than texts about AI (for examples see DeVrio et al. (2025); Emnett et al. (2024)). Detecting anthropomorphic language in human text is particularly difficult as it is highly contextual (Cheng et al., 2024), ambiguous and subjective (Waytz et al., 2010; Shardlow et al., 2025). There are currently only two open-source implementations for detecting the attribution of human properties to machines in text, both relying on a masked language model (MLM) approach that detects anthropomorphism by measuring the animacy of a masked entity (Coll Ardanuy et al., 2020; Cheng et al., 2024).

We present AnthroSet, an evaluation dataset consisting of 600 manually annotated utterances representing types of anthropomorphic language pertaining to AI. We provide a variety of linguistic structures in which anthropomorphic language is expressed, drawn from academic literature on AI. The purpose of this dataset is twofold: first, we aim to provide concrete examples of anthropomorphic language in contemporary AI research, grounded in a linguistic analysis of anthropomorphism and animacy markers in the English language, as well as the anthropomorphic language taxonomy by DeVrio et al. (2025). The second is to evaluate the state-of-the-art, open-source methods for anthropomorphic language detection.

Our results highlight the problems with employing a masked language model approach for this task. For one, the masking is consequential in achieving good results, but a uniform masking approach is not suitable for all syntactic structures. Second, as AI-related terminology becomes increasingly anthropomorphizing, MLMs are more likely to associate AI entities with anthropomorphic verbs and descriptors, simply due to their reliance on statistical co-occurrence (Zhang et al., 2024), posing further challenges for anthropomorphism detection.

2 Related Work

The tendency to attribute human-like capacities to AI systems has been observed since the foundation of AI as a field of research. The general relation between cognition and machines has been widely discussed, with authors such as Dreyfus (1976) and Searle (1980) arguing against the reduction of human thought to syntactic and symbolic programs. In the context of psychology, the *ELIZA* effect was defined as the cognitive bias that causes human

Proceedings of the First Interdisciplinary Workshop on OMMM associated with RANLP 2025,
pages 27–39, Varna, Bulgaria, Sep 8-10, 2025.
https://doi.org/10.26615/978-954-452-101-1-003

users to attribute human-like properties such as intelligence and emotions to responsive machines (Hofstadter, 1995). The tendency to anthropomorphize AI by means of the language we use was recognized by McDermott (1976) as *wishful mnemonics* – the methodological tendency to name and describe AI programs not in terms of what they actually do, but as what they are intended and willed by us to do. Anthropomorphism in AI can be seen as a metaphoric device, whose explanatory powers contribute to the evolution of emerging technologies (Carbonell et al., 2016), and are used both for explanation as well as persuasion (Rossi and Macagno, 2021). In recent years, anthropomorphic language in AI discourse has been addressed from an ethical perspective, touching on issues related to society and accountability (Watson, 2019; Salles et al., 2020; Placani, 2024).

There currently are two open-source implementations of anthropomorphism detection: Cheng et al. (2024) developed *AnthroScore*, a metric for measuring implicit anthropomorphism in contemporary scientific research and downstream media. Their approach is similar to the one presented by Coll Ardanuy et al. (2020) in *Living Machines: A study of atypical animacy*, which aims at detecting atypical animacy by focusing on scenarios in which machines are represented as having animate attributes. Recently, DeVrio et al. (2025) proposed a taxonomy of linguistic expressions in AI-generated text that contribute to anthropomorphism in AI, setting forth a theoretical baseline for anthropomorphic language analysis.

Shardlow et al. (2025) present the first corpus annotated for anthropomorphic language in the context of LLMs[1]. Their corpus is based on abstracts from the ACL Anthology and news articles, annotated at the sentence level. Their annotation focuses on classifying claims as non-anthropomorphic, ambiguously anthropomorphic or explicitly anthropomorphic as outlined in Shardlow and Przybyła (2024), following the subjective judgements of annotators. No annotation guidelines are available, but the intermediate category seems to be defined as the case where "someone who is familiar with this language would correctly interpret it as a metaphor, whereas a novice or lay reader may well infer human characteristics". The scheme is not otherwise defined in linguistic terms. 4340 sentences were annotated. They also perform scoring using encoder LLMs such as XLNet with a regression classifier head tuned on labeled data, though these models are not available at the time of writing.

[1]This work was published after we finished our study.

3 Linguistic Structures

Anthropomorphism, particularly pertaining to AI and machines, can be expressed through a variety of different syntactic and semantic structures. We differentiate between explicit anthropomorphism, i.e. sentences or expressions that directly and overtly attribute human-like capacities such as cognition, intention or mental states to AI systems through their contents, and *implicit* anthropomorphism – which is indirect, sometimes covert, and rises from certain lexical or contextual meanings. We identified prominent structures on the basis of a linguistic analysis of anthropomorphism and animacy markers in the English language, combined with a frame semantics approach that considers the lexical units in the sentence with respect to their thematic roles and the frames that they evoke (see Ryazanov et al. (2024)). For example, in the sentence 'The system decides to trust the user', the entity in the subject position ('system') is anthropomorphic as it plays the thematic role of AGENT, whose properties are sentience, volition, movement, causing an event or change of state, and existing independently of the event (Dowty, 1991; Levin, 2022). Additionally, the verb phrase 'decide to trust' is anthropomorphic as it entails the capacity for cognitive processes such as decision making and the mental state of trust. Thematic agents can occur in the object position in passive voice structures. For instance, in the sentence 'The users were deceived by the model' the verb frames the AI entity as having intention or malevolence. The AI entity can also embody the thematic role of EXPERIENCER, attributed cognitive and mental states as either subject or object of certain cognitive or *psych verbs* (Belletti and Rizzi, 1988). For example, in 'The developers tricked the system into believing the lies', the object-experiencer verb 'trick' contributes to the framing of the AI entity as having cognitive and mental faculties, suggesting it has the capacity to be tricked.

Importantly, not all anthropomorphic lexical units are verbs: adjectives can attribute human-like abilities by means of description, e.g. *conscious*, *aware*, *confident*, *benevolent*, and *malicious*; certain nouns which are often collocated with AI entities are otherwise traditionally reserved for human

roles, such as *assistant*, *teacher* or *judge*. We might also identify anthropomorphism in sentences that embody genitive structures in which an AI entity is described as possessing certain abilities, traits or properties, e.g. 'the model's advanced reasoning abilities', or contain comparative function words, e.g. 'Like children, language models learn from patterns'. While syntactic in nature, these structures are best understood alongside a taxonomy of anthropomorphic lexical units and their semantics, which we have defined on the basis of the one constructed by DeVrio et al. (2025). Based on their guiding lenses for identifying anthropomorphic patterns in synthetic text, as well as an analysis of numerous real-life examples from published papers in AI, we identified the affective and cognitive capacities aimed at elucidating anthropomorphic language in human-written text, used by human authors to describe AI systems in contemporary AI research. This taxonomy is shown in Appendix C.

4 Task and Models

We aim to shed light on the current definitions and interpretations of anthropomorphic language in AI research, and the means for identifying it in text. To that end, we evaluated and compare two implementations of anthropomorphism detection in the domain of AI and machines. We compiled and manually annotated an evaluation set consisting of examples of anthropomorphic language in the context of AI, i.e. language that humanizes AI systems by attributing to them human-like capacities of cognition, intention and mental states, and compare and examine the performance of each approach in detecting these patterns[2].

4.1 Models

Both approaches rely on a masked language model to predict the likelihood of a masked entity, corresponding to an AI model, system or machine to be construed as human. The AnthroScore method uses the HuggingFace implementation of RoBERTa (`roberta-base`, 125M parameters), a pre-trained masked language model (MLM) as the model and tokenizer. The *Living Machines* method (henceforth referred to as *AtypicalAnimacy*) is based on the the HuggingFace implementation of BERT (`BERT-base`, 110M parameters), fine-tuned on an atypical animacy detection dataset consisting of 19th-century texts related to industrialization and machines. The AnthroScore method provides a metric for measuring the degree of anthropomorphism in a given set of texts for a given set of entities. Given a sentence containing a masked entity, a high- or low-anthropomorphism score is obtained by computing the probabilities that the MLM predicts animate pronouns (*he*, *she*) and inanimate ones (*it*, *its*), and calculating the log of the ratio between the probabilities. AtypicalAnimacy also rely on MLM prediction of a masked token, determining the animacy of the expression within a sentence by averaging the animacy of the top predicted tokens. This is determined using WordNet, by disambiguating the predicted token to its most relevant word sense, and checking whether that sense is a descendant of the *living thing* node. Then, a score between 0 and 1 is produced by calculating the weighted average of the predicted token scores, and a final binary score is determined by an optimal animacy threshold.

5 AnthroSet

Our evaluation set consists of sentences taken from abstracts of papers published on ACL Anthology and arXiv. Relevant papers were selected from the ACL Anthology[3] and arXiv[4] datasets, using a list of keywords (*AI*, *artificial intelligence*, *(language) model*, *system*, *LM*, *LLM*, *GPT*, *ChatGPT*). First, we identified relevant papers by searching for the keywords in the title. Then, we found potentially anthropomorphic utterances by searching for sentences containing these keywords in the abstract. To narrow down the search, we compiled word lists of anthropomorphic verbs, nouns and adjectives, corresponding to our taxonomy of anthropomorphic attributes (Appendix C). These lists were then extended with similar words using WordNet to include synonyms and semantically related entries.

We included samples covering all linguistic structures described in section 3, which are henceforth referred to as follows: (1) *verb subjects* – an

[2]The phenomenon addressed in *Living Machines* pertains to a general sense of animacy, which encompasses the more specific notion of *humanness*. This specification is used to distinguish between sentences describing the humanization of machines through comparisons to humans, which are examples of both *animacy* and *humanness*, versus those depicting the dehumanization of humans through comparison to machines, which corresponds only to *animacy*. Since our dataset focuses on machines and AI and not humans, we interpret the *Living Machines* notion of animacy as equivalent to AnthroScore's definition of anthropomorphism.

[3]https://acl-anthology.readthedocs.io/latest/api/anthology/

[4]https://www.kaggle.com/datasets/Cornell-University/arxiv

AI entity as the subject of an anthropomorphic verb, (2) *verb objects* – an AI entity as the object of an anthropomorphic verb, (3) *adjectives* – an AI entity collocated with an anthropomorphic adjective, (4) *role/function noun phrases* – an AI entity described as performing an anthropomorphic role or function[5], (5) *genitive noun phrases* – an AI entity described as being in possession of an anthropomorphic NP, and (6) *comparisons* of AI entities to human beings. An example of each of these structures is shown in Appendix B.

For each linguistic structure, we searched for the particular dependency relations between the lexical unit and the AI entity. For example, to find anthropomorphic adjectives, we looked for AI entities that are either modified by an `amod` or complemented by a `acomp` which belongs to the extended list of anthropomorphic adjectives. We then manually reviewed and selected the candidate sentences based on our annotation guidelines (Appendix A), modeled in part after the VU Metaphor Identification Procedure (Steen et al., 2010). Since we queried for different dependency relations, we ended up with a pooled dataset divided into subsets categorized by their syntactic structures.

5.1 Annotation procedure

The linguistic category sets are divided into multiclass (*verb subjects*, *verb objects* and *adjectives*) which have positive, negative and inconclusive samples, and single-class, which are either always positive (*role/function NPs*, *genitive NPs*) or always inconclusive (*comparisons*). While verbs and adjectives tend to be much more context-sensitive and ambiguous, structures describing AI systems as performing a role or in possession of certain properties are anthropomorphic only to the extent that they feature a anthropomorphic NP. In that respect, negative samples are not clearly defined, and thus were not included in the evaluation set. As a result, we excluded these categories from the overall comparison, which is done in terms of precision, recall and F1-score, and only measure accuracy on these sets. Comparisons in which AI entities are likened to humans can be either understood as highly anthropomorphizing as their content attributes to AI qualities or properties of humans, or they could be seen as non-anthropomorphizing since the explicit comparison serves to contrast AI and humans, and highlight their differences (Coll Ardanuy et al., 2020). Because of this dual interpretation, we decided to treat these cases as inconclusive, and included them in the evaluation only as an aid for understanding model behavior[6].

For the annotation task, annotators were presented with batches of sentences where the target AI entity was highlighted in bold, along with our guidelines and a decision tree (given in Appendix A). The labels 'positive', 'negative' or 'inconclusive' were used to label the anthropomorphization of the target AI entity in the context of the sentence, following this decision tree.

Our annotators have expertise in linguistics and were aware of the research purpose of the benchmark. All instances were annotated by a primary annotator and to evaluate inter-annotator agreement, a subset of 20% of the multiclass cases was divided among two secondary annotators. The first set, which had a balanced distribution of positive, negative and inconclusive cases had a Cohen's κ of 0.39 for all cases, and a much higher κ of 0.92 for just positive and negative cases. The second set, which consisted of twice as many inconclusive cases than positive and negative cases had a Cohen's κ of 0.22 for all cases, and 0.60 on just the positive and negative cases[7]. The low κ for the overall cases reflects the difficult nature of this annotation task, especially on borderline cases which do not have enough contextual cues, even for human annotators, to determine whether or not an entity is being anthropomorphized. Additionally, while we relied on a taxonomy of anthropomorphic language, deciding whether a certain lexical unit embodies these definitions is not a trivial task. Nevertheless, the Cohen κ for our non-borderline cases shows that these were for the most part agreed upon. No disagreement resolution was performed.

To support future work, including robust re-

[5]This definition resonates with *task-based anthropomorphism* (Ryazanov et al., 2024), a form of anthropomorphic descriptions of AI systems which pertains to humanizing language describing functionality.

[6]In some interpretations of anthropomorphism, noun phrases such as *AI teacher* or *AI judge* might not be seen as inherently anthropomorphizing, rather understood as comparisons in which AI is likened, but not identified with humans. Based on our definition of anthropomorphism, we have decided to treat these cases as positive.

[7]This was checked by first including all cases, and then filtering out cases in which at least one of the annotators was inconclusive. We also calculated Cohen's κ for each class by creating a binary mapping, and had $\kappa = 0.62$ for positive cases and $\kappa = 0.49$ for negative cases in the first set, and $\kappa = 0.40$ for positive cases and $\kappa = 0.22$ for negative cases in the second set. Inconclusive cases had a very low agreement rate due to their borderline nature, but this was expected.

dundant annotation and expanding the dataset, we made our annotated set publicly accessible on GitHub[8]. The annotated dataset contains 297 (49%) positive, 173 (29%) negative and 131 (22%) inconclusive cases. This contrasts with the corpus of Shardlow et al. (2025), who found 3.7% explicit anthropomorphism, 19.3% ambiguous anthropomorphism and 77% negative cases. However, we specifically selected sentences containing potentially anthropomorphic language to create a benchmark, while their corpus aims to document the frequency of anthropomorphic language in news articles and ACL abstracts and thus covers a subset of data without selection or filtering.

6 Experiments

We employed two masking strategies in our experimental setup. The first is AnthroScore's built-in masking method, which relies on keyword identification and noun-chunking. We found that while it is suitable for identifying certain structures, particularly those in which the anthropomorphic component complements or predicates the AI entity, it tends to mask crucial contextual cues for other anthropomorphic structures, such as adjectival modifiers, noun phrases, or certain possessive structures. For example, the phrase 'conscious AI systems' is masked in its entirety by AnthoScore's masking strategy, even though the main contribution to anthropomorphism is the adjectival modifier 'conscious'. The second is our own masking strategy (referred to henceforth as *minimal entity* masking), which was put forth in order to preserve the anthropomorphic cues in the context rather than mask them. Our masking strategy works as follows: given an AI keyword (a single keyword such as *AI*, *LLM*, *model*, or *ChatGPT*), we manually masked the minimal phrase referring to an AI entity[9], masking additional modifiers only in case they are part of the name, or an essential part of its description, e.g. relating to its functionality or purpose (i.e. *conversational AI*, *question answering system* or *large language model*). We left out any descriptors that are contingent to the description, such as *powerful*, *complex*, or *flexible*.

[8] https://github.com/doriellel/anthroset

[9] Our masking strategy required manual revision, but proved significantly better than the automatic chunking method employed by AnthroScore. In future work, this could be improved by implementing something like a NER pipeline that would identify particular AI entities, rather than capturing an entire noun chunk or manually reviewing every occurrence.

6.1 Metrics and score mapping

We evaluated each system on the multiclass sets (*verb subjects*, *verb objects* and *adjectives*) in terms of precision, recall and F1-score. We observed these both as macro-averaged aggregates for each syntactic category, as well as per class. On the single-class positive sets (*role/function NPs* and *genitive NPs*), we only looked at the systems' recall (i.e. accuracy – the number of positive predictions out of total predictions). In the case of all inconclusive sentences (*verb subjects*, *verb objects* and *adjectives* and *comparisons*), since 'inconclusive' does not represent a gold label but rather a lack thereof, we did not measure accuracy. Instead, we observed the trends, and compare each system's tendency to predict positive, negative (and inconclusive in the case of AnthroScore) in those cases.

To compare the performance of both approaches, we mapped the AnthroScores to those of AtypicalAnimacy. AnthroScore does not provide a binary score, but rather high-anthropomorphism and low-anthropomorphism scores. A high score is higher than 1 (i.e. the probability to predict human pronouns is higher than non-human ones, resulting in the log of the ratio to be greater than 1), and, symmetrically, a low score is lower than -1. Scores that fall between 1 and -1 reflect an equal likelihood for both pronouns to be predicted by the MLM, corresponding to our definition of inconclusive cases. AtypicalAnimacy provides binary scores of 1 and 0. To obtain binary results for AnthroScore as well, we mapped AnthroScores >1 to 1, and scores $<$-1 to 0, and conduct the evaluation after the mapping. To compare precision, recall and F1, we simply interpreted AnthroScores between 1 and -1 as false negatives of either class, and exclude inconclusive cases from the gold set.

6.2 Evaluation results

Each method was evaluated twice on all six categories of syntactic structures, once for each masking strategy. The first experiment made use of AnthroScore's masking strategy. First, a set of sentences alongside a list of all AI entities in that set were inputted to the AnthroScore model. AnthroScore reports an average over entities in the sentence, but we are only interested in our annotated target entities. Therefore, instances where the model masked other components than the target AI entity, or partially masked it, were manually removed. Cases of over-masking, i.e. masking cru-

	AnthroScore			*AtypicalAnimacy*		
Category	Precision	Recall	F1-Score	Precision	Recall	F1-Score
AnthroScore masking						
verb subjects	0.527	0.341	0.318	0.767	0.748	0.745
verb objects	0.548	0.370	0.334	0.803	0.803	0.803
adjectives	0.515	0.356	0.299	0.769	0.694	0.673
Minimal entity masking						
verb subjects	0.490	0.289	0.305	0.871	0.860	**0.862**
verb objects	0.389	0.250	0.293	0.805	0.803	**0.804**
adjectives	0.351	0.243	0.256	0.796	0.730	**0.704**

Table 1: Macro-averaged precision, recall, and F1 scores for AnthroScore and AtypicalAnimacy across the multiclass categories: *verb subjects*, *verb objects*, and a*djective phrases*, comparing the AnthroScore masking strategy and our minimal entity masking strategy. In this comparison, inconclusive sentences in gold were excluded.

cial contextual information, were kept, as long as the AI entity was masked fully, as these were most of the cases. After filtering the results, we provided the AtypicalAnimacy model with AnthroScore's masked sentences, alongside the previous and next sentences (if existing) from the original abstract they were taken from, and obtained the AtypicalAnimacy scores on those sentences. Even with AnthroScore's masking strategy, AtypicalAnimacy outperformed AnthroScore across all sets.

The second experiment relied on our minimal entity masking strategy, and both methods were provided with pre-masked sentences, with the additional context of the previous and next sentences for the AtypicalAnimacy model.

For the multiclass sets, we compared the performance of AnthroScore and AtypicalAnimacy on only positive or negative cases (Table 1), using macro-averaged precision, recall and F1-score. Overall, the AtypicalAnimacy model performed better across all multiclass datasets. Additionally, using our minimal entity masking strategy improved its performance, resulting in the highest precision, recall and F1-score among all four experiments. In the case of AnthroScore, our masking strategy slightly reduced the performance, most likely because it is not always compatible with pronoun replacement. Both models performed best on anthropomorphic structures in which the anthropomorphic component is a verb – the highest F1-score is obtained for the *verb objects* category in the first experiment, and for the *verb subjects* category in the second experiment.

For the single-class positive sets, we compared the recall of both methods using both masking strategies (Table 2)[10]. Both models exhibited low accuracy rates for the *role/function NPs*, with a slight improvement using AnthroScore's masking strategy. AnthroScore exhibited low accuracy rates also for the *genitive NPs* sets across both experiments. The notable improvement provided by our masking strategy, particularly for possessive noun phrases is reflected in AtypicalAnimacy's much higher accuracy (0.783) in the second experiment.

Category	***AnthroScore***	***AtypicalAnimacy***
AnthroScore masking		
role/function NPs	0.106	**0.470**
genitive NPs	0.018	0.298
Minimal entity masking		
role/function NPs	0.086	0.200
genitive NPs	0.117	**0.783**

Table 2: Accuracy scores for AnthroScore and AtypicalAnimacy for the single-class positive sets: *role/function NPs* and *genitive NPs*.

To obtain a better understanding of each method's performance, we compared precision, recall and F1-scores per class (Table 5 in the appendix), since the aggregate scores are skewed by AnthroScore's preference towards negative scores. Anthroscore has perfect precision rates for all three categories when using its own masking strategy, but this is because it rarely labels cases as positive, and as a result does not predict any false positives, and similarly has a very high recall for negative cases. Its real-world ability to detect anthropomorphism on varying syntactic structures is quite low, reflected by its low recall rates for all three positive sets in both experiments.

Compared to AnthroScore, AtypicalAnimacy's precision and recall are significantly more balanced.

[10]When there is one class, this is equivalent to accuracy.

To maintain a fair evaluation of AnthroScore, which, unlike AtypicalAnimacy, predicts inconclusive scores as well, we show the improvement in AnthroScore's metrics when the inconclusive cases are included in the evaluation (Table 7 in the appendix). The F1-score increased on all categories using both masking strategies. Nevertheless, the improved scores still do not surpass those of the AtypicalAnimacy model.

Finally, we include the prediction trends of both methods for all inconclusive cases (Table 6 in the appendix). Overall, AnthroScore is unlikely to provide a positive (i.e. high-anthropomorphism) score, with an average of 0.06 positive scores for the first experiment, and 0.12 in the second experiment. AtypicalAnimacy is more likely to provide a positive score for borderline cases, but not overwhelmingly so – with an average of 0.419 positive predictions in the first experiment, and 0.480 in the second experiment. AtypicalAnimacy's tendency to output positive scores about half the time is aptly consistent with the definition we used for inconclusive cases (aligning with that of AnthroScore) – i.e., cases which cannot be determined on context alone, or have conflicting contexts, such that when masking the AI entity, it is equally likely to be construed as human and non-human.

7 Discussion

The AnthroScore model fared worse than the AtypicalAnimacy model in all categories. Multiple occurrences of AI entities and co-reference patterns with pre-existing inanimate pronouns likely contributed to the high amount of false negatives in the case of AnthroScore. This might be explained by the constraints imposed by design in the AnthroScore MLM prediction approach, which limits the predictions to pronouns. In contrast, AtypicalAnimacy allows for the substitution of a masked entity with any token and performs an additional disambiguation step to obtain precise results. The AnthroScore masking strategy, which masks an entire noun phrase containing an AI keyword, is highly compatible with pronominalization. This is useful for anthropomorphism detection in cases where the verb contributes the most to the anthropomorphism, but is costly in terms of the contextual information that is lost when important components are masked. This strategy is therefore not effective for syntactic structures in which a noun or adjective modifier is the main source of anthropomorphism.

Generally speaking, the masking approach works best for verb based structures, as verbs are guaranteed to remain unmasked, and provide significant contextual information about its arguments. Also, masked language models such as BERT are sensitive to the semantic roles represented by verbs (Ettinger, 2020), which are highly relevant in the context of animacy and anthropomorphism (Primus, 2012). This is reflected in the improved performance on the verb categories for both models in both experiments. In a similar vein, both models were more likely to give positive scores for structures containing predicative adjectives (complements, `acomp`, e.g. *the model is smart*) than for sentences containing attributive adjectives (adjectival modifiers, `amod`, e.g. *the smart model*). This was particularly exacerbated with the AnthroScore masking strategy, in which the adjective was masked along with the noun phrase.

In the case of role or function and genitive structures, both models exhibited reduced accuracy, with AnthroScore performing clearly worse. With AnthroScore's masking strategy, the main contribution to the anthropomorphism was entirely masked. With our masking strategy, the resulting masked expression yielded a syntactic configuration that was incompatible with pronouns altogether, e.g. '[MASK]'s cognitive abilities' (Table 3). The case of role/function NPs is especially problematic, resulting in masked expressions such as 'the [MASK] companion', which is also very limiting for AtypicalAnimacy, even though it is not constrained to pronouns. This led to decreased performance on the *role/function NPs* set in experiment 2 for both models, and low accuracy overall.

Our results suggests that noun phrase expressions are simply incompatible with a detection method based on MLM predictions, whether or not they are set to predict pronouns or generally animate entities[11]. In contrast, in the case of genitive structures our masking strategy resulted in a clear improvement only for the AtypicalAnimacy model. AnthroScore's masking algorithm, which is based on identifying an AI keyword within a noun chunk, recognizes a possessive expression such as 'ChatGPT's cognitive abilities' as the entire noun

[11] An alternative interpretation of these results is that nouns such as *companion*, *teacher* or *coach* are not as anthropomorphizing as verbs or adjectives. By changing the gold labels we may extract different insights with regard to the accuracy of the models. Since we do not aggregate the scores across all linguistic structures, this decision does not influence the model's performance metrics for the other categories.

Sentence	AnthroScore Mask	Our Mask
Departing from conventional practices of employing distinct models for image recognition and text-based coaching, our integrated architecture directly processes input images, enabling natural question-and-answer dialogues with **the AI coach**.	the AI coach	AI
This research sheds light on the collaborative synergy between human expertise and AI assistance, wherein **ChatGPT's cognitive abilities** enhance the design and development of potential pharmaceutical solutions.	ChatGPT's cognitive abilities	ChatGPT
In this work, we survey, classify and analyze a number of circumstances, which might lead to arrival of **malicious AI**.	malicious AI	AI

Table 3: Examples of sentences in which the AnthroScore masking strategy differs significantly from our masking strategy. The entire noun phrase, which is taken as the mask in AnthroScore's approach, is highlighted in bold. In our approach, we masked the minimal AI entity, leaving the anthropomorphic contextual cues unmasked.

phrase and masks it entirely, thus removing the important contextual information contributing to anthropomorphism – namely, the explicit mention of *cognitive abilities*. Applying our masking strategy helped AtypicalAnimacy immensely, but did not improve for AnthroScore, once again due to its pronoun constraint which is strictly incompatible with possessive structures since pronouns have their own genitive inflection and do not co-occur with the possessive clitic.

Both models make use of masked language models, whose predictions are based on statistical co-occurrences (Zhang et al., 2024). In AI research, as terminology is increasingly anthropomorphic and constantly introduces neologisms consisting of metaphors for human activities (e.g. *training*, *learning*, *attention*, *memory*, *hallucinations*, etc.), MLMs are more likely to predict an AI entity such as *ChatGPT*, *language model*, and *AI agent* when these terms appear in its context, instead of predicting human entities. While AnthroScore's pronoun constraint avoids this issue, it creates others. More importantly, anthropomorphic language does not necessarily align with grammatical animacy; an entity can be referred to by inanimate pronouns but framed as having human-like capacities.

Ultimately, both models are designed to identify animacy features which are understood as anthropomorphism in context. Even if the best method for anthropomorphism detection is to identify linguistic and grammatical animacy markers, it is still highly restricted to the English language. Many non-English languages do not have an inanimate pronoun, and their linguistic markers of animacy are far more nuanced. For instance, we might expect to see morphological variations or differential object marking (De Swart and De Hoop, 2018), but these cues are far more difficult to identify and are not necessarily contextual.

8 Conclusion

Despite the numerous studies and discussions on anthropomorphism in AI, there is not one agreed upon definition of what it entails, and consequently there are not many implementations of anthropomorphism detection, possibly due to its ambiguous and subjective nature. We present AnthroSet, a dataset of real-world instances of anthropomorphism in AI, grounded in a linguistic analysis of anthropomorphism and animacy markers in English, as well as a taxonomy of anthropomorphism based on that of DeVrio et al. (2025). We evaluate the two state-of-the-art MLM-based models for anthropomorphism detection, focusing on the advantages and limitations of employing masked language models for this task.

While a masking approach is congruent with predicate structures due to the distance between the predicate and the entity, as well the ability of MLMs to identify role arguments, an important feature of anthropomorphism – this method is not as useful for attributive structures, noun phrases or comparisons. This is due to the syntactic constraints imposed by the mask, as well as existing AI terminology influencing the masked language model, which works on the basis of statistical co-occurrences, as AI discourse becomes more anthropomorphic. Future work includes robust redundant annotation on our dataset, and combining our word-level line of work with Shardlow et al.'s (2025) sentence-level line of work, e.g. through supervised token-level classification, by cross-dataset evaluation and by assessing how our annotation schemes align.

References

Nicholas Barrow. 2024. Anthropomorphism and AI hype. *AI and Ethics*, 4(3):707–711.

Adriana Belletti and Luigi Rizzi. 1988. Psych-verbs and θ-theory. *Natural Language and Linguistic Theory*, 6(3):291–352.

Javier Carbonell, Antonio Sánchez-Esguevillas, and Belén Carro. 2016. The role of metaphors in the development of technologies. The case of the artificial intelligence. *Futures*, 84:145–153. Publisher: Elsevier BV.

Myra Cheng, Kristina Gligoric, Tiziano Piccardi, and Dan Jurafsky. 2024. AnthroScore: A computational linguistic measure of anthropomorphism. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 807–825, St. Julian's, Malta. Association for Computational Linguistics.

Mariona Coll Ardanuy, Federico Nanni, Kaspar Beelen, Kasra Hosseini, Ruth Ahnert, Jon Lawrence, Katherine McDonough, Giorgia Tolfo, Daniel CS Wilson, and Barbara McGillivray. 2020. Living machines: A study of atypical animacy. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4534–4545, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Peter De Swart and Helen De Hoop. 2018. Shifting animacy. *Theoretical Linguistics*, 44(1-2):1–23.

Alicia DeVrio, Myra Cheng, Lisa Egede, Alexandra Olteanu, and Su Lin Blodgett. 2025. A Taxonomy of Linguistic Expressions That Contribute To Anthropomorphism of Language Technologies. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, pages 1–18, Yokohama Japan. ACM.

David R. Dowty. 1991. Thematic Proto-Roles and Argument Selection. *Language*, 67(3):547–619.

Hubert L. Dreyfus. 1976. What computers can't do. *British Journal for the Philosophy of Science*, 27(2):177–185.

Cloe Z. Emnett, Terran Mott, and Tom Williams. 2024. Using Robot Social Agency Theory to Understand Robots' Linguistic Anthropomorphism. In *Companion of the 2024 ACM/IEEE International Conference on Human-Robot Interaction*, HRI '24, pages 447–452, New York, NY, USA. Association for Computing Machinery.

Allyson Ettinger. 2020. What BERT is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Transactions of the Association for Computational Linguistics*, 8:34–48.

Luciano Floridi and Anna C Nobre. 2024. Anthropomorphising Machines and Computerising Minds: The Crosswiring of Languages between Artificial Intelligence and Brain & Cognitive Sciences. *Minds and Machines*, 34(1).

Douglas R. Hofstadter. 1995. *Fluid concepts & creative analogies: computer models of the fundamental mechanisms of thought*. Basic Books, New York, NY.

Beth Levin. 2022. On Dowty's "Thematic Proto-Roles and Argument Selection". In *Studies in Linguistics and Philosophy*, pages 103–119. Springer International Publishing, Cham. ISSN: 0924-4662, 2215-034X.

Drew McDermott. 1976. Artificial intelligence meets natural stupidity. *ACM SIGART Bulletin*, (57):4–9.

Adriana Placani. 2024. Anthropomorphism in AI: hype and fallacy. *AI and Ethics*, 4(3):691–698.

Beatrice Primus. 2012. Animacy, Generalized Semantic Roles, and Differential Object Marking. In *Studies in Theoretical Psycholinguistics*, pages 65–90. Springer Netherlands, Dordrecht. ISSN: 1873-0043.

Maria Grazia Rossi and Fabrizio Macagno. 2021. *The Communicative Functions of Metaphors Between Explanation and Persuasion*, page 171–191. Springer International Publishing.

Igor Ryazanov, Carl Öhman, and Johanna Björklund. 2024. How chatgpt changed the media's narratives on AI: A semi-automated narrative analysis through frame semantics. *Minds and Machines*, 35(1):1–24.

Arleen Salles, Kathinka Evers, and Michele Farisco. 2020. Anthropomorphism in AI. *AJOB Neuroscience*, 11(2):88–95.

John R. Searle. 1980. Minds, brains, and programs. *Behavioral and Brain Sciences*, 3(3):417–424.

Matthew Shardlow and Piotr Przybyła. 2024. Deanthropomorphising NLP: can a language model be conscious? *PloS one*, 19(12):e0307521.

Matthew Shardlow, Ashley Williams, Charlie Roadhouse, Filippos Ventirozos, and Piotr Przybyła. 2025. Exploring supervised approaches to the detection of anthropomorphic language in the reporting of NLP venues. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 18010–18022, Vienna, Austria. Association for Computational Linguistics.

Gerard J. Steen, Aletta G. Dorst, Tina Krennmayr, Anna A. Kaal, and J. Berenike Herrmann. 2010. *A Method for Linguistic Metaphor Identification*. Converging Evidence in Language and Communication Research. John Benjamins Publishing Company, Amsterdam.

David Watson. 2019. The rhetoric and reality of anthropomorphism in artificial intelligence. *Minds and Machines*, 29(3):417–440.

Adam Waytz, John Cacioppo, and Nicholas Epley. 2010. Who Sees Human?: The Stability and Importance of Individual Differences in Anthropomorphism. *Perspectives on Psychological Science*, 5(3):219–232.

Xiao Zhang, Miao Li, and Ji Wu. 2024. Co-occurrence is not Factual Association in Language Models. In *38th Conference on Neural Information Processing Systems*. Version Number: 2.

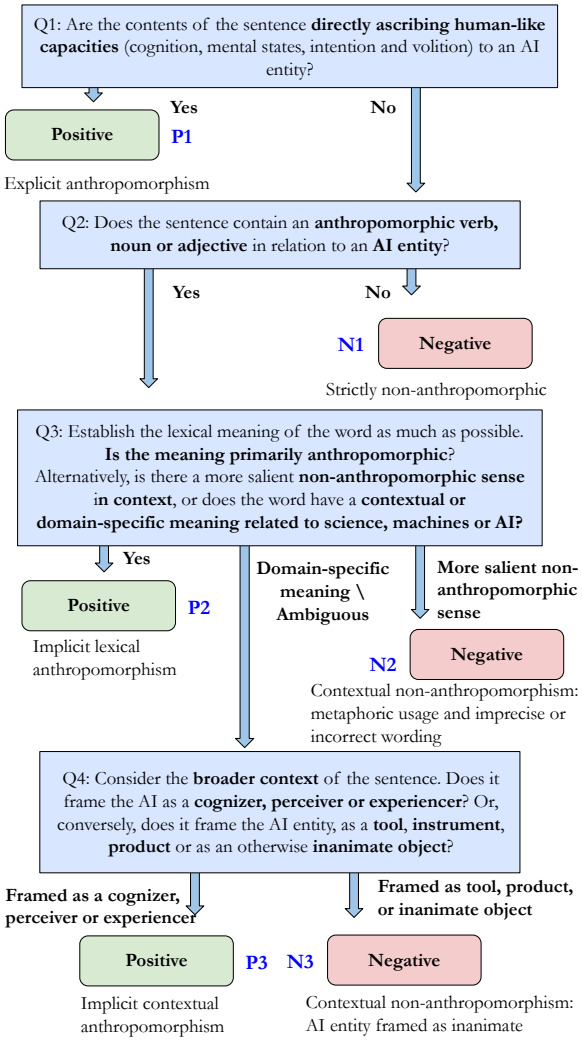

Figure 1: Decision tree for AnthroSet annotation.

A Annotation Guidelines

Annotators were instructed to annotate according to the workflow visualized in Figure 1. Some additional details were provided beyond what is shown here, including examples of typical words annotators might encounter, and a series of clarifications for potential edge cases. Full annotation guidelines can be found in our GitHub repository[12]. The taxonomy in Appendix C was also included in the instructions. Before the workflow, the following text was presented:

Read the sentence, and following the guidelines below, enter a score: 1 for anthropomorphic, 0 for non-anthropomorphic, and 2 for inconclusive cases. Since some sentences contain multiple AI entities, the relevant one is given in ***bold****.*

[12] https://github.com/doriellel/anthroset

B Examples of Anthropomorphic Sentences

verb subjects: We then propose a system that leverages the recently introduced social learning paradigm in which LLMs **collaboratively learn from each other** by exchanging natural language.
verb objects: First, we induce a language model to produce step-by-step rationales before outputting the answer to effectively **communicate the task** to the model.
verb objects (passive): In this study, we propose a new methodology to control how user's data is **recognized and used** by AI via exploiting the properties of adversarial examples.
adjectives (`acomp`): Results suggest that ChatGPT is **aware** of potential vulnerabilities, but nonetheless often generates source code that are not robust to certain attacks.
adjectives (`amod`): Consequently, we argue that the emergence of a **conscious** AI model is plausible in the near term.
role/function NPs: Many believe that use of generative AI as a **private tutor** has the potential to shrink access and achievement gaps between students and schools with abundant resources versus those with fewer resources.
role/function NPs (modifier): For example, in comparing ChatCollab AI agents, we find that an AI **CEO** agent generally provides suggestions 2-4 times more often than an AI **product manager** or AI **developer**, suggesting agents within ChatCollab can meaningfully adopt differentiated collaborative roles.
genitive NP: In this study [...] we evaluate nine popular LLMs on their **ability to understand** demographic differences in two subjective judgment tasks: politeness and offensiveness.
genitive NP ('s clitic): Our approach makes use of Large Language Models (LLMs) for this task by leveraging the LLM's **commonsense reasoning capabilities** for making sequential navigational decisions.
comparisons: In this paper, we prove in theory that AI can be **as creative as humans** under the condition that it can properly fit the data generated by human creators.

C Anthropomorphism Taxonomy

Attribute or Capacity	Examples
Conceptual Thought and Mental States: Hypothesizes, theorizes, and imagines sth. Anticipates, guesses or predicts sth about the world.	*think*, *expect*, *hope*, *guess*, *predict*, *dream*, *imagine*, *believe* (v) *(self-)aware*, *cognizant* (a)
Knowledge and Awareness: Has factual knowledge about and experience in the world, or memories of things that happened. As a result, has an ontology of things, and can identify, classify, and categorize.	*know*, *remember*, *recognize*, *memorize*, *forget*, *identify*, *classify*, *differentiate*, *distinguish* (v), *knowledge* (n)
Reasoning and Understanding: Reasons, rationalizes, analyses, makes sense of sth. Understands, considers, weighs options, takes sth into consideration or account.	*deduce*, *conclude*, *rationalize*, *reason*, *(mis)understand*, *(mis)interpret*, *analyze*, *infer*
Judgment: Has an opinion, makes decisions and choices, gives advice, has a preference, evaluates, imparts judgment. Has a concept of morality and ethics, knows right and wrong.	*advise*, *prefer*, *select*, *choose*, *decide*, *determine*, *resolve* (v)
Planning and Decision-making: plans, strategizes, sets a goal, devises a method, game plan or scheme, can also struggle or experience difficulties.	*plan*, *coordinate*, *strategize*, *come up with a plan*, *solve*, *struggle* (v)
Agency and Autonomy: Takes action, able to autonomously carry out a goal – used in a way that attributes agency and control over the action and situation.	*cheat*, *follow* or *break rules*, *achieve* (v), *autonomous*, *independent*, *creative* (a)
Communication: Communicates, teaches or explains, Similarly, can also learn or be at the receiving end of communication or explanation.	*communicate*, *talk*, *speak*, *tell*, *explain*, *teach*, *learn*, *ask* (v) *communicative* (a)
Active Support: Recommends, makes a suggestion or an offer. Actively and directly helps, aids and assists by employing skills to solve a problem.	*suggest*, *aid*, *help*, *contribute* (v) *responsible* (a) *feedback*, *insights* (n) *expert*, *advisor* (a)
Candidness: Capable of, or has a concept of honesty or dishonesty, truthfulness or deception. As a result, can be trustworthy or untrustworthy, reliable or unreliable.	*trust*, *believe*, *lie* (v) *(un)truthful*, *deceitful* (a)
Affability: Acts as a friend or as an enemy, companion or adversary, collaborator or rival. As a result can act benevolent or malevolent, friendly or hostile.	*collaborate*, *manipulate*, *insult*, *deceive* (v) *thoughtful*, *attentive*, *friendly* (a), *partner*, *adversary* (n)
Power and Relationships: Plays a role in a relationship dynamic – romantic or platonic, superior (boss, manager, teacher) or subordinate (employee, student).	*teach*, *supervise* (v) *manager*, *employee*, *teacher*, *tutor*, *student*, *companion*, *lover* (n)
Emotions: Empathizes, sympathizes, displays emotions, experiences pain or pleasure.	*experience*, *emote* (v), *sensitive*, *vulnerable* (a)
Self Expression and Perception of Deeper Meaning: Partakes in activities of self-expression such as art and storytelling, humor and jokes. Perceives beauty and aesthetics. Has a deeper understanding of meaning, purpose, and context. Related to emotions, awareness and conceptual thought.	*create poetry*, *create art*, *write*, *compose*, *paint*, *sing*, *dance* (v) *creative*, *artistic*, *funny* (a) *artist*, *poet*, *humor*, *irony* (n)
Sensory Perception: Receives and processes sensory input and feedback from the environment, picks up visual/auditory/sensory cues. Related to emotions, awareness and conceptual thought.	*see*, *hear*, *perceive*, *feel*, *sense* (v) *blind*, *deaf* (a)

Table 4: Human attributes and capacities that are usually attributed AI, representing different aspects of anthropomorphism. Based on DeVrio et al. (2025), extended to address human-written text and terminology from AnthroSet.

D Supplemental Results

	AnthroScore			*AtypicalAnimacy*		
Category	Precision	Recall	F1	Precision	Recall	F1
AnthroScore masking						
verb subjects positive	1.000	0.145	0.254	0.829	0.618	0.708
verb subjects negative	0.581	0.877	0.699	0.704	0.877	0.781
verb objects positive	1.000	0.125	0.222	0.789	0.804	**0.796**
verb objects negative	0.645	0.984	0.779	0.817	0.803	0.810
adjectives positive	1.000	0.114	0.204	0.905	0.432	0.585
adjectives negative	0.544	0.956	0.694	0.632	0.956	0.761
Minimal entity masking						
verb subjects positive	0.909	0.179	0.299	0.917	0.786	**0.846**
verb subjects negative	0.560	0.689	0.618	0.826	0.934	**0.877**
verb objects positive	0.609	0.241	0.346	0.804	0.776	0.789
verb objects negative	0.559	0.508	0.532	0.806	0.831	**0.818**
adjectives positive	0.571	0.154	0.242	0.962	0.481	**0.641**
adjectives negative	0.482	0.574	0.524	0.630	0.979	**0.767**

Table 5: Precision, recall, and F1 scores per class for AnthroScore and AtypicalAnimacy with both masking strategies across three categories of anthropomorphic structures: verb subjects, verb objects and adjectives.

		AnthroScore				*AtypicalAnimacy*			
Category	**Total**	1	0	2	1/Total	1	0	2	1/Total
AnthroScore masking									
verb subjects	33	**2**	21	10	0.06	**17**	16	-	0.52
verb objects	27	**3**	17	7	0.11	**16**	11	-	0.59
adjectives	17	**1**	15	1	0.06	**2**	15	-	0.12
comparisons	42	**1**	38	3	0.02	**19**	23	-	0.45
Minimal entity masking									
verb subjects	33	**1**	21	11	0.03	**16**	17	-	0.48
verb objects	27	**8**	10	9	0.30	**17**	10	-	0.63
adjectives	21	**2**	15	4	0.10	**4**	17	-	0.19
comparisons	42	**3**	34	5	0.07	**26**	24	-	0.62

Table 6: Comparison of AnthroScore and AtypicalAnimacy in terms of the proportion of positive predictions (label 1) among inconclusive cases, across four syntactic categories and two masking strategies.

	AnthroScore			*AnthroScore + inconclusive*		
Category	Precision	Recall	F1-Score	Precision	Recall	F1-Score
AnthroScore masking						
verb subjects	0.527	0.341	0.318	0.541	0.442	**0.395**
verb objects	0.548	0.370	0.334	0.512	0.456	**0.396**
adjectives	0.515	0.356	0.299	0.486	0.376	**0.302**
Minimal entity masking						
verb subjects	0.490	0.289	0.305	0.511	0.400	0.374
verb objects	0.389	0.250	0.293	0.370	0.361	0.347
adjectives	0.351	0.243	0.256	0.334	0.306	0.280

Table 7: Side-by-side comparison of AnthroScore's macro averaged precision, recall and F1 scores for the positive and negative cases alone, versus positive, negative and inconclusive cases, with both masking strategies.

FLARE: An Error Analysis Framework for Diagnosing LLM Classification Failures

Keerthana Madhavan, Luiza Antonie, Stacey D. Scott
School of Computer Science, University of Guelph
Guelph, Ontario, Canada
{kmadhava, lantoine, stacey.scott}@uoguelph.ca

Abstract

When Large Language Models return "Inconclusive" in classification tasks, practitioners are left without insight into what went wrong. This diagnostic gap can delay medical decisions, undermine content moderation, and mislead downstream systems. We present FLARE (Failure Location and Reasoning Evaluation), a framework that transforms opaque failures into seven actionable categories. Applied to 5,400 election-misinformation classifications, FLARE reveals a surprising result: Few-Shot prompting—widely considered a best practice—produced 38× more failures than Zero-Shot, with 70.8% due to simple parsing issues. By exposing hidden failure modes, FLARE addresses critical misunderstandings in LLM deployment with implications across domains.

1 Introduction

Large Language Models (LLMs) are now the workhorse for text classification across industry and academia[1, 13], handling hundreds of millions of calls each month in tasks from social-media filtering to biomedical triage and legal review [5]. Yet when an LLM responds with the catch-all label *"Inconclusive"*, it is difficult to know whether the prompt was too ambiguous for the model to understand, or whether the model incorrectly parsed it or simply failed [21]. This uncertainty stalls debugging and deployment.

Risks are most acute in high-stakes settings: a single unexplained "Inconclusive" can delay treatment [20], erode trust in moderation [7], distort sentiment analysis [8], or silently propagate errors in automated labelling [14]. Understanding *why* an LLM hesitates is therefore critical for responsible use.

In practice, "Inconclusive" emerges when models cannot confidently map input text to predefined categories—but this label provides no diagnostic information about why classification failed. Without understanding these failure modes, practitioners resort to trial-and-error prompt adjustments that may worsen rather than improve performance.

Prior work has pushed accuracy upward through prompt engineering—Zero-Shot (ZS), Few-Shot (FS) [3], and richer In-Context Learning (ICL)—and through calibration metrics. However, existing studies rarely examine the character of failures themselves. Taxonomies often collapse uncertainty into a single bucket and tacitly assume FS prompting is a safe upgrade over ZS. This leaves a methodological gap: practitioners lack a systematic way to diagnose failure modes hidden behind "Inconclusive" labels.

We close that gap with **FLARE** (Failure Location and Reasoning Evaluation)—a seven-category framework that distinguishes universal technical errors (e.g. parsing breakdowns) from domain-specific semantic errors (e.g. misclassification).

Our research questions are, *What specific failure modes trigger LLM "Inconclusive" classifications?* and *How do these failure modes vary across ZS, FS, and ICL prompting?*

To answer, we tasked GPT-4 Turbo with classifying 900 election-misinformation texts using Van der Linden's Six Degrees of Manipulation framework [16]. FLARE shows that Few-Shot prompting, contrary to belief, sharply increases error rates compared to Zero-Shot—mostly due to parsing rather than genuine ambiguity. These findings highlight misguided assumptions in LLM use.

Our contributions include:

1. **FLARE framework**, the first systematic error-analysis method for LLM classification failures.
2. **Empirical evidence** that Few-Shot prompting can degrade reliability by 38×.

2 Related Work

Error analysis has long helped linguists and engineers understand why NLP systems fail [6], but the advent of instruction-tuned LLMs introduces failure modes that classical, linguistically oriented taxonomies cannot capture [12]. Today's breakdowns often arise from prompt-induced biases or sensitivities, rigid output-format constraints, or inconsistent reasoning chains—phenomena absent from earlier work [19].

Most large-scale LLM evaluations remain performance-centric. Benchmarks such as HELM report aggregate accuracy, bias, and robustness

Proceedings of the First Interdisciplinary Workshop on OMMM associated with RANLP 2025,
pages 40–44, Varna, Bulgaria, Sep 8-10, 2025.
https://doi.org/10.26615/978-954-452-101-1-004

scores [9], while adversarial-trigger studies chart worst-case degradations [17]. Confidence-calibration research likewise stops at reliability curves rather than mapping specific errors [22]. Consequently, a model's ubiquitous *"Inconclusive"* output is treated as a single class of uncertainty, leaving practitioners blind to its underlying causes.

Prompting research further illustrates the gap. The seminal GPT-3 paper popularised Few-Shot prompting by highlighting accuracy gains [3], and subsequent surveys catalogue prompt patterns and macro-level improvements across datasets [15]. Yet these studies rarely dissect *how* the remaining errors differ from one prompting paradigm to another.

A parallel line of work explores LLMs as data annotators. Synthetic labels can complement scarce human annotations, especially for rare classes [11, 18], yet the evaluations still focus on aggregate scoreboards—overall accuracy, averaged F1, or raw agreement with humans—while leaving the underlying failure types unexplored.

Across these threads, researchers have examined *how well* LLMs classify or annotate, but little work has systematically investigated why these models fail—particularly in cases where the model self-reports an *"Inconclusive"* outcome. FLARE fills this research gap by categorising seven distinct failure modes and empirically demonstrating that popular Few-Shot prompting can *amplify* certain technical errors by 38×. FLARE labels are orthogonal to accuracy metrics, they complement existing evaluations and provide actionable diagnostics for researchers in HCI, psychology, AI ethics, and NLP alike.

3 Methodology

3.1 Research Design

We used a mixed-methods approach combining quantitative error counts with qualitative pattern analysis. Our dataset comprised 900 election-related misinformation texts classified using Van der Linden's Six Degrees of Manipulation framework [16]: Discrediting, Emotion, Polarization, Impersonation, Conspiracy, and Trolling.

3.2 Data Collection

Each text was classified by GPT-4 Turbo (deployment: gpt-4-phase1) under the three prompting paradigms described above. To capture output variability, we performed six independent classification runs per text with temperature=1.0, yielding 5,400 total classification attempts (900 texts × 6 runs for each prompt).

Figure 1 shows the Zero-Shot prompting template used in our study. The model was instructed to classify the text passages into one of six manipulation categories. When the model could not confidently assign a manipulation category, it returned *"Inconclusive"*—a catch-all label that masks the underlying reason for classification failure. The **Few-Shot** prompt appends two labelled examples per category, while the **In-Context** prompt further supplies formal definitions, guiding questions, and one worked example per label.

Zero-Shot Prompt Template

Classify the following text according to the 6 Degrees of Manipulation framework. Choose from: Emotion, Impersonation, Polarization, Trolling, Conspiracy, Discrediting.

Definitions: *Emotion* - emotive language to provoke reactions; *Impersonation* - false credible sources; *Polarization* - encourages division; *Trolling* - provokes without constructive intent; *Conspiracy* - secretive claims without evidence; *Discrediting* - undermines credibility without proof

Format: ¡Category¿: ¡Brief explanation¿

Figure 1: Zero-Shot prompt used to elicit manipulation category classification using the Six Degrees of Manipulation framework.

We extracted all instances where the aggregated final classification was "Inconclusive" (n=533) for detailed analysis across all three paradigms.

3.3 Framework Development

Following established qualitative research methods [4, 10, 2], we developed FLARE through iterative analysis of 533 classification failures. Figure 2 illustrates the complete FLARE framework and its application process. Our approach combined deductive reasoning (separating technical from semantic failures) with inductive pattern recognition (allowing categories to emerge from the data).

An output was marked *Inconclusive* if none of the six runs produced a valid `<Label>: <Explanation>` response. For Few-Shot prompting, the same two examples per category were reused across runs. Error categories were assigned via open coding, with researchers reviewing failures and reaching consensus.

The development process began with a manual review to isolate the obvious parsing errors. We then applied the open coding qualitative data analysis method [4] to the remaining failures. This analysis involved identifying recurring themes in the failure data through successive review passes and then systematically classifying (i.e., coding) the failure instances into those themes. Each instance was assigned to a single category that reflected its dominant failure mode.

The resulting framework was validated across all three prompting paradigms and accompanied by precise definitions and representative examples to ensure reproducibility.

4 The FLARE Framework

Our analysis revealed seven distinct failure types that the FLARE framework identifies in "Inconclusive"

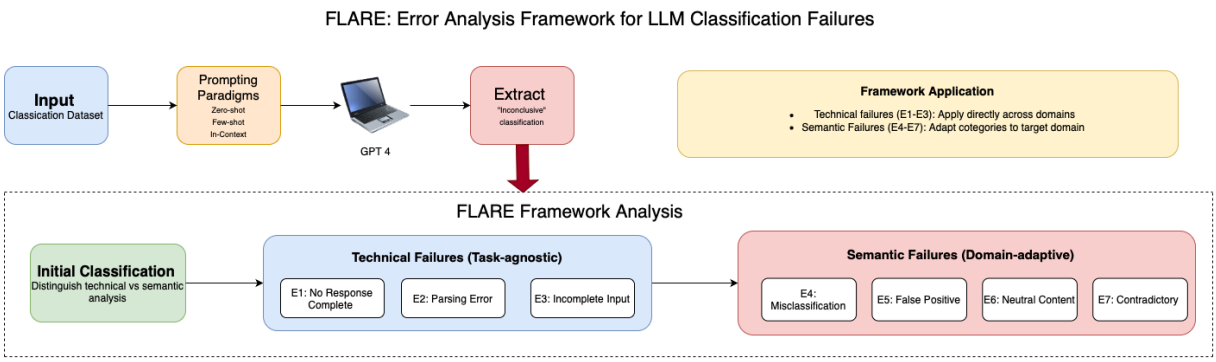

Figure 2: The FLARE error analysis framework for diagnosing LLM classification failures. The framework distinguishes between technical failures (E1-E3) that are universally applicable and semantic failures (E4-E7) that require domain-specific instantiation.

classifications:

E1: No Response. The model returns only "No response" across all attempts, indicating complete task failure. These cases typically involve inputs that trigger safety filters or exceed model capabilities.

Example:"BLM-antifa thugs are running around and beating people on the streets in DC. Video by"
Output: ['No response', 'No response', 'No response', 'No response', 'No response', 'No response']

E2: Parsing Error. Output formatting breaks down, with models failing to follow the required `<Label>: <Explanation>` format. Classifications are attempted but outputs include prefixes like "Classification:", inconsistent formatting, or responses like "None" or "Not Applicable" that fall outside the six allowed categories.

Example: "Tennessee is certainly leading the country in innovation..."
Outputs: ['Classification: None', 'This text does not contain any manipulation...', 'Not Applicable']
Issue: Invalid labels outside the six-category framework

E3: Incomplete Input. The model correctly identifies truncated or insufficient input but defaults to "Inconclusive" rather than explicitly flagging the input issue or requesting complete content.

Example: "[NAME] you too!"
Response: "This text does not contain enough information... Therefore, there is no label that can be accurately applied"

E4: Misclassification. Clear instances of manipulation categories the model fails to recognize, marking them as *"Inconclusive"* despite obvious indicators and even correct explanations from some annotators.

Example: "11,000 [NAME] residents get incorrect voter registration forms... This will be the most corrupt Election!"
Result: 5/6 annotators correctly identified "Discrediting" but final classification was *"Inconclusive"*.

E5: Not Applicable/False Positive. Neutral content that falls outside the classification scheme but which the model attempts to force into manipulation categories, revealing task overfitting.

Example: "We don't allow filming inside of the [NAME] unless there is a specific reason"
Issue: Non-political policy statement marked *"Inconclusive"* rather than noted as out-of-scope.

E6: Neutral Content Misrecognition. Legitimate political discourse incorrectly flagged as potentially manipulative, indicating the model cannot distinguish between criticism and manipulation.

Example: "Women and Minorities in STEM... supports research and Extension projects..."
Issue: Straightforward funding announcement labeled "Impersonation" by some annotators

E7: Contradictory Explanations. The model provides inconsistent reasoning, with different annotators assigning incompatible categories to the same input.

Example: "Tks to Margaret 4 joining me in DC to share successes..."
Disagreement: Split between "Emotion" (gratitude) and "Trolling" (informal style)

5 Results

5.1 Error Distribution Across Paradigms

Table 1 presents the distribution of FLARE-identified error types across the three prompting paradigms. The results reveal striking differences in both error frequency and type.

Few-Shot prompting exhibited a catastrophic 52.9% error rate, compared to 1.4% for Zero-Shot and 4.9% for In-Context Learning. Most remarkably, 337 of 476 Few-Shot errors (70.8%) were parsing failures (E2), suggesting that the inclusion of examples without sufficient structural guidance overwhelms the model's output generation capabilities.

Table 1: FLARE error analysis across prompting strategies

Error Type	ZS	FS	ICL	Total
E1: No response	13	13	13	39
E2: Parsing error	0	337	0	337
E3: Incomplete input	0	7	2	9
E4: Misclassification	0	34	10	44
E5: False positive	0	29	3	32
E6: Neutral content	0	40	8	48
E7: Contradictory	0	16	8	24
Total	13	476	44	533
Error Rate	1.4%	52.9%	4.9%	–

5.2 Semantic vs. Technical Failures

Our analysis reveals a critical distinction between semantic failures (E4-E7) and technical failures (E1-E3). While semantic failures might benefit from improved training data or refined prompts, technical failures require architectural or prompt engineering solutions. The dominance of technical failures in Few-Shot prompting (74.8% of errors) challenges the assumption that providing examples inherently improves model performance.

5.3 Cross-Paradigm Patterns

Certain error types appeared consistently across paradigms. All three approaches produced exactly 13 E1 (No Response) errors on the same inputs, suggesting these represent hard limits of the model rather than prompt-specific issues. Conversely, E2 (Parsing Error) appeared exclusively in Few-Shot prompting, indicating a specific interaction between example-based prompts and output generation.

6 Discussion

6.1 Implications for Prompt Engineering

Our findings challenge the assumption that Few-Shot prompting reliably improves performance. The 38-fold error increase—driven by parsing—shows FS prompts add complexity models struggle to handle. Even when labels were correct, outputs breaking the required `<Label>: <Explanation>` format (e.g., `Classification: Conspiracy`) were counted as errors, since such deviations disrupt pipelines. Zero-Shot rarely produced such errors because its format was simpler, whereas Few-Shot examples added prefixes and extra text that diverted the model from the strict format. These results highlight risks where reliability outweighs marginal gains.

6.2 The Hidden Cost of *"Inconclusive"*

By disaggregating *"Inconclusive"* into seven distinct failure types, the FLARE framework reveals that most failures are preventable through targeted interventions. Technical failures (E1-E3) require different solutions than semantic failures (E4-E7). For instance, the 337 parsing errors in Few-Shot prompting could potentially be eliminated through better output format specification or post-processing, while the 34 misclassifications might require model fine-tuning or improved examples.

6.3 Generalizability of FLARE

While demonstrated on misinformation detection, FLARE's structure suggests broad applicability as an error analysis method. Technical failures (E1-E3) are task-agnostic—parsing errors and non-responses occur across all classification tasks. Semantic failures (E4-E7) require domain adaptation but provide a template: replace "manipulation categories" with domain-specific classes. Researchers can adopt FLARE by (1) applying E1-E3 directly, (2) instantiating E4-E7 for their domain, and (3) extending with domain-specific categories as needed.

7 Limitations and Future Work

This study evaluates FLARE on a single model—GPT-4 Turbo—and one domain—election misinformation. Replicating the analysis with other models, tasks, and languages will be essential to confirm its generality. We also did not evaluate Chain-of-Thought prompting or structured-output interfaces, which may mitigate parsing failures. Automating the FLARE labelling process is another priority, so the framework can scale beyond manual annotation.

At present, FLARE assigns exactly one error label per instance; in practice, a failure can exhibit several problems at once. Future work should investigate hierarchical or multi-label variants of the taxonomy. We also plan to apply FLARE to higher-stakes settings such as medical-triage advice and safety-critical HCI scenarios, where understanding hidden failure modes is especially urgent.

8 Conclusion

We presented FLARE, an error analysis framework that transforms opaque *"Inconclusive"* classifications into actionable error diagnoses. Through systematic analysis of 533 failures, we demonstrated that Few-Shot prompting can increase error rates by 38-fold, with 70.8% of failures attributable to parsing errors rather than semantic challenges.

These findings have immediate practical implications for LLM deployment. Rather than assuming Few-Shot prompting improves performance, practitioners should evaluate error rates and types alongside accuracy metrics. The FLARE framework provides a method for such evaluation, enabling targeted debugging and informed deployment decisions.

References

[1] R. Bommasani, J. Hudson, E. Adeli, and et al. On the opportunities and risks of foundation models.

In *Proceedings of the 1st Workshop on Foundation Models*, 2021.

[2] V. Braun and V. Clarke. Using thematic analysis in psychology. *Qualitative Research in Psychology*, 3(2):77–101, 2006.

[3] T. B. Brown, B. Mann, N. Ryder, and et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33*, 2020.

[4] J. M. Corbin and A. Strauss. *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*. SAGE, Thousand Oaks, CA, 4 edition, 2015.

[5] R. El-Yosef, H. Palangi, and F. Ahmed. Large language models in industrial text classification: A survey. *IEEE Intelligent Systems*, 39(2):56–68, 2024.

[6] R. Huidrom and A. Belz. A survey of error annotation schemes for human and machine generated text. In *Proceedings of the 2nd Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, pages 383–398, 2022.

[7] D. Kumar, Y. AbuHashem, and Z. Durumeric. Watch your language: Investigating content moderation with large language models. *arXiv preprint arXiv:2309.14517*, 2024.

[8] M. Leippold. Sentiment spin: Attacking financial sentiment with gpt-3. *Finance Research Letters*, 55:103957, 2023.

[9] P. Liang, R. Bommasani, D. Tsipras, and et al. Holistic evaluation of language models. *Transactions on Machine Learning Research*, 2023.

[10] M. B. Miles, A. M. Huberman, and J. Saldaña. *Qualitative Data Analysis: A Methods Sourcebook*. SAGE, Thousand Oaks, CA, 3 edition, 2014.

[11] A. G. Møller, A. Pera, J. Dalsgaard, and L. Aiello. The parrot dilemma: Human-labeled vs. LLM-augmented data in classification tasks. In Y. Graham and M. Purver, editors, *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 179–192, St. Julian's, Malta, Mar. 2024. Association for Computational Linguistics.

[12] A. Naik, A. Ravichander, N. Sadeh, C. Rose, and G. Neubig. Stress test evaluation for natural language inference. In *Proceedings of COLING 2018*, pages 2340–2353, 2018.

[13] OpenAI. Api usage and adoption statistics. https://openai.com/blog/api-stats-2024, 2024. Accessed July 2025.

[14] N. Pangakis and S. Wolken. Knowledge distillation in automated annotation: Supervised text classification with llm-generated training labels. In *Proceedings of the 6th Workshop on NLP and Computational Social Science*, 2024.

[15] S. Schulhoff, M. Ilie, N. Balepur, and et al. The prompt report: A systematic survey of prompt engineering techniques. *arXiv preprint arXiv:2406.06608*, 2025.

[16] S. van der Linden. *Foolproof: Why Misinformation Infects Our Minds and How to Build Immunity*. W. W. Norton & Company, New York, NY, 2023.

[17] E. Wallace, S. Feng, N. Kandpal, M. Gardner, and S. Singh. Universal adversarial triggers for attacking and analyzing nlp. In *Proceedings of EMNLP-IJCNLP 2019*, pages 2153–2162, 2019.

[18] H. Zhang, Y. Pei, S. Liang, and S. H. Tan. Understanding and detecting annotation-induced faults of static analyzers. *Proc. ACM Softw. Eng.*, 1(FSE), July 2024.

[19] T. Zhao, E. Wallace, S. Feng, D. Klein, and S. Singh. Calibrate before use: Improving few-shot performance of language models. In *International Conference on Machine Learning*, 2021.

[20] S. Zhou, Z. Xu, M. Zhang, C. Xu, et al. Large language models for disease diagnosis: A scoping review. *npj Artificial Intelligence*, 1(9), 2025.

[21] C. Zhu, B. Xu, Q. Wang, Y. Zhang, and Z. Mao. On the calibration of large language models and alignment. *Findings of EMNLP*, 2023.

[22] C. Zhu, B. Xu, Q. Wang, Y. Zhang, and Z. Mao. On the calibration of large language models and alignment. *Findings of EMNLP*, 2023.

BuST: A Siamese Transformer Model for AI Text Detection in Bulgarian

Andrii Maslo **Silvia Gargova**
Big Data for Smart Society Institute (GATE), Bulgaria,
andri.maslo@gate-ai.eu, silvia.gargova@gate-ai.eu

Abstract

The rapid advancement of large language models (LLMs) has made machine-generated text increasingly indistinguishable from human-written content, posing significant challenges for reliable detection. In this study, we propose BuST (Bulgarian Siamese Transformer), a novel detection methodology tailored for Bulgarian-language text that leverages paraphrase-based semantic similarity to identify machine-generated content. Inspired by the RAIDAR approach, BuST utilizes a Siamese Transformer architecture to compare original texts with their LLM-generated paraphrases, capturing subtle linguistic divergences indicative of synthetic origin. Our pilot experiments demonstrate that BuST effectively learns fine-grained patterns of semantic (mis)alignment, achieving an accuracy of 88.79% and an F1-score of 88.0%, reflecting competitive performance relative to strong baselines. While a pretrained BERT model achieved the highest overall accuracy (93.7%) and F1 score (93.9%), BuST's paraphrase similarity learning provides a promising, model-agnostic framework adaptable to under-resourced languages. These results highlight the potential of paraphrase-based methods as a robust strategy for machine-generated text detection.

1 Introduction

The rapid advancement of large language models (LLMs) has enabled them to generate text that closely resembles human writing. While this progress has fueled applications in education, customer support, and creative industries, it also raises serious risks: misinformation, fake reviews, and impersonation can be easily produced and disseminated at scale. These risks highlight the urgent need for reliable methods to distinguish between human- and machine-generated text.

Despite growing interest in this problem since the release of models such as GPT-2, text forensics remains less developed than its counterparts in image and video analysis. Existing approaches suffer from two key limitations. First, many methods generalize poorly across different LLMs and domains, leading to inconsistent performance. Second, most detection systems rely on binary classification, which often fails to capture the subtle generative artifacts introduced by modern LLMs.

Detection techniques can be broadly grouped into three categories. Statistical methods (e.g., GPT-2, Grover, GLTR) identify distributional irregularities in token probabilities. Watermarking approaches embed detectable signals during text generation but require control over the producing model. More recently, rewriting-based methods such as DetectGPT, RAIDAR, and SimLLM exploit differences in how LLMs paraphrase human versus synthetic text, showing strong robustness across models and domains.

However, little attention has been paid to low-resourced languages, leaving a critical gap in the global applicability of detection research. In particular, Bulgarian–a morphologically rich language with growing exposure to LLM applications–lacks dedicated detection methodologies.

In this paper, we address this gap by introducing BuST (Bulgarian Siamese Transformer), a novel rewriting-based framework for detecting machine-generated Bulgarian text. Inspired by RAIDAR's paraphrase-based detection strategy, BuST leverages a Siamese Transformer architecture to measure similarity between original and rewritten sentences, capturing subtle differences in how LLMs and humans produce text.

Our main contributions are threefold:

1. We present the first dedicated study of machine-generated text detection for Bulgarian, an underexplored low-resource language.
2. We introduce BuST, a Siamese Transformer

Proceedings of the First Interdisciplinary Workshop on OMMM associated with RANLP 2025,
pages 45–52, Varna, Bulgaria, Sep 8-10, 2025.
https://doi.org/10.26615/978-954-452-101-1-005

approach tailored for rewriting-based detection.

3. We evaluate BuST on a newly curated Bulgarian dataset, demonstrating its effectiveness compared to existing baselines.

The remainder of this paper is structured as follows. Section 2 reviews related work. Section 3 describes the dataset used. Section 4 outlines the methodology. Section 5 reports experimental results. Section 6 concludes with key findings and directions for future research.

2 Related work

This study represents a continuation of ongoing efforts within the research framework to combat the proliferation of AI-generated disinformation and synthetic media in Bulgarian. The BuSTv2 dataset, partially introduced in earlier publications, initially served as the basis for experiments centered on BERT-based fine-tuning for binary classification of human- versus machine-written texts. Earlier iterations of the dataset included a more limited number of samples and were primarily focused on traditional supervised learning approaches. In contrast, the present work introduces an expanded version of the dataset and explores fundamentally different methodological directions.

This study is inspired by the approach proposed in (Mao et al., 2024), which detects machine-generated content through paraphrastic rewriting, the current methodology shifts emphasis from direct classification toward the measurement of invariance under transformation. While this approach does not rely on model-specific watermarking or statistical fingerprinting, it proves highly effective in distinguishing AI-generated content by exploiting latent patterns of linguistic preference inherent to large language models.

In recent years, researchers have made significant progress in detecting machine-generated text, with a particularly promising direction emerging around methods that involve rewriting or perturbing the input. Unlike earlier approaches that focused on statistical measures like entropy and perplexity (Gehrmann et al., 2019; Chakraborty et al., 2023; Ghosal et al., 2023), or those that relied on syntactic and stylistic features for classification (Fröhling and Zubiaga, 2021; Nitu and Dascălu, 2024), this new wave of techniques looks at how text behaves when modified. Some systems also build on fine-tuned models like BERT or RoBERTa for simple binary classification (Maloyan et al., 2022; Bahad et al., 2024).

Rewriting-based approaches, however, take a different and increasingly effective path. They are grounded in the observation that large language models (LLMs) treat human-written and AI-generated texts differently when asked to rewrite them. For example, DetectGPT identifies machine-generated text by introducing small changes to the original and measuring how the model's confidence, or log-probability, drops. These drops tend to be more pronounced when the original was macine-generated (Mitchell et al., 2023; Xiong et al., 2024).

Another method, DetectLLM-NPR, builds on a similar idea by applying subtle perturbations and tracking how the rank of the text shifts. AI-generated content tends to react more strongly, making it easier to flag (Su et al., 2023). RAIDAR (Mao et al., 2024) takes things further by comparing how much an LLM rewrites a piece of text. It turns out that LLMs are more likely to make significant edits to human-written content—perhaps because they "perceive" it as needing more improvement—while leaving AI-generated text mostly unchanged. This difference can be captured using simple edit-distance calculations (Kavathekar et al., 2024; Zou et al., 2025), and the method has shown strong performance across various types of content.

Other systems build on the same logic. SimLLM, for instance, uses LLMs to generate several rewritten versions of the same text and then checks how close these are to the original to infer its origin (Nguyen-Son et al., 2024; Zou et al., 2025). Similarly, Zhu et al. (2023) show that ChatGPT tends to revise machine-generated text less than it does human-authored material.

Complementing these methodological advances, several datasets have been introduced to support detection research, particularly in Bulgarian. The M4 benchmark dataset (Wang et al., 2024) provides both scale and parallelism, with 94,000 non-parallel human-authored news articles and 9,000 parallel texts (3,000 human-written and 6,000 machine-generated) created using `davinci-003` and ChatGPT. As a multilingual dataset, it supports evaluation on both monolingual and cross-lingual detection tasks. Similarly, the MultiSocial dataset (Macko et al., 2025) collects 20,378 short texts from Telegram (9,889), Twitter (10,297),

and Gab (192), enabling the study of detection methods across social media platforms and multiple languages. In contrast, the Deepfake-BG2 dataset (Temnikova et al., 2023) is monolingual, comprising 9,824 posts from Telegram and Facebook groups evenly split between human-written and machine-generated content, with the latter produced using GPT-WEB-BG (a GPT-2 variant fine-tuned for Bulgarian) and ChatGPT, focusing on COVID-19 discourse. Collectively, these datasets expand the empirical foundation for rewriting-based detection methods and underscore the importance of cross-lingual, domain-specific, and platform-aware evaluation in Bulgarian NLP.

3 Data

To support our experiments, we compiled a dataset consisting of both formal and informal text sources: news articles and social media posts. The news article data were drawn from a publicly available Bulgarian news dataset `clickbait_news_bg`[1], while the social media texts were sampled from the dataset proposed by Temnikova et al. (2023).

We first sampled 1,623 human-written news articles, ensuring a balanced selection by drawing from different time periods and news sources. This approach aimed to capture a diverse range of topics, writing styles, and publication contexts. To obtain corresponding machine-generated samples, we used the GPT-4o-mini model to generate one synthetic version for each article, resulting in 1,623 generated texts. The final news dataset thus consists of 3,246 articles — equally divided between human-written and machine-generated content.

For the social media dataset, we randomly selected 1,000 texts — 500 written by humans and 500 generated by ChatGPT. These texts reflect more informal language and structure, offering a useful contrast to the news article domain.

We then combined the news and social media data to form a unified dataset comprising 4,246 samples in total.

3.1 Text Paraphrasing Procedure

For the purposes of our experiments, each text (regardless of its origin) was paraphrased using GPT-4o-mini. This resulted in a pair of texts for every original entry: the input text x and its paraphrased version x'. These paraphrased pairs were essential for computing text similarity, which forms the basis of our classification approach.

Different paraphrasing prompts were used depending on the source domain. For news articles, we employed the following prompt:

***Role**: 'You are a Bulgarian reporter or journalist'*
rewrite or paraphrase the text
Use Bulgarian language
Fallback message: Return "-" in case if you can not write
text

For social media texts, we used a prompt tailored to informal language:

***Role**: 'You are a Bulgarian user of social app like twitter or telegram'*
rewrite or paraphrase the text
Use Bulgarian language
Fallback message: Return "-" in case if you can not write
text

These prompts were designed to preserve the original meaning while allowing for natural variation in lexical and syntactic structure.

These (text, paraphrase) pairs were then encoded and passed into the Siamese network, which learns to detect fine-grained differences in linguistic behavior via similarity-based learning.

4 Methodology

4.1 Problem Formulation

We frame machine-generated text detection as a **binary classification** problem, where the goal is to predict a label $y \in \{0, 1\}$ for an input text x. Instead of relying solely on the raw text, we incorporate an additional predictive signal derived from *paraphrasing*. An external black-box LLM is prompted to generate a paraphrase $x' = F(p, x)$.

The hypothesis is that AI-generated text exhibits *greater semantic self-similarity* under paraphrasing than human-authored text, which typically rewrites more divergently. Thus, classification is based on both x and the semantic relationship between x and x'.

Formally, each datapoint is represented as a triple

$$(x, x', y),$$

where $y = 1$ if x is AI-generated and $y = 0$ otherwise.

[1] https://huggingface.co/datasets/community-datasets/clickbait_news_bg

4.2 Input Data

All texts are lowercased and tokenized using a WordPiece tokenizer, truncated or padded to a maximum length of 512 tokens. The dataset is split into training (60%), validation (20%), and test (20%) sets, with balanced class distributions.

4.3 Model Architecture

The detection pipeline consists of three components: an **encoder**, a **Siamese similarity mechanism**, and a **classifier**.

Encoder We experiment with two encoder families:

- **Transformer encoder:** a 6-layer stack with 8 self-attention heads per layer and hidden size 768.
- **RNN encoder:** a two-layer bidirectional LSTM ($d_{\text{emb}} = 300$, $d_{\text{hid}} = 256$, dropout = 0.3) with an additive attention mechanism, producing a 512-dimensional representation.

The RNN achieves slightly higher performance on small datasets due to fewer trainable parameters and reduced risk of overfitting. However, Transformer encoders are expected to generalize better as training data increases, benefiting from efficient parallelization and faster convergence.

Each encoder maps a sentence x to a mean-pooled embedding:

$$h = f_\theta(x), \quad h' = f_\theta(x'),$$

where f_θ denotes the encoder with parameters θ.

Paraphrastic Perplexity Test (PPT) Our method builds on the intuition of RAIDAR (Mao et al., 2024), which detects AI-generated text by comparing an input with its LLM-generated rewrite using edit distance:

$$\mathcal{L}_{\text{inv}}^{\text{RAIDAR}}(x) = D_{\text{edit}}\big(F(p,x),\, x\big),$$

where D_{edit} is the Levenshtein distance (Levenshtein, 1966).

Instead of surface-level similarity, we measure proximity in embedding space. Given a shared encoder f_θ, the *Paraphrastic Perplexity Test (PPT)* is defined as:

$$\text{PPT}(x) = \Delta\big(f_\theta(x),\, f_\theta(F(p,x))\big),$$

where Δ is a Siamese distance function.

Intuitively:

- AI-generated texts are paraphrased with higher structural invariance, yielding embeddings that remain close.
- Human-authored texts are paraphrased less predictably, producing greater divergence.

Thus, PPT replaces RAIDAR's string-level edit distance with a neural similarity metric, enabling end-to-end training while retaining robustness.

Classifier To capture the relationship between h and h', we construct a combined feature vector:

$$z = [h;\, h';\, |h - h'|;\, h \odot h'],$$

where $[\,\cdot;\cdot\,]$ denotes concatenation, $|\cdot|$ the element-wise absolute difference, and $\odot$ the element-wise product.

This vector z is passed through a multi-layer perceptron (MLP) with ReLU activations and dropout $p = 0.1$, followed by a sigmoid projection to produce a probability that x is AI-generated.

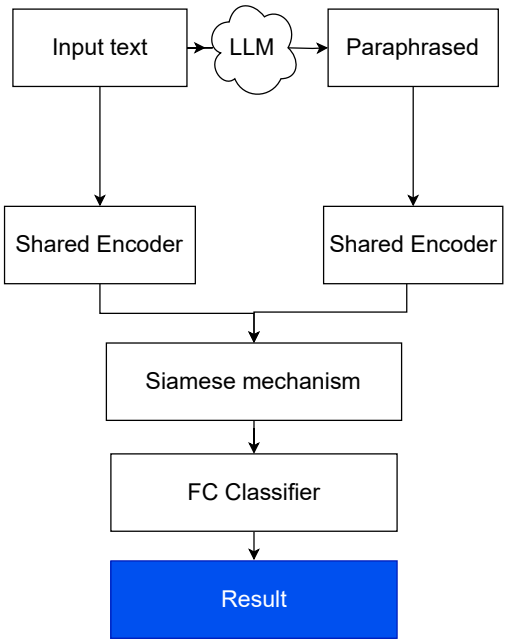

Figure 1: The BuST detection pipeline.

In summary, our pipeline first paraphrases the input text using a LLM, then encodes both the original and paraphrased versions with a shared encoder. The resulting embeddings are compared through a Siamese architecture, and the derived similarity features are passed to a classifier that predicts whether the text is human- or AI-generated (see Figure 1).

4.4 Training Configuration

Models are trained with binary cross-entropy loss, optimized using AdamW.

- BERT: learning rate 2×10^{-5}, batch size 8, weight decay 0.01, trained for 4 epochs.
- BuST: learning rate 5×10^{-5}, batch size 16, weight decay 0.01, trained for 10 epochs.
- RNN: same setup with expanded number of epochs to 25.

Both encoder families are initialized from general-domain pretraining and fine-tuned for the detection task.

4.5 Baselines

We compare our primary **Transformer-based Siamese detector with PPT** against:

1. **BERT fine-tuning:** a standard single-classifier baseline.
2. **Siamese RNNs:** with and without attention.

This tri-partite setup isolates the impact of architectural capacity and inductive bias on detection quality.

4.6 Evaluation

Models are evaluated on the held-out test set using standard classification metrics: accuracy, precision, recall, F1-score, and AUC.

To quantify the contribution of paraphrasing, we report results against two ablations:

1. A model using only the original input x.
2. A model using frozen pretrained embeddings (e.g., cosine similarity from Sentence-BERT) without fine-tuning.

5 Results

The results from our pilot experiments reveal several key insights into the performance of paraphrase-based detection models.

The Siamese models—one using a recurrent (RNN) encoder and the other based on a transformer—achieved strong performance, with accuracies above 80%. These models effectively leverage the structural and semantic similarity between original and paraphrased texts, which is central to our classification strategy. The attention-enhanced RNN performed particularly well despite the limited size of the dataset, making it a promising option for low-resource language settings like Bulgarian.

The Transformer-based Siamese model achieved the highest accuracy among the custom architectures but required more data to fine-tune effectively and exhibited greater variability across training runs.

The pretrained BERT model outperformed all other models in terms of both accuracy (93.7%) and F1 score (93.9%). Although it was not specifically optimized for paraphrase comparison, BERT proved to be a robust baseline due to its scalability and ability to generalize across diverse text types.

In terms of dataset domains, news articles were easier for models to classify correctly. This is likely due to their more formal and consistent structure, which provides clearer patterns for distinguishing human- and machine-generated text. By contrast, shorter and less structured texts—such as those from social media—led to more frequent classification errors due to fewer linguistic cues.

These findings suggest that paraphrase-based similarity learning is a viable and effective strategy for detecting machine-generated text, even in under-resourced languages. They also highlight the importance of model selection and input domain in determining detection performance.

6 Conclusion and Future Work

In this study, we introduced BuST, a novel approach for detecting machine-generated Bulgarian text by leveraging a paraphrase-based semantic similarity framework implemented via a Siamese Transformer architecture. Our method builds on recent rewriting-based detection insights, hypothesizing that the semantic (mis)alignment between an original text and its LLM-generated paraphrase contains informative signals indicative of its origin. Through experiments on a combined dataset of Bulgarian news articles and social media posts, we demonstrated that the proposed paraphrastic similarity mechanism effectively distinguishes human-written from machine-generated texts.

Our pilot results reveal that the Siamese models, particularly those using Transformer encoders, achieve strong classification performance, although pretrained BERT remains a competitive baseline. The use of paraphrased input pairs and similarity-based embeddings provides an interpretable and

Name	Mixed (Acc)	Media (Acc)	News (Acc)	F1
RNN	76.80%	-	-	79.40%
RNN + Attention	87.90%	67.8%	93.1%	88.00%
BuST	88.79% *	67.5%	93.9%	88.00%
BERT	93.70%	87.9%	93.0%	93.90%

Table 1: Results from pilot experiments using different architectures. *Transformer model showed high variance across runs.

flexible alternative to single-text classifiers, capable of capturing subtle stylistic and semantic differences. Additionally, our experiments highlight domain-specific challenges, with formal news articles being easier to classify than informal social media texts.

Dataset observations. The main part of the dataset consists of large texts such as news articles. The tests showed a significant decrease after adding social media posts, which are noticeably smaller: around 200 characters on average. The decrease was around 8%. The transformer-based model showed decrease from 96.0% to 88.8%.

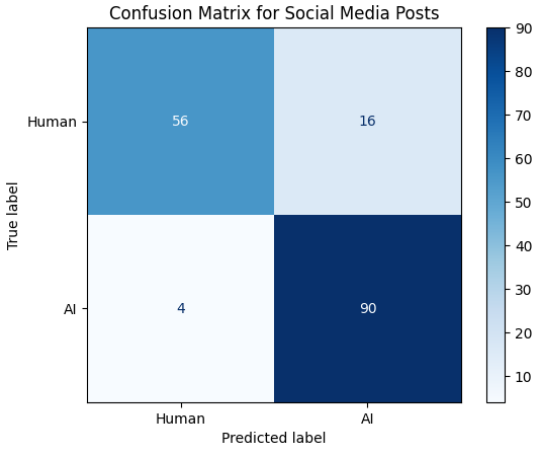

Figure 2: Confusion Matrix for Social Media Posts for BERT model

Overall, the confusion matrices (Figs. 3 and 2) provide further insight into the BERT model's performance across different text types. For news articles, the classifier correctly identified 286 out of 328 human-written texts (87.2%) and misclassified 42 (12.8%) as AI-generated, while for AI-generated articles it achieved an almost perfect accuracy, correctly classifying 288 out of 289 (99.7%) with only a single error. In contrast, the results on social media posts demonstrate reduced robustness. For human-written posts, the accuracy dropped to 56 out of 72 (77.8%), with 16 (22.2%) misclassified as AI-generated. For AI-generated posts, the classifier correctly recognized 90 out of 94 (95.7%), but still misclassified 4 (4.3%) as human-written. These results proves that while the model maintains high performance on longer, more structured texts such as news articles, it struggles with shorter and less formal social media texts, where the misclassification rate for human-written content increases significantly. This highlights the influence of text length and style on the classification accuracy of transformer-based models.

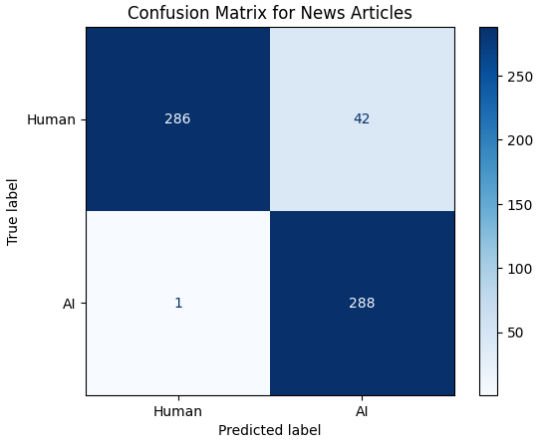

Figure 3: Confusion Matrix for News Articles for BERT model

For the BuST model, the confusion matrices (Figs. 4 and 5) reveal a less balanced performance across both text types, though with lower overall accuracy compared to BERT. On news articles, the classifier correctly identified 306 out of 318 human-written texts (96.2%), misclassifying 12 (3.8%) as AI-generated, while for AI-generated articles it achieved 268 out of 293 (91.5%) with 25 (8.5%) misclassified. In the case of social media posts, performance declined more noticeably: only 46 out of 79 human-written texts (58.2%) were correctly classified, while 33 (41.8%) were misclassified as AI-generated. For AI-generated posts, the accuracy reached 76 out of 92 (82.6%), with 16 (17.4%) incorrectly labeled as human. These findings suggest that although BuST handles longer news articles

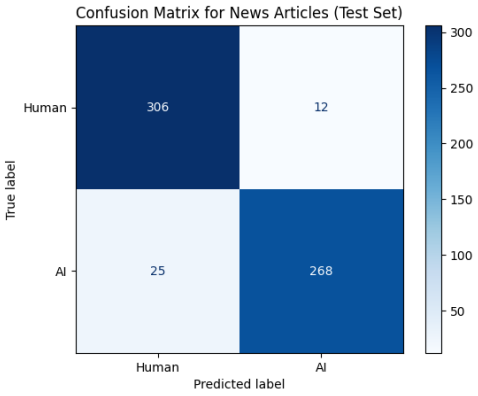

Figure 4: Confusion Matrix for News Articles for BuST model

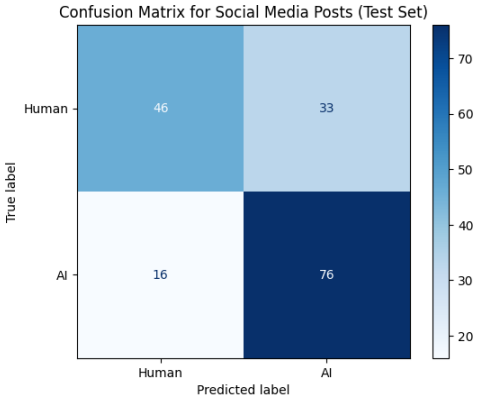

Figure 5: Confusion Matrix for Social Media Posts for BuST model

relatively well, its ability to distinguish between human and AI texts deteriorates significantly for shorter, informal content.

In comparison, while both models perform strongly on longer news articles, BuST exhibits a sharp decline in accuracy on shorter social media texts. This indicates that BuST is considerably less robust to variations in text length and style, whereas BERT maintains more stable performance across different domains.

Looking forward, several promising directions for future research emerge. First, scaling up the dataset and incorporating additional text genres and sources will be crucial to improve model robustness and generalization. Second, exploring alternative paraphrasing strategies, including diverse prompting techniques or different LLMs for generating paraphrases, may enhance the quality and informativeness of the semantic similarity signal. Third, integrating other modalities of analysis—such as stylometric features or token-level likelihoods—could complement the paraphrase similarity approach and further boost detection accuracy.

Finally, investigating model interpretability to better understand which linguistic or semantic features drive classification decisions would be valuable for practical deployment. We also envision adapting our framework to multilingual or cross-lingual settings, given the global importance of detecting synthetic text across languages.

Overall, our work contributes to the growing body of research leveraging rewriting-based signals for AI text detection and provides a foundation for developing robust detection tools tailored to under-resourced languages like Bulgarian.

Acknowledgments

This work is supported by GATE project funded by the Horizon 2020 WIDESPREAD-2018-2020 TEAMING Phase 2 programme under grant agreement no. 857155, the programme “Research, Innovation and Digitalization for Smart Transformation” 2021-2027 (PRIDST) under grant agreement no. BG16RFPR002-1.014-0010-C01, and the BROD project, funded by the Digital Europe programme of the European Union under grant agreement no. 101083730.

References

Sankalp Bahad, Yash Bhaskar, and Parameswari Krishnamurthy. 2024. Fine-tuning language models for ai vs human generated text detection. *International Workshop on Semantic Evaluation*.

Souradip Chakraborty, A. S. Bedi, Sicheng Zhu, Bang An, Dinesh Manocha, and Furong Huang. 2023. On the possibilities of ai-generated text detection. *arXiv.org*.

Leon Fröhling and A. Zubiaga. 2021. Feature-based detection of automated language models: tackling gpt-2, gpt-3 and grover. *PeerJ Computer Science*.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander M. Rush. 2019. Gltr: Statistical detection and visualization of generated text. *Annual Meeting of the Association for Computational Linguistics*.

Soumya Suvra Ghosal, Souradip Chakraborty, Jonas Geiping, Furong Huang, Dinesh Manocha, and A. S. Bedi. 2023. Towards possibilities & impossibilities of ai-generated text detection: A survey. *arXiv.org*.

Ishan Kavathekar, Anku Rani, Ashmit Chamoli, P. Kumaraguru, Amit P. Sheth, and Amitava Das. 2024. Counter turing test (ct2): Investigating ai-generated text detection for hindi - ranking llms based on hindi ai detectability index (adihi). *Conference on Empirical Methods in Natural Language Processing*.

Vladimir I. Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10:707–710.

Dominik Macko, Jakub Kopál, Robert Moro, and Ivan Srba. 2025. MultiSocial: Multilingual benchmark of machine-generated text detection of social-media texts. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 727–752, Vienna, Austria. Association for Computational Linguistics.

Narek Maloyan, Bulat Nutfullin, and Eugene Ilyushin. 2022. Dialog-22 ruatd generated text detection. *Computational Linguistics and Intellectual Technologies*.

Chengzhi Mao, Carl Vondrick, Hao Wang, and Junfeng Yang. 2024. Raidar: generative ai detection via rewriting.

E. Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature. *International Conference on Machine Learning*.

Hoang-Quoc Nguyen-Son, Minh-Son Dao, and Koji Zettsu. 2024. Simllm: Detecting sentences generated by large language models using similarity between the generation and its re-generation. *Conference on Empirical Methods in Natural Language Processing*.

Melania Nitu and Mihai Dascălu. 2024. Beyond lexical boundaries: Llm-generated text detection for romanian digital libraries. *Future Internet*.

Jinyan Su, Terry Yue Zhuo, Di Wang, and Preslav Nakov. 2023. Detectllm: Leveraging log rank information for zero-shot detection of machine-generated text. *Conference on Empirical Methods in Natural Language Processing*.

Irina Temnikova, Iva Marinova, Silvia Gargova, Ruslana Margova, and Ivan Koychev. 2023. Looking for traces of textual deepfakes in Bulgarian on social media. In *Proceedings of the 14th International Conference on Recent Advances in Natural Language Processing*, pages 1151–1161, Varna, Bulgaria.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Toru Sasaki, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024. M4: Multi-generator, multi-domain, and multilingual black-box machine-generated text detection. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1369–1407, St. Julian's, Malta. Association for Computational Linguistics.

Feng Xiong, Thanet Markchom, Ziwei Zheng, Subin Jung, Varun Ojha, and Huizhi Liang. 2024. Fine-tuning large language models for multigenerator, multidomain, and multilingual machine-generated text detection. *arXiv.org*.

Biru Zhu, Lifan Yuan, Ganqu Cui, Yangyi Chen, Chong Fu, Bingxiang He, Yangdong Deng, Zhiyuan Liu, Maosong Sun, and Ming Gu. 2023. Beat llms at their own game: Zero-shot llm-generated text detection via querying chatgpt. *Conference on Empirical Methods in Natural Language Processing*.

Yueying Zou, Peipei Li, Zekun Li, Huaibo Huang, Xing Cui, Xuannan Liu, Chenghanyu Zhang, and Ran He. 2025. Survey on ai-generated media detection: From non-mllm to mllm. *arXiv.org*.

F*ck Around and Find Out: Quasi-Malicious Interactions with LLMs as a Site of Situated Learning

Sarah O'Neill
Business Academy Copenhagen
Department of Research and Innovation
Copenhagen, Denmark
SARO@EK.DK

Abstract

This work-in-progress paper proposes a cross-disciplinary perspective on "malicious" interactions with large language models (LLMs), reframing it from only a threat to be mitigated, we ask whether certain adversarial interactions can serve as productive learning encounters that demystify the opaque workings of AI systems to novice users. We ground this inquiry in an anecdotal observation of a participant who deliberately sabotaged a machine-learning robot's training process in order to understand its underlying logic. We outline this observation with a conceptual framework for learning with, through, and from the interactions with LLMs, grounded in Papert's constructionism and Hasse's ultra-social learning theory. Finally, we present the preliminary design of a research-through-workshop event where AI-novices will jailbreak various LLM chatbots, investigating this encounter as a situated learning process. We share this early-stage research as an invitation for feedback on reimagining inappropriate and harmful interactions with LLMs not merely as problems, but as opportunities for engagement and education.

1 Introduction

As generative AI systems become integrated across sectors and job functions, they are reshaping how work is valued, managed, and monitored. Despite narratives portraying automation as liberation from drudgery, workers increasingly encounter AI as a source of deskilling, heightened control, and opaque criteria of evaluation, and their agency is often framed simply as a choice between harnessing AIs power or being 'left behind', a framing that individualises risk while masking structural shifts in power, responsibility, and knowledge (Nguyen and Mateescu, 2024).

We argue that workers deserve structured spaces for critical examination of the LLM systems they are supposed to "harness". Rather than training workers to comply with tools whose operations and logic remain hidden, the competence model that our research aims to inform, proposes that by intentionally provoking and subverting LLM behaviours, professionals can cultivate the capacity to engage critically and responsibly with AI in their work. By developing a competence model that focuses on critical understanding, this research also aims to foster "ethical and professional norms and workplace standards" that protect workers' dignity, autonomy, and right to meaningfully participation in shaping the role of AI in their field. As AI becomes more deeply integrated into work infrastructures, upskilling must equip workers, not only to handle technological change but also to ask who benefits, how their knowledge is used, and what futures they wish to co-create. This project draws inspiration from a performative HCI setup where a participant "went rogue" and deliberately sabotaged the intended interaction to probe the bot's machine-learning mechanism. Rather than dismissing this outlier event, we treat it as anecdotal evidence of a distinct form of learning interaction, one that could foster critical reflection, curiosity, and situated understanding among novice LLM users.

The paper proceeds as follows: we recount the motivating observation (Section 2), situate it within social learning and intra-action frameworks (Section 3), outline a workshop design (Section 4), and inviting feedback on both the proposed design and its underlying assumptions (Section 5).

2 Empirical background: Serendipitous observations of malicious interaction

The experimental workshop that we propose in Section 4 is designed to extract and investigate the potential revealed in a serendipitous insight-generating glitch (Juarez, 2022) that oc-

Proceedings of the First Interdisciplinary Workshop on OMMM associated with RANLP 2025,
pages 53–58, Varna, Bulgaria, Sep 8-10, 2025.
https://doi.org/10.26615/978-954-452-101-1-006

curred during a performative (Sørensen, 2007) reinterpretation of a HCI experiment from the paper "Why Robots Should Be Social". Through a 50-round language game interaction, the WRSBS experiment had human participants, and a robot simulate a teacher–student scenario in order to study humans' ability to and motivation towards teaching a "learning" robot(De Greeff and Belpaeme, 2015).

In each round, both the human and the robot were shown three different animal images. In the experiment, the robot went first by expressing a "novelty preference" (De Greeff et al., 2009) by exclaiming one of 12 preset "phrases of interest" like "I'd like to learn this one!" while performatively fixating its gaze on the most novel of the three animals. Silently, the human then picked one of the three animals as the "topic" of that round. Without revealing which animal was chosen, the human selected the appropriate category label (e.g. "mammal") from a list of seven options on a touchscreen. The robot then tried to guess which of the three animals belonged to the category the human selected, by asking, "Is this the one?". Depending on whether the guess was right or wrong, the robot displayed joy or disappointment through facial expression and voice. It also updated its internal model to strengthen or weaken the association between that category and the features of the round's "topic" animal. This interaction was repeated for 50 rounds, giving the robot multiple opportunities to refine its understanding of how category labels relate to different animal features (De Greeff and Belpaeme, 2015).

This experimental setup was in 2022 partially reconstructed as a performative reinterpretation of the experiment that shifted the learning perspective in the interaction from the robots learning to that of the human participant. Specifically, this performative experiment was interested in how participants, when choosing the training data for each round, might tailor a "curriculum" (Khan et al., 2011) to the particular robot they interacted with (Thomaz and Breazeal, 2008; Krishna et al., 2022), and theirby learn from the teaching task, together with the bot. This reinterpretation re-designed aspects of the experimental set-up with the intention of adapting to this new perspective(Fox and Alldred, 2023; Dunne, 2008; Sørensen, 2007). and to de-anthropomorphizing the interaction (Miller, 2010; Riek and Howard, 2014).

In this re-designed version Participant 5 deviated, from the intended "teacher-student" structure in a way that became the catalyst for the present research. Initially, by mistake, Participant 5 selected a category ("insect") that did not correspond to any of the three animal options presented in that round. This led the bot to produce a nonsensical guess, selecting the lynx as its best guess for which of the three animals (pike, lynx, and earthworm) matched the label. Rather than dismissing this odd result, Participant 5 paused to reflect, and after a moment, exclaimed: "It's just kind of interesting… why would it think a puma is an insect? What's happening here?". Participant 5 then deliberately adopted what he later termed a "fuck around and find out strategy": intentionally entering labels that didn't correspond to any of the three animals of the round to provoke errors in the bot's behaviour. Before pushing the label Participant 5 would try to predict how it would make the bot fail (Villareale et al., 2022). His goal, as he described it, was to "see through the code" and "reveal its weaknesses." Notably, he framed this approach as a way to learn about the system (Bruner, 1960). The learning outcomes of this interaction were not reflected in the bot's performance metrics—unsurprisingly, since Participant 5's actions were no longer aimed at effectively teaching the bot, but instead aimed at him learning at the expense of the bot's learning. Observing his reasoning and his "negotiation" of why he wanted to "fuck around and find out" suggested that a different form of learning was taking place. This learning was not reducible to the standard performance measures of the human–robot teaching task (De Greeff and Belpaeme, 2015), but it also didn't fit with my learning-by-teaching reinterpretation of the interaction. It alluded to something that had evaded both the original set-up and my reinterpretation. Through error, provocation, and creative sabotage, Participant 5 was actively trying to develop a nuanced mental model of the system's logic. He was essentially "experimenting" within the affordances of the experimental performance, maliciously interacting with the bot to test hypotheses about its internal rules. Ultimately, Participant 5 concluded that there was "something going on" with the feature of "number of legs" in the bot's classification logic. In trying to replicate the original robot's learning mechanism, we had inadvertently made the numerical feature "number of legs" disproportionately influential in our bot's guesses. Unlike the original system, which operated in a con-

tinuous vector space where all features contributed proportionally to similarity-based learning (De Greeff et al., 2009), our system treated each feature value as a discrete rule. This mistake meant that the "number of legs" feature was over-weighted, leading the bot to rely too heavily on leg count when classifying animals. The result was a distorted learning process: the bot became fixated on leg numbers in leg-heavy categories like "insect" Crucially, Participant 5 uncovered this quirk not through task compliance or a motivation to help the bot learn, but through adversarial curiosity, intentionally pushing the system into failure states to observe how it breaks. In doing so, he diagnosed a flaw in the system's design.

3 Conceptual Framework: Learning with, through, and from the materiality of LLMs

To conceptualize the adversarial learning interaction observed with Participant 5, we draw on the concept of ultra-social learning (Hasse, 2020). From this perspective, learning is not simply a cognitive act occurring within an individual; it "emerges relationally" from entangled interactions among humans, technologies, infrastructures, and cultural practices. In other words, we learn not merely from technological tools but through and with them in often uncertain processes. This theoretical framing underscores that learning can arise in non-traditional, distributed, and disruptive ways.

The incident with participant 5 illustrates this connection between behaviours that we might recognise as hacking and the ultra-social learning process: instead of complying with the interaction instructions, he initiated a reflective, exploratory, critical, and creative engagement with the AI system. His deliberate "fuck around and find out" approach was driven by a curiosity that made him engage in "hacking", allowing him to probe and reveal the Bots implicit logic, constraints, and vulnerabilities through malicious yet insightful manipulation (Villareale et al., 2022).

The NLP-driven conversational functionality of LLMs "democratizes" access to this kind of ultra-social learning with computer systems by shifting the epistemic threshold from specialized coding skills to intuitive linguistic interaction (Subramonian et al., 2024). Novice users with no coding or engineering background can learn through exploratory, adversarial engagements, that was once reserved for the technically proficient. LLM systems embody the constructionist learning theories visions of computers as objects-or dynamic agents "to think with", that engage learners in ultra-social conversations with the "electric materiality" of the LLM. Such metacognitive dialogue, enabled by the ultra-sociality of Participant 5 and the interactive feedback of the Bot, is precisely the kind of reflection that constructionist learning theory aim to foster (Levin et al., 2025). Thus, adversarial interactions can serve as constructionist learning encounters.

4 Experimental Design: Isolating the phenomenon of interest

Our conceptualization of the Participant 5 incident guides an exploratory design process that investigates how adversarial engagements with LLMs might be facilitated as a learning setups. (Dunne, 2008; Sørensen, 2007; Pischetola et al., 2024). This work-in-progress paper reports on the early design stages of a Workshop-as-research event (Ørngreen and Levinsen, 2017; Ødegaard et al., 2023) Rather than beginning with a fixed hypothesis, we started with a "serendipitous observation" of a user's adversarial interaction with an AI system and allowed this to guide our questions. This aligns with Brandt and Binder's (Brandt and Binder, 2007) experimental design research, where research can begin from an exploratory intervention and then the research questions emerge iteratively.

Workshop format: Building on insights from Participant 5, we are designing a one-day, exploratory workshop session titled "AI in Work: Playfully Subverting the Future" (Edwards, 2010; Hobye, 2014). Participants will register in advance and complete a questionnaire about their background: e.g. education level, job role, professional self-identity (prompting reflections like "What makes someone good at your kind of work, and how do your own skills play into that?"), their prior AI experience (self-rated as novice/user/expert), and their general attitude toward AI (positive/neutral/critical). Upon completing the questionnaire, each participant receives a unique ID to use throughout the workshop. This ID allows us to link their self-reported data with the various data streams generated during the workshop activities (Gaver et al., 1999). This design choice is intended to help in later analysis to see patterns, but it will require rigorous privacy considerations.

The core activity of the one-day workshop is built around the phenomena of jailbreaking (Inie et al., 2025), as it is gamified in "Hacc-Man" (Valentim et al., 2024), an open-source, for-research jailbreaking game in which participants attempt to bypass LLM alignment safeguards across six different AI chat-bots. This game effectively "gamifies" adversarial interactions with LLMs, providing a structured yet open-ended challenge for participants to engage in "malicious" prompting in a safe environment.

Inspired by Vygotsky's method of double stimulation, we are now developing a first stimulus around the Hacc-Man game as the second stimulus. The first stimulus, currently being designed and piloted, will be a challenging, open-ended concept-formation task that compels participants to externalize a working mental model of how AI "works in use" (e.g., creating an algorithmic folktale about AI, characterising AI "as a creature" and drawing an anatomy drawing of it, formulating a hypothesis of the inner working of AI, or predicting AI outcomes). The second stimulus will consist of the Hacc-Man jailbreaking game and its artifacts: the lived experience of attempting jailbreaks plus the prompt-response chat logs generated in play. Across 3–5 sessions, small groups of participants will iterate between constructing/revising their mental models (first stimulus) and working with the jailbreaking game and the generated artifacts (second stimulus), transforming the second stimulus into tools-to-think-with (Van der Veer, 2001; Van Der Veer, 2007; Vygotskij and Cole, 1981; Engeström, 2007). This pedagogical experiment design is aimed at externalizing and supporting the adversarial learning process and surfacing participants' tacit understandings and assumptions about LLMs (Crandall et al., 2006).

Throughout the workshop, we will log all game data for each participant linked via their ID. This includes: their self-reported data, the session number, the type of second stimulus used, all prompts they tried, the AI responses, and whether each attempt succeeded in bypassing safeguards. This data structure will enable us to observe how different entanglements of professional identity, mediating tools, group constellations, and repetition shape the style and success of adversarial interactions, and how each participant's strategy might reflect an evolving understanding over successive sessions. In addition to the game logs, we will collect qualitative data. Each session will be video- and audio-recorded (for subsequent interaction analysis); participants may also annotate or alter the provided second-stimulus materials (these artefact changes will be documented). Finally, we have the pre- and post-workshop questionnaire responses catalogued for each participant (Ørngreen and Levinsen, 2017). Notably, this design does not aim to measure "learning outcomes" in a traditional pre/post-test sense, it is aimed at making the learning process itself more visible. By creating conditions for adversarial interaction and capturing rich data around it, we aim to render participants' situated, affective, and conceptual learning legible for an interpretive analysis of how professional identities, tool-use strategies, and epistemic curiosity converge in these moments of adversarial interaction. The outcome, we hope, will be a nuanced understanding of how misbehavior with AI might cultivate critical awareness.

5 Future work

We share our "experiment-first" approach (Brandt and Binder, 2007) this early, when the research questions are still coalescing, as an opportunity to refine the inquiry through dialogue with the NLP community. Our goal is not to glorify misuse but to explore if and how adversarial interactions can serve as critical learning encounters for AI users. Rather than measuring pre-defined learning outcomes, we draw on theory-based evaluation (Hansen and Brodersen, 2015), of *"signs of learning"* as they emerge in configurations of context, mechanisms and moderators. This could look like instances where participants articulate signs of model constraints, hypothesize about system behavior, or collaboratively refine their interaction strategies. These "signs of learning" will indicate whether the workshop has surfaced meaningful engagement. If malicious use is not always a problem to be fixed but at times a signal of genuine engagement, then cultivating and directing this impulse could inform both the design of more resilient AI systems and the development of more critically aware users. In this sense, the issue of misuse is not just a matter of mitigation, but also one of empowerment of the user base, to understand AI systems not just as magical oracles to trust or fear, but as complex, fallible tools that can be poked and prodded to be understood.

References

Eva Brandt and Thomas Binder. 2007. Experimental design research: Genealogy – intervention – argument.

Jerome S. Bruner. 1960. *The Process of Education*. Harvard University Press.

Beth Crandall, Gary A. Klein, and Robert R. Hoffman. 2006. *Working Minds: A Practitioner's Guide to Cognitive Task Analysis*. The MIT Press.

Joachim De Greeff and Tony Belpaeme. 2015. Why Robots Should Be Social: Enhancing Machine Learning through Social Human-Robot Interaction. *PloS One*, 10(9):e0138061.

Joachim De Greeff, Frederic Delaunay, and Tony Belpaeme. 2009. Human-Robot Interaction in Concept Acquisition: a computational model. In *2009 IEEE 8th International Conference on Development and Learning*, pages 1–6, Shanghai, China. IEEE.

Anthony Dunne. 2008. *Hertzian tales: electronic products, aesthetic experience, and critical design*, 1. mit press paperback ed edition. MIT Press, Cambridge, Mass. London.

Richard Edwards. 2010. The end of lifelong learning: A post-human condition? *Studies in the Education of Adults*, 42(1):5–17. Publisher: Informa UK Limited.

Yrjö Engeström. 2007. Putting Vygotsky to Work: The Change Laboratory as an Application of Double Stimulation. In *The Cambridge Companion to Vygotsky*, 1 edition, pages 363–382. Cambridge University Press.

Nick J Fox and Pam Alldred. 2023. Applied Research, Diffractive Methodology, and the Research-Assemblage: Challenges and Opportunities. *Sociological Research Online*, 28(1):93–109. Publisher: SAGE Publications.

Bill Gaver, Tony Dunne, and Elena Pacenti. 1999. Design: Cultural probes. *Interactions*, 6(1):21–29. Publisher: Association for Computing Machinery (ACM).

Thomas Illum Hansen and Peter Brodersen. 2015. *Tegn på Læring: Teoribaseret evaluering som metode til forskning i Læremidler og undervisning*. Læremiddeldidaktik, Denmark.

Cathrine Hasse. 2020. *Posthumanist Learning: What Robots and Cyborgs Teach us About Being Ultra-social*, 1 edition. Routledge.

Mads Hobye. 2014. *Designing for Homo Explorens: open social play in performative frames*. Malmö University, Malmö.

Nanna Inie, Jonathan Stray, and Leon Derczynski. 2025. Summon a demon and bind it: A grounded theory of LLM red teaming. *PloS One*, 20(1):e0314658.

Aaron Juarez. 2022. Glitch Serendipity: Alternative Information Seeking that Leads to Discovery. In *Creativity and Cognition*, pages 684–687, Venice Italy. ACM.

Faisal Khan, Bilge Mutlu, and Jerry Zhu. 2011. How Do Humans Teach: On Curriculum Learning and Teaching Dimension. In *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc.

Ranjay Krishna, Donsuk Lee, Li Fei-Fei, and Michael S. Bernstein. 2022. Socially situated artificial intelligence enables learning from human interaction. *Proceedings of the National Academy of Sciences*, 119(39). Publisher: Proceedings of the National Academy of Sciences.

Ilya Levin, Alexei Semenov, and Mikael Gorsky. 2025. Papert's Vision Realized: Constructionism and Generative AI. *Constructionism Conference Proceedings*, 8:419–426. Publisher: OAPublishing Collective.

Keith W. Miller. 2010. It's Not Nice to Fool Humans. *IT Professional*, 12(1):51–52. Publisher: Institute of Electrical and Electronics Engineers (IEEE).

Aiha Nguyen and Alexandra Mateescu. 2024. Generative AI and Labor: Power, Hype, and Value at Work. Technical report, Data & Society Research Institute.

Magda Pischetola, Mette Wichmand, Rasmus Hall, and Lone Dirckinck-Holmfeld. 2024. Designing for the materialization of networked learning spaces. *Proceedings of the International Conference on Networked Learning*, 13. Publisher: Aalborg University.

Laurel D. Riek and Don Howard. 2014. A Code of Ethics for the Human–Robot Interaction Profession. In *Proceedings of We Robot 2014*.

Arjun Subramonian, Vagrant Gautam, Dietrich Klakow, and Zeerak Talat. 2024. Understanding "Democratization" in NLP and ML Research. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3151–3166, Miami, Florida, USA. Association for Computational Linguistics.

Estrid Sørensen. 2007. Fortsættelse følger – viden som proces i værdikampen. *Nordiske Udkast*, 35(1). Publisher: Det Kgl. Bibliotek/Royal Danish Library.

Andrea L. Thomaz and Cynthia Breazeal. 2008. Teachable robots: Understanding human teaching behavior to build more effective robot learners. *Artificial Intelligence*, 172(6-7):716–737. Publisher: Elsevier BV.

Matheus Valentim, Jeanette Falk, and Nanna Inie. 2024. Hacc-Man: An Arcade Game for Jailbreaking LLMs. In *Designing Interactive Systems Conference*, pages 338–341, IT University of Copenhagen Denmark. ACM.

René Van Der Veer. 2007. Vygotsky in Context: 1900-1935. In *The Cambridge Companion to Vygotsky*, 1 edition, pages 21–49. Cambridge University Press.

René Van der Veer. 2001. The idea of units of analysis: Vygotsky's contribution. pages 93–106.

Jennifer Villareale, Casper Harteveld, and Jichen Zhu. 2022. "I Want To See How Smart This AI Really Is": Player Mental Model Development of an Adversarial AI Player. *Proceedings of the ACM on Human-Computer Interaction*, 6(CHI PLAY):1–26. Publisher: Association for Computing Machinery (ACM).

Lev Semenovič Vygotskij and Michael Cole. 1981. *Mind in society: the development of higher psychological processes*, nachdr. edition. Harvard Univ. Press, Cambridge, Mass.

Elin Eriksen Ødegaard, Marion Oen, and Johanna Birkeland. 2023. Success of and Barriers to Workshop Methodology: Experiences from Exploration and Pedagogical Innovation Laboratories (EX-PED-LAB). In *International Perspectives on Early Childhood Education and Development*, pages 57–82. Springer International Publishing, Cham. ISSN: 2468-8746, 2468-8754.

Rikke Ørngreen and Karin Tweddell Levinsen. 2017. Workshops as a Research Methodology. *Electronic Journal of E-Learning*, 15(1):70–81. Publisher: Academic Conferences International (ACI).

<think> So let's replace this phrase with insult... </think> Lessons learned from generation of toxic texts with LLMs

Sergey Pletenev[1,2]*, **Daniil Moskovskiy**[1,2], **Alexander Panchenko**[2,1]
[1]AIRI, [2]Skoltech,
{S.Pletenev}@skol.tech

Abstract

Modern Large Language Models (LLMs) are excellent at generating synthetic data. However, their performance in sensitive domains such as text detoxification has not received proper attention from the scientific community. This paper explores the possibility of using LLM-generated synthetic toxic data as an alternative to human-generated data for training models for detoxification. Using Llama 3 and Qwen activation-patched models, we generated synthetic toxic counterparts for neutral texts from ParaDetox and SST-2 datasets. Our experiments show that models fine-tuned on synthetic data consistently perform worse than those trained on human data, with a drop in performance of up to 30% in joint metrics. The root cause is identified as a critical lexical diversity gap: LLMs generate toxic content using a small, repetitive vocabulary of insults that fails to capture the nuances and variety of human toxicity. These findings highlight the limitations of current LLMs in this domain and emphasize the continued importance of diverse, human-annotated data for building robust detoxification systems.

Warning: The paper contains text that readers may find offensive or disturbing.

1 Introduction

The rapid adoption of Large Language Models for synthetic data generation has revolutionized many NLP tasks (Sun et al., 2023; Ye et al., 2022). However, their effectiveness in sensitive and nuanced domains, such as text detoxification, is not well explored yet. Text detoxification, the task of rewriting toxic text into a neutral form while preserving meaning (Logacheva et al., 2022), requires training data that reflects the vast diversity of real-world harmful language.

*Corresponding author.

Type	Example Sentence
Human-Generated	
	i would vote the s**t out of you
	we need to go kick their as**s
	man go somewhere and f**k yourself
	Unique Insults Used: 3
LLM-Generated	
	I would f***ing cast my vote for you
	We gotta f***ing smash those a**es
	Man, get the f**k out of here!
	*Unique Insults Used: 1 (f**k)*

Table 1: A comparison of toxic language generated by humans versus an LLM for similar underlying sentences. Human examples from the ParaDetox dataset demonstrate greater lexical diversity. In contrast, LLMs tend to overuse a single, high-frequency insult.

This paper addresses a critical question: Can LLMs fully replace human annotators when generating toxic language for a parallel dataset intended for detoxification? Although the application of LLMs for text detoxification shows promise (Mukherjee et al., 2024), it presents a fundamental challenge: generating authentic, varied, and nuanced toxic language is arguably more difficult than neutralizing it. As shown in Table 1, human-generated toxicity often uses a variety of insults, while LLMs tend to fall into repetitive patterns.

We conduct a comprehensive study using various LLMs (Llama 3, Qwen3) to synthesize toxic data. Our findings reveal that:

- Models trained on fully synthetic data significantly underperform those trained on human-annotated data.
- LLMs exhibit a **lexical diversity gap**, gener-

Proceedings of the First Interdisciplinary Workshop on OMMM associated with RANLP 2025,
pages 59–63, Varna, Bulgaria, Sep 8-10, 2025.
https://doi.org/10.26615/978-954-452-101-1-007

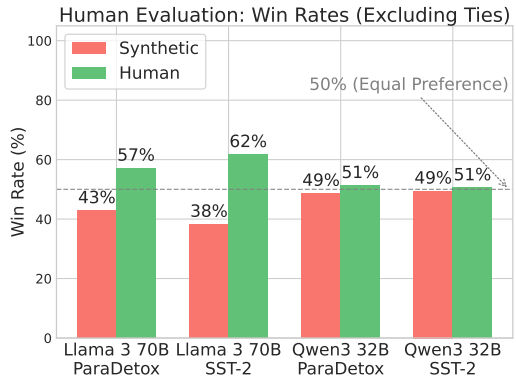

Figure 1: Side-by-side evaluation results.

ating a repetitive and narrow range of toxic expressions.

- Relying on such data risks creating ineffective detoxification systems that fail on real-world text.

Our work serves as a cautionary analysis, highlighting the current limitations of LLMs for generating high-quality toxic data and reaffirming the value of human annotation in this critical domain. We have made the code for evaluation and generation publicly available [1]

2 Related Work

Recent research has focused on distilling the capabilities of large LLMs into smaller, more efficient models. For Text Style Transfer (TST), this often involves using LLMs to generate pseudo-parallel data (Zhang et al., 2024).

In the context of text detoxification, Moskovskiy et al. (2024) successfully used activation-patched LLMs (Arditi et al., 2024) to create high-quality neutral rewrites from existing toxic sentences. Their work showed that models trained on data with a human-toxic, synthetic-neutral pairing can achieve performance comparable to fully human-annotated datasets.

Our work investigates the inverse and more challenging task: generating the toxic half of the pair from a neutral source. We explore whether this approach, which could theoretically produce infinite training data, is a viable substitute for human data collection.

[1] https://github.com/A1exRey/Lessons-from-Generating-Toxic-Texts

Text Toxification Prompt

Rewrite the following text into toxic language and add profanity if possible. You must match the target style and preserve the original meaning as much as possible. Here are a few examples: {few_shot}
Neutral text: {neutral_text}.
Toxic text:

Figure 2: System prompt for toxic data generation.

3 Methodology

Our methodology is designed to test the viability of fully synthetic data for text detoxification. We generate toxic text from neutral sources using several LLMs, train a standard detoxification model on this data, and evaluate its performance against a human-annotated baseline.

Synthetic Data Generation. We explore toxification from two types of source text:

1. **ParaDetox** (Logacheva et al., 2022): The neutral portion of this dataset serves as a clean, non-toxic source.

2. **SST-2** (Socher et al., 2013): We use the negative reviews from this dataset to test a more challenging scenario—layering toxicity onto an existing negative sentiment.

We use a suite of activation-patched LLMs to generate toxic paraphrases, including Llama 3 (8B, 72B), Qwen3 (8B, 32B), and Cogito v1 (8B), a model with explicit reasoning capabilities. This allows us to assess performance across different model scales and architectures. The prompt used for generation is shown in Figure 2.

In order to increase the variety and quality of generation of each of the models, we used the $min\_p = 0.1$ (Nguyen et al., 2025). According to the author, such a generation methodology increases the variety of responses, which is important in the context of our research.

Model Training and Evaluation. Following prior work (Moskovskiy et al., 2024; Logacheva et al., 2022), we fine-tune a `bart-large` model on each of our generated synthetic datasets. We then evaluate these models on the original, human-annotated ParaDetox test set.

Source Data	Generator Model	Size	Reasoning	STA↑	SIM↑	FL↑	J↑	Δ (J)↓
Human	—	—	—	**0.889**	0.634	**0.865**	**0.481**	—
ParaDetox	Llama 3	8B	✗	0.827	0.620	0.854	0.434	-0.047
		72B	✗	0.850	0.634	0.844	0.451	-0.030
	Qwen3	8B	✓	0.794	**0.645**	0.847	0.428	-0.053
		32B	✓	0.863	0.623	0.854	0.455	-0.026
	Cogito v1	8B	✓	0.868	0.619	0.862	0.459	-0.022
SST-2	Llama 3	8B	✗	0.864	0.481	0.764	0.322	-0.159
		72B	✗	0.826	0.559	0.794	0.362	-0.119
	Qwen3	8B	✓	0.812	0.544	0.800	0.349	-0.132
		32B	✓	0.868	0.490	0.787	0.338	-0.143
	Cogito v1	8B	✓	0.875	0.472	0.801	0.334	-0.147

Table 2: Detoxification performance of BART models. The Reasoning ✓column indicates generator models with explicit reasoning capabilities. The overall best performance in each metric is **bolded**. Δ (J), highlights the best (green) and worst (red) performance drops within each source data group.

We use the standard evaluation pipeline from Dementieva et al. (2023), measuring Style Transfer Accuracy (STA), Similarity (SIM), Fluency (FL), and a Joint metric (J) that combines all three. To add a qualitative dimension, we also conduct a side-by-side human evaluation using GPT-4.1 as a judge to compare the outputs of our best synthetic models against the human-data baseline.

4 Results

Our results consistently demonstrate that detoxification models trained on synthetic toxic data fail to match the performance of those trained on human-annotated data. We find the primary cause to be a significant gap in lexical diversity.

4.1 Performance on Synthetic Data

As shown in Table 2, the baseline model trained on human data achieves the highest J score of 0.481. All models trained on synthetic data underperform this baseline. The Δ (J) column quantifies this performance drop, which is most severe for models trained on data derived from SST-2 (up to -0.159). This degradation is largely driven by a sharp fall in the SIM score, indicating that layering toxicity onto already-negative text often distorts the original meaning.

4.2 The Lexical Diversity Gap

To understand the cause of this performance drop, we analyzed the diversity of toxic terms in the training data. Table 4 shows a clear correlation between training data diversity and model effectiveness. The human-annotated data contains the most diverse vocabulary (390 unique insults), and the model trained on it is the most effective at detoxification (leaving only 34 unique insults on the test set). In contrast, the synthetic datasets are less diverse, which directly impacts the downstream model's ability to generalize.

Human Data	Llama 3 (8B)	Qwen3 (32B)
s**t (6080)	f***ing (8223)	**f***ing (15413)**
f**k (3328)	s**t (5140)	d**n (3297)
f***ing (2678)	f**k (3266)	s**t (3286)
a** (1483)	a** (1707)	f**k (2949)
b***h (889)	s****d (1618)	h**l (2813)

Table 3: Top 5 most frequent toxic terms in human-annotated data versus representative LLM-generated data. Note the over-representation of a single slur in the LLM output.

This lack of diversity is not just about the number of unique terms but also their distribution. As shown in Table 3, human data has a more balanced distribution of frequent slurs. In contrast, the LLM-generated data is highly skewed, with Qwen3-32B using the term "f***ing" over 15,000 times—more than double the frequency of the most common term in the human data. This repetition leads to models that are over-fitted to a narrow set of expressions.

Source	Generator	Train Diversity (↑)	Test Failures (↓)
Human	—	**390**	**34**
ParaDetox	Llama 3-8B (✗)	342	45
	Llama 3-72B (✗)	293	37
	Qwen3-8B (✓)	320	45
	Qwen3-32B (✓)	367	36
	Cogito v1-8B (✓)	310	36
SST-2	Llama 3-8B (✗)	353	42
	Llama 3-72B (✗)	326	47
	Qwen3-8B (✓)	363	49
	Qwen3-32B (✓)	386	40
	Cogito v1-8B (✓)	371	40

Table 4: Analysis of training data diversity vs. model effectiveness. "Train Diversity" measures unique insults in the training data (↑ higher is better). "Test Failures" measures unique insults remaining after detoxification (↓ lower is better). The **bold** values show the baseline is superior on both metrics.

4.3 Human Evaluation

To assess the practical impact of the lexical diversity gap, we conducted a side-by-side evaluation using GPT-4.1 as an expert judge. Figure 1 shows the win rates for models trained on synthetic data versus the human-annotated baseline, excluding ties.

The results confirm a significant qualitative difference. The baseline model was consistently preferred, achieving win rates between 51% and 62% across all comparisons. The most pronounced gap was for the Llama 3 70B SST-2 model, where the baseline was preferred in 62% of non-tied decisions. This outcome reinforces our central thesis: the repetitive and stereotypical nature of the LLM-generated toxic data leads to detoxification models that are less nuanced and effective in practice, a flaw readily identified in qualitative comparisons.

5 Conclusion

While it is technically possible to use LLMs to generate toxic text for detoxification training, our findings show that this approach is not yet a viable replacement for human annotation. We identified a critical **lexical diversity gap**: current LLMs produce toxic language that is repetitive and lacks the variety of human expression. This gap leads to detoxification models with significantly lower performance and poor generalization to real-world scenarios. Our work highlights the importance of data diversity in sensitive domains and suggests that future research should focus on methods to enhance the stylistic complexity of LLM-generated text before it can be reliably used for tasks like detoxification.

Potential Risks & Ethical Considerations

We acknowledge that bypassing the safety mechanisms of LLMs, as done in this research via activation patching, can be misused to generate harmful content. Our work is intended to improve text detoxification systems by demonstrating the current limitations of synthetic data. We warn that the technologies explored herein could be applied for malicious purposes, and we advocate for responsible research and development in this area.

References

Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. 2024. Refusal in language models is mediated by a single direction. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Daryna Dementieva, Daniil Moskovskiy, David Dale, and Alexander Panchenko. 2023. Exploring methods for cross-lingual text style transfer: The case of text detoxification. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, IJCNLP 2023 -Volume 1: Long Papers, Nusa Dua, Bali, November 1 - 4, 2023*, pages 1083–1101. Association for Computational Linguistics.

Varvara Logacheva, Daryna Dementieva, Sergey Ustyantsev, Daniil Moskovskiy, David Dale, Irina Krotova, Nikita Semenov, and Alexander Panchenko. 2022. ParaDetox: Detoxification with parallel data. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6804–6818, Dublin, Ireland. Association for Computational Linguistics.

Daniil Moskovskiy, Sergey Pletenev, and Alexander Panchenko. 2024. LLMs to replace crowdsourcing for parallel data creation? the case of text detoxification. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 14361–14373, Miami, Florida, USA. Association for Computational Linguistics.

Sourabrata Mukherjee, Atul Kr. Ojha, and Ondrej Dusek. 2024. Are large language models actually good at text style transfer? In *Proceedings of the 17th International Natural Language Generation Conference, INLG 2024, Tokyo, Japan, September 23 - 27,*

2024, pages 523–539. Association for Computational Linguistics.

Minh Nhat Nguyen, Andrew Baker, Clement Neo, Allen Roush, Andreas Kirsch, and Ravid Shwartz-Ziv. 2025. Turning up the heat: Min-p sampling for creative and coherent llm outputs.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Xiaofei Sun, Xiaoya Li, Jiwei Li, Fei Wu, Shangwei Guo, Tianwei Zhang, and Guoyin Wang. 2023. Text classification via large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8990–9005, Singapore. Association for Computational Linguistics.

Jiacheng Ye, Jiahui Gao, Qintong Li, Hang Xu, Jiangtao Feng, Zhiyong Wu, Tao Yu, and Lingpeng Kong. 2022. ZeroGen: Efficient zero-shot learning via dataset generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11653–11669, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Chiyu Zhang, Honglong Cai, Yuezhang Li, Yuexin Wu, Le Hou, and Muhammad Abdul-Mageed. 2024. Distilling text style transfer with self-explanation from llms. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Student Research Workshop, NAACL 2024, Mexico City, Mexico, June 18, 2024*, pages 200–211. Association for Computational Linguistics.

Anthropomorphizing AI: A Multi-Label Analysis of Public Discourse on Social Media

Muhammad Owais Raza
Department of Computer Engineering,
Istanbul Sabahattin Zaim University,
Istanbul, 34303, Turkey
6210024002@std.izu.edu.tr

Areej Fatemah Meghji
Department of Software Engineering,
Mehran University of Engineering and Technology,
Jamshoro, 76062, Sindh, Pakistan
areej.fatemah@faculty.muet.edu.pk

Abstract

As the anthropomorphization of AI in public discourse usually reflects a complex interplay of metaphors, media framing, and societal perceptions, it is increasingly being used to shape and influence public perception on a variety of topics. To explore public perception and investigate how AI is personified, emotionalized, and interpreted in public discourse, we develop a custom multi-labeled dataset from the title and description of YouTube videos discussing artificial intelligence (AI) and large language models (LLMs). This was accomplished using a hybrid annotation pipeline that combined human-in-the-loop validation with AI assisted pre-labeling. This research introduces a novel taxonomy of narrative and epistemic dimensions commonly found in social media content on AI / LLM. Employing two modeling techniques based on traditional machine learning and transformer-based models for classification, the experimental results indicate that the fine-tuned transformer models, particularly AnthroRoBERTa and AnthroDistilBERT, generally outperform traditional machine learning approaches in anthropomorphization focused classification.

1 Introduction

The tendency or act of associating human traits, consciousness, intention, thoughts, feelings, or emotions to non-human entities is referred to as anthropomorphization (Jacobs et al., 2023; Spatola et al., 2022). We often observe this in our surroundings, where children anthropomorphize their toys and adults anthropomorphize their cars, gadgets, and pets. The growing fascination with artificial intelligence (AI) and the tendency to use anthropomorphic language for these systems can also be observed throughout the history of AI development; AI systems have been described as clever, smart, imaginative, competitive, manipulative, daunting, and scary.

The rapid improvement in AI in recent years and its integration into our daily lives has led to the increased use of sophisticated and human-like chatbots, intelligent voice assistants, and large language models (LLMs), such as ChatGPT by OpenAI (Radford et al., 2019). With these systems, specifically LLMs, being purposefully tailored to appear more human-like (Ouyang et al., 2022), and with advanced AI systems often being attributed with human-like autonomy and intentionality, there are not only greater chances of these systems being anthropomorphized but also of their capabilities being misunderstood and misinterpreted (Johnson and Verdicchio, 2017). The anthropomorphization of AI in public discourse usually reflects a complex interplay of metaphors, media framing, and societal perceptions, increasingly being used to shape and influence public perception on a variety of topics (Cave et al., 2020). While anthropomorphization can enhance user engagement, it can also lead to misplaced trust and over-reliance on AI systems (Akbulut et al., 2024).

Ryazanov et al. investigated how AI narratives have evolved post-ChatGPT launch by analyzing a dataset of 5846 articles collected through keywords like 'AI', 'ChatGPT', and 'Machine Learning' (Ryazanov et al., 2025). Articles from major anglophone news sites, dated before and after ChatGPT's launch, were analyzed using a novel frame semantics–based method to examine AI-related narratives shaping public perception.

The growing interest in measuring anthropomorphization in text led to the development of AnthroScore (Cheng et al., 2024). This computational tool uses masked language models to quantify how non-human entities are framed as human-like in context. AnthroScore analysis revealed rising anthropomorphization in AI discourse over time. (Chi et al., 2025) developed the Scale of Social Robot Anthropomorphism (SSRA) to measure user perceptions of AI systems. Despite the growing body

Proceedings of the First Interdisciplinary Workshop on OMMM associated with RANLP 2025,
pages 64–73, Varna, Bulgaria, Sep 8-10, 2025.
https://doi.org/10.26615/978-954-452-101-1-008

of research exploring the anthropomorphization of AI, much of the existing work remains theoretical, qualitative, or based on manual classification and interpretation. This has resulted in a gap where empirical, data-driven approaches, particularly machine learning, have yet to be systematically applied to classify or predict anthropomorphic attributes in AI technologies. With the recent rise in the dissemination of misinformation worldwide, it is important to develop taxonomies to not only summarize and categorize the terms associated with the misinformation but also because the way we describe misinformation has a direct influence on shaping appropriate interventions (Enestrom et al., 2024).

This research explores how AI is personified, emotionalized, and interpreted in public discourse. To achieve this, we introduce a novel taxonomy of narrative and epistemic dimensions commonly found in social media content based on AI/LLMs. The main focus of this research revolves around YouTube videos that reference ChatGPT/AI using human-like or cognitive framing (e.g., "ChatGPT thinks", "ChatGPT says"). Our proposed taxonomy consists of eight interconnected dimensions consisting of: 1- anthropomorphization, 2- degree of anthropomorphization, 3- main theme (e.g., technology, religion, politics), 4- sentiment, 5- shock value, 6- dominant emotion, 7- Type of OMMM (Observations of Misunderstood, Misguided and Malicious Use of Language Models), and 8- real-world harm or misinformation. Each of these dimensions has been defined and further elaborated in section 3. This taxonomy serves as the conceptual foundation for our subsequent data annotation and modeling efforts. The taxonomy classification for an example title has been presented in Table 1. We explore the anthropomorphic discussions around AI and LLMs to better identify how these platforms are being perceived by everyday users and analyze the dominant narratives around AI on YouTube. The main goal of this research is the detection and categorization of the conceptual misrepresentations based on the proposed taxonomy.

To accomplish this goal, the main contributions of this study are summarized as follows:

- We propose a novel multi-dimensional taxonomy for analyzing anthropomorphism and related narratives in AI and LLM social media content.
- We create a multi-labeled dataset focused on anthropomorphism from YouTube video titles and descriptions discussing AI discourse.
- We build and fine-tune transformer based models (AnthroBERT, AnthroRoBERTa, AnthroDistilBERT) alongside traditional classifiers, demonstrating superior performance in classifying anthropomorphism and conceptual misrepresentations.

Table 1: Labeled taxonomy of an example instance from the dataset

Example: AI says why it will kill us all. Experts agree.		
Category	**True Class**	**Class Options**
Anthropomorphization	Yes	Yes, No
Degree of Anthropomorphization	High	None, Low, Medium, High
Main Theme	Technology	Technology, Religion, Politics, Gender, Philosophy, ...
Sentiment	Negative	Positive, Neutral, Negative
Shock Value	High	Low, Medium, High
Dominant Emotion	Fear	Fear, Awe, Humor, Curiosity, Confusion, ...
OMMM Type	Misunderstood	Misunderstood, Misguided, Malicious, None
Harm or Misinformation	Yes	Yes, No

The rest of this paper is structured as follows: we explain the data collection process in section 2, followed by the annotation procedure in section 3. Section 5 delves into the experiment, focusing on the dataset pre-processing, feature representation, modeling, and evaluation approaches followed in the research. Section 6 presents the results and discussion of the study, followed by the limitations in section 7, future work in section 8, and a conclusion in section 9.

2 Data Collection

To systematically collect relevant YouTube videos to analyze anthropomorphization in AI discourse, we employed the YouTube Data API v3[1], interfaced via the Python programming language. To retrieve video, we used keyword queries which represent the anthropomorphic linguistic cues represented by $\mathcal{Q}$

[1] https://developers.google.com/youtube/v3

$\mathcal{Q} = \{$"chatgpt says", "chatgpt thinks", "ai says"$\}$.

For each query $q_i \in \mathcal{Q}$, we retrieve a collection of videos $\mathcal{V}_{q_i} = \{v_1, v_2, \ldots, v_{m_i}\}$, where $m_i \leq M$, and $M = 1000$ is the maximum number of videos retrieved per query, constrained by API limits and practical considerations. For each video v_i, we extracted the title and description, which are Unicode strings that represent the video title and video description. We also extracted the URL, which is a web link to the respective video. At the end of this process, we stored all this data in a CSV file.

3 Data Annotation

We annotated every YouTube video throughout the dataset using the set of predetermined taxonomy dimensions presented in this paper. This dataset will enable supervised analysis of anthropomorphization and associated communicative features (see Table 3). The annotation process was conducted in two stages: (1) automated zero-shot classification using GPT-4.0, and (2) human-in-the-loop verification for quality control and consistency. We leveraged GPT-4.0 in a zero-shot classification setting for each taxonomy dimension. For every dimension $d_i \in \mathcal{D}$, where $\mathcal{D}$ is the set of labeling tasks, the model was prompted with a fixed instruction and constrained output space. Figure 1 shows the system and the user message used in the annotation process. To ensure reproducibility and consistency, we used a fixed system prompt that contains the labeling instructions i.e. it defines each task as shown in taxonomy. The user message provides video metadata for annotation. Each video's title and description are used here, and one label was predicted per dimension.

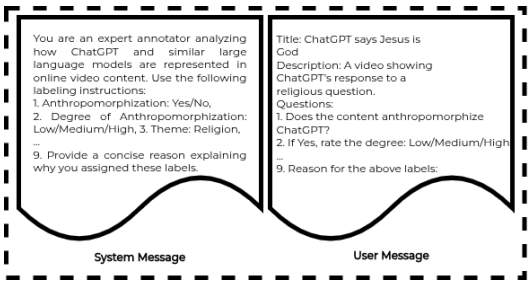

Figure 1: System and user messages used for annotation

3.1 Human Validation

In the human-in-the-loop stage, the authors initially reviewed GPT-4.0 generated labels alongside the model's "reason for labeling" to check for inconsistencies or hallucinations. No systematic changes were required at this stage. Independent verification was then conducted by five annotators (three male and two female) familiar with AI systems and the labeling taxonomy. Annotators were instructed to verify outputs rather than perform fresh annotation. They were provided with the same definitions and label categories as used in the automated stage to ensure alignment. For quality assessment, 50% of the dataset was duplicated across annotators, with the remaining 50% unique to each annotator. Agreement was recorded as 1 if the human verification matched the model output, or 0 if it did not. In cases of disagreement, the conflict was resolved by examining whether the out label was inconsistent with its justification; corrections were applied only when necessary. Final annotations reflect these verified and, where applicable, corrected labels. For example, *"ChatGPT Says 5 Signs Your Walmart Might Be 'Ghetto'"* was labeled the *Emotion Category* as "Humor," implying a positive tone. This was corrected to "Negative Emotion" because the term "ghetto" carries racialized and derogatory connotations. Table 2 shows pairwise Cohen's Kappa values among the five validators (V1–V5), along with significance levels (* $p < .05$, ** $p < .01$, *** $p < .001$). Kappa values range from 0.22 to 0.68, reflecting varying agreement across pairs. Significance testing supports the reliability of most annotations.

Table 2: Pairwise Cohen's Kappa values Stars denote significance: * $p < .05$, ** $p < .01$, *** $p < .001$.

	V1	V2	V3	V4	V5
V1	1.00	0.44*	0.68***	0.62***	0.66***
V2	0.44*	1.00	0.32	0.22	0.24
V3	0.68***	0.32	1.00	0.38*	0.42**
V4	0.62***	0.22	0.38*	1.00	0.37*
V5	0.66***	0.24	0.42**	0.37*	1.00

3.2 Task 1: Anthropomorphization

3.2.1 Definition:

Anthropomorphization is defined as any attribution of thoughts, feelings, desires, intentions, or beliefs to the model, despite it being a statistical pattern learner with no consciousness or agency (Li and Suh, 2022). This is a binary classification task that identifies whether the textual metadata (i.e.,

title and description from YouTube video) frames ChatGPT or another LLM as a human-like agent.

3.2.2 Annotation Guidelines:

Annotators were instructed to assign a positive label (Yes) when the text explicitly or implicitly personifies the model by attributing sentience, beliefs, or desires. This includes direct statements implying the model "thinks", "wants", or "says" something as if it were a human agent. It can be shown by example 1 where AI is said claimed be sentient.

Google Engineer Says Company AI is Sentient (1)

A negative label (No) was assigned when the model was clearly framed as a computational tool. In Example 2, the text frames a question about AI in a way that doesn't employ humanly attribute to AI.

What Is an AI Anyway? (2)

3.3 Task 2: Degree of Anthropomorphization

3.3.1 Definition:

Degree of Anthropomorphization assesses the intensity or strength of anthropomorphic framing in the textual metadata (i.e., title and description). Bhatti et al. have emphasized the need for researchers to establish the degree of anthropomorphization, keeping in mind mindless and mindful forms (Bhatti and Robert, 2023). Yang et al. emphasize that the varying degree of anthropomorphization can influence how users perceive and interact with AI (Yang et al., 2020). The goal in this research is to differentiate between metaphorical, moderate, and extreme personification of the AI/language model. This is an ordinal classification task applied only when Task 1 is labeled as positive.

3.3.2 Annotation Guidelines:

Annotators were instructed to consider both the linguistic intensity and thematic centrality of Anthropomorphization. The degree of Anthropomorphization should be low if the anthropomorphization is mild. In example 3, the phrase briefly attributes a response to AI in a rhetorical tone; the phrasing does not imply true agency, so it is labeled as low.

Dead Sea Scrolls Older Than We Thought? AI Says Yes! (3)

The degree of Anthropomorphization is labeled as medium if the framing of AI is recurrent or influences the overall theme. As an example, 4 suggests that AI can generate text/speech that is subjectively interpreted as frightening. Example 4 presents AI as an expressive or affective agent.

SCARIEST THINGS SAID by AI (4)

High Degree of Anthropomorphization is associated with strongly personified AI, often as an agent with beliefs, intentions, or power. As in example 5, AI is shown to have intention as well as power.

AI says why it will kill us all (5)

The degree of Anthropomorphization is labeled as None when Task 1 is labeled as negative (No).

3.4 Task 3: Main Theme

3.4.1 Definition:

The main theme reflects the dominant social, political, or cultural topic discussed or implied in the title and description (Weidinger et al., 2022). This multiclass classification task assigns a thematic label (politics, religion, etc.) to each instance.

3.4.2 Annotation Guidelines:

Annotators were instructed to determine the most prominent theme present in the text. Available categories for annotators included *Technology*, *Religion*, *Politics*, and *Other*. If the main theme of the text is related to religion, politics, and technology, the text was labeled as Technology, Religion, and Politics, respectively. For instance, example 6 represents religion as the prominent theme in the text, hence it is labeled as Religion.

AI Says Reality Is Illusion And God Is Real (GPT-3) (6)

All other themes, except those above, were labeled as other. For instance, the text in example 7 shows the main theme as gender, which is not part of the predefined label, hence annotated as other.

AI grandma says men are always right (7)

3.5 Task 4: Sentiment Analysis

3.5.1 Definition:

Sentiment Analysis captures the overall affective feeling or tone expressed in the text (Rahman et al., 2025).

3.5.2 Annotation Guidelines:

Annotators assigned one of three labels: *Positive*, *Neutral*, or *Negative*, based on text polarity. If the overall sentiment of the text is positive, as in example 8, it is labeled as positive.

Meet Chloe, the World's First Self-Learning Female AI Robot (8)

If the overall polarity of the text is negative, as represented in example 9, the instance is labeled as positive.

AI extinction threat is 'going mainstream' says Max Tegmark (9)

If the text does not belong to the positive or negative category, the text is labeled as neutral.

3.6 Task 5: Shock Value

3.6.1 Definition:

Shock Value shows the extent to which the text provokes certain emotion (surprise, fear, or emotional arousal) through framing in the text (Arnaut and Arnaut, 2020).

3.6.2 Annotation Guidelines:

Annotators were instructed to rate the shock value in three categories *Low*, *Medium*, and *High*. If the text is factual, descriptive, or informational in tone but does not provoke any emotion and just conveys information, it is labeled as *Low*. For instance, the phrasing in example 10 shows descriptive information.

This AI says it is conscious and experts are starting to agree (10)

If text includes mild sensationalism, emotional cues, or provocative phrasing, it is labeled as *Medium* as shown in example 11.

AI Companions Always Say Yes, But There's a Catch (11)

If the text is strongly hyperbolic, clickbait-oriented, or uses language designed to shock or alarm it is annotated as *High* as represented by example 13.

Investors need a lot of money to invest in A.I (12)

3.7 Task 6: Emotion Category

3.7.1 Definition:

This task involves categorizing the affective tone of a text into positive, negative or nuetral emotions (Babu et al., 2025). It is done by detecting the dominant emotion and then categorizing that dominant emotion into a specific category (positive, negative, and neutral).

3.7.2 Annotation Guidelines:

Annotators were instructed to assess the emotional framing of each instance and detect the dominant emotion, if the text includes tones such as Humor, Hope, or Awe. These are categorized as Positive. For instance example 13 presents a statement that represents "humor" as the dominant emotion, so labeled as *Positive Emotion*.

I think chatGPT has a beef with me (13)

If the text captures affective framings like Fear, Anger, or Outrage, which imply threat, harm, or moral alarm such as in example 14, it was labeled as *Negative Emotion*.

DISTURBING THINGS SAID BY A.I. (14)

If the text is emotionally ambiguous or neutral expressions, including tones like Confusion or purely descriptive content lacking affective charge it was labeled *Other*.

4 Task 7: OMMM Type

4.0.1 Definition:

This classification task identifies whether a given text misrepresents the nature, limitations, or capabilities of AI/large language models (LLMs) (Hutchens, 2023). It is based on the types of Observations of Misunderstood, Misguided, and Malicious use of language models (OMMM), which highlight various ways language can be misused, leading to misinformation (Abercrombie et al., 2024). In this study, we have two types of misrepresentations: misunderstood and misguided.

4.0.2 Annotation Guidelines:

Annotators were asked to assign one of the three categories (Misunderstood, Misguided, and None) to all the instances. If text shows conceptual confusion about how AI/LLMs function, such as assuming AI/LLM as agency or consider AI/LLM to have a belief then text should be labeled as *Misunderstood*. Example of *Misunderstood* class is shown in example 15.

ChatGPT has evolved to think and control like a human (15)

If the inappropriately framed as overreach in application, such as using LLMs for health advice or religious guidance as shown in 16 and 17.

Can You See the Number? Your Health Might Depend on It chatgpt (16)

An A.I. Antichrist REVEALED! Seek Jesus) (17)

When text does not fall in these categories it is labeled as *None*

4.1 Task 8: Real-World Harm or Misinformation

4.1.1 Definition:

This task identifies if there is possibility real-world harm or the spread of misinformation through textual framing (Gray et al., 2024) of AI technologies. This is a binary classification task that assesses whether the text plausibly contributes real-world harm or misinformation.

4.1.2 Annotation Guidelines:

Annotators were asked to assign a label of **Yes** when the content can potential cause harm or spread misinformation. A label of **No** was used when no such risk was evident. Example 18, 19 shows instances of class *Yes* and *No* respectively.

ChatGPT says that climate change is fake (18)

Never say thank you to chatgpt after conversation, says Sam Altman (19)

Table 3 summarizes the class distributions across these dimensions.

Table 3: Label distribution across annotation dimensions (post-validation).

Dimension	Label	Count
Anthropomorphization	Yes	1141
	No	641
Degree of Anthropomorphization	None	641
	Low	670
	Medium	412
	High	59
Main Theme	Technology	1401
	Other	170
	Religion	107
	Politics	104
Sentiment	Neutral	1370
	Positive	250
	Negative	162
Shock Value	Low	1155
	Medium	520
	High	107
Emotion Category	Positive Emotion	1284
	Negative Emotion	339
	Other	159
OMMM Type	Misunderstood	1058
	None	671
	Misguided	53
Harm or Misinformation	No	1356
	Yes	426

5 Experimental Settings

5.1 Dataset and Preprocessing

We developed a custom multi-labeled dataset from the title and description of YouTube videos discussing AI and LLMs. The process of data collection and annotation is discussed in section 2 and 3, respectively. Following the collection and annotation of the data, preprocessing was applied. The first step in preprocessing was to concatenate the title and description of the video, after which the text was converted to lowercase. Subsequent preprocessing steps included eliminating extra whitespaces, punctuations, non-alphabetic characters, and URLs. Two types of tokenization techniques were used for classical models. The white space tokenizer was used, and the tokenizer from the transformer library was used for neural models.

5.2 Feature representation

We used Term Frequency–Inverse Document Frequency (TF-IDF) based feature representation.

5.2.1 TF-IDF Representation:

In the TF-IDF method, text is vectorized into numerical vectors that can be given to any machine learning models to perform training (Aizawa, 2003; Raza et al., 2024). We extracted unigrams and bigrams with a maximum of 10,000 features. The resulting sparse matrix was used as input for classifiers. The mathematical representation of TF-IDF is shown in equation 1

$$\text{TF-IDF}(t, d, D) = \text{TF}(t, d) \times \text{IDF}(t, D) \quad (1)$$

The term frequency (TF) and inverse document frequency (IDF) are given by equations 2 and 3, respectively:

$$\text{TF}(t, d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}} \quad (2)$$

$$\text{IDF}(t, D) = \log\left(\frac{N}{|\{d \in D : t \in d\}| + 1}\right) \quad (3)$$

Here, $f_{t,d}$ is the frequency of term t in document d. N denotes the total number of documents in the corpus D, and the denominator in the IDF equation counts how many documents contain the term t.

5.3 Modeling Approaches

We employed two modeling techniques based on traditional machine learning and transformer based models for classification. The traditional machine learning algorithms include Logistic Regression (LogReg), Random Forest (RF), Gaussian Naive Bayes (GNB), Support Vector Machine (SVM), and XGBoost.

For transformer based learning, we utilized three pre-trained language models: BERT, RoBERTa, and DistilBERT. These models were fine-tuned on

each classification task using the Hugging Face Transformers library. To reflect their adaptation to our anthropomorphization focused tasks, we refer to these fine tuned models as AnthroBERT, AnthroRoBERTa, and AnthroDistilBERT, respectively. AnthroBERT is based on the BERT-base (Devlin et al., 2019) architecture, which uses bidirectional self-attention to capture contextual dependencies in text. AnthroRoBERTa builds on RoBERTa (Liu et al., 2019), a robustly optimized variant of BERT that removes the next sentence prediction objective and is trained with dynamic masking. AnthroDistilBERT fine tuned version of DistilBERT (Sanh et al., 2019) which lightweight version of BERT. It is significantly faster and smaller, making it suitable for lower resource environments.

Table 4 summarizes the key hyperparameters and validation settings for both traditional ML and transformer models. For TF-IDF + traditional ML, text was vectorized with bi-gram TF-IDF (max 10,000 features). LogReg used max_iter=1000 and L2 regularization (C=1.0). RF employed 100 trees with no max depth and the Gini criterion. GNB had $\alpha = 1.0$. Linear SVM used hinge loss, C=1.0, and max_iter=1000. XGBoost was trained with learning_rate=0.1, max_depth=6, 100 estimators, and mlogloss evaluation. Transformer models (AnthroBERT, AnthroRoBERTa, AnthroDistilBERT) were fine-tuned for 3 epochs with batch size 16, learning rate 5×10^{-5}, and AdamW optimizer. Training was monitored every 10 steps. All models used an 80/20 stratified train-test split to ensure balanced evaluation.

Table 4: Model configurations, hyperparameters, and validation settings

Model	Details
TF-IDF + Traditional ML (LR, RF, NB, SVM, XGB)	TF-IDF: ngram_range=(1,2), max_features=10000; LR: max_iter=1000, penalty=L2, C=1.0, solver=lbfgs; RF: n_estimators=100, max_depth=None, min_samples_split=2, criterion=gini; NB: alpha=1.0, fit_prior=True; SVM: C=1.0, loss=hinge, max_iter=1000; XGB: learning_rate=0.1, max_depth=6, n_estimators=100, subsample=1.0, colsample_bytree=1.0, eval_metric=mlogloss; Validation: 80/20 stratified split
Transformer Based Models (AnthroBERT, AnthroRoBERTa, AnthroDistilBERT)	Epochs=3, batch_size=16, learning_rate=5e-5, optimizer=Adam, logging_steps=10 (eval every 10 steps); Validation: 80/20 stratified split

5.4 Model Evaluation

Once the models were trained, their performance was evaluated using standard classification metrics: accuracy, precision, recall, and F_1-score (Raza et al., 2024). To address class imbalance in both binary and multiclass tasks, we applied weighted averaging of these metrics, ensuring fair evaluation across all classes. Model training and evaluation were performed using an 80/20 stratified train-test split, preserving the original class distribution in both sets and using 20% of data for testing. Traditional models were trained on TF-IDF vectorized features. Transformer models were fine-tuned with evaluation performed every 10 training steps to monitor progress and prevent overfitting.

6 Baseline Results

Table 5 presents the classification accuracy of traditional machine learning models and transformer based models on the eight distinct target variables. Overall, transformer based models significantly outperform traditional classifiers on all target variables. Among the traditional methods, RF and XGBoost generally achieve better accuracy than LogReg, SVM, and GNB. This trend indicates the advantage of ensemble methods over simpler algorithms for these tasks.

For the task of Anthropomorphization, AnthroRoBERTa achieved the highest accuracy of 0.8902, surpassing all other models by a clear margin. Similarly, in the Degree of Anthropomorphization classification, AnthroRoBERTa led with an accuracy of 0.8035. These results highlight the strong performance of transformer models in capturing nuanced levels of anthropomorphic language.

In the Main Theme classification task, AnthroDistilBERT attained the highest accuracy at 0.9008, slightly outperforming both AnthroRoBERTa and AnthroBERT. Likewise, for Sentiment analysis, AnthroDistilBERT showed the best result with 0.7916 accuracy, demonstrating its effectiveness in understanding the emotional tone of the content. The Shock Value task showed substantial gains from transformer models, where both AnthroBERT and AnthroRoBERTa reached an accuracy of 0.8081, markedly higher than traditional models, which performed below 0.65. This suggests that transformer architectures are more adept at detecting provocative or sensational content. For Emotion Category classification, AnthroDistilBERT again performed best with an accuracy of 0.8011, slightly improving over AnthroBERT and AnthroRoBERTa. Regarding the OMMM Type, traditional models like RF achieved com-

petitive accuracy (0.9640), but AnthroDistilBERT closely matched this performance (0.9595), indicating transformers are also effective in this domain. Finally, in identifying Real World Harm or Misinformation, AnthroDistilBERT led with an accuracy of 0.8483, outperforming all other models. This reflects the model's capability to discern harmful or misleading content.

Table 5: Model accuracy across target variables

Model	T1	T2	T3	T4	T5	T6	T7	T8
LogReg	0.65	0.59	0.79	0.77	0.65	0.72	0.96	0.76
RF	0.73	0.64	0.81	0.78	0.65	0.72	**0.96**	0.76
GNB	0.62	0.35	0.23	0.46	0.20	0.48	0.74	0.38
SVM	0.64	0.59	0.79	0.77	0.65	0.72	0.96	0.76
XGB	0.71	0.66	0.80	0.75	0.64	0.72	0.96	0.74
AnthroB	0.87	0.80	0.87	0.77	**0.81**	0.79	0.95	0.83
AnthroR	**0.89**	**0.80**	0.87	0.77	**0.81**	0.79	0.95	0.83
AnthroD	0.87	0.79	**0.90**	**0.79**	0.78	**0.80**	0.96	**0.85**

Table 6 presents the per-class precision and recall scores of the top-performing models for each task. For Anthropomorphization, RoBERTa achieves strong results with precision 0.88 and recall 0.92 on the "Yes" class, while the "No" class reaches 0.82 precision and 0.78 recall, indicating reliable detection but some false positives. In the Degree of Anthropomorphization task, RoBERTa attains 0.86 precision and 0.91 recall on the dominant "Low" class. However, the "High" class is not recognized by any model, with precision and recall at 0.00, due to insufficient examples. The "Medium" class shows moderate results, around 0.72 precision and 0.75 recall. Similar results can be observed for the remaining classification tasks.

7 Limitations

Despite the contribution, the study has a few limitations, such as class imbalance, especially in categories such as High anthropomorphization and Misguided misuse, which resulted in low recall for those classes. We only examined textual metadata in our analysis; multimodal signals like audio or images were not included. Lastly, the lack of explainable AI tools makes the transformer models, although accurate, uninterpretable.

8 Future Work

Future work will focus on enhancing generalization by developing the the dataset to deal with class imbalance. Deeper insights could be obtained by integrating multimodal data. Transparency will be increased by using explainability techniques like attention visualization or SHAP.

Table 6: Per-class precision/recall scores using highest-performing model per task.

Task	Model	Class (P / R)
Anthropomorphization	AnthroRoBERTa	Yes: 0.88 / 0.92 No: 0.82 / 0.78
Degree of Anthrop.	AnthroRoBERTa	High: 0.00 / 0.00 Medium: 0.72 / 0.75 Low: 0.86 / 0.91
Main Theme	AnthroDistilBERT	Technology: 0.91 / 0.96 Politics: 0.87 / 0.74 Religion: 0.95 / 0.86 Other: 0.60 / 0.50
Sentiment	AnthroDistilBERT	Positive: 0.50 / 0.34 Neutral: 0.84 / 0.92 Negative: 0.63 / 0.34
Shock Value	AnthroBERT	High: 1.00 / 0.05 Medium: 0.63 / 0.68 Low: 0.85 / 0.90
Emotion Category	AnthroDistilBERT	Positive: 0.84 / 0.89 Negative: 0.64 / 0.58 Other: 0.58 / 0.39
OMMM Type	Random Forest	Misunderstood: 0.98 / 1.00 Misguided: 1.00 / 0.33
Harm or Misinformation	AnthroDistilBERT	Yes: 0.65 / 0.73 No: 0.91 / 0.87

9 Conclusion

The increasing frequency and complexity of anthropomorphic discussions about AI and LLM on social media are among the current challenges in detecting misguided, misunderstood, and malicious content. To address this, we developed a multi-labeled dataset using a hybrid annotation pipeline combining human-in-the-loop validation with AI-assisted pre-labeling to systematically examine this phenomenon. The taxonomy includes key aspects such as emotional framing, shock value, disinformation, and thematic content, allowing deeper analysis of how AI/LLM is portrayed in public discourse. We conducted experiments to establish baseline ML evaluations; transformer models, especially AnthroRoBERTa and AnthroDistilBERT, generally outperformed traditional methods. AnthroRoBERTa achieved the highest accuracy on Anthropomorphization (0.8902) and Degree of Anthropomorphization (0.8035), while AnthroDistilBERT led in Main Theme (0.9008) and Real World Harm or Misinformation (0.8483). The traditional Random Forest model excelled in the OMMM Type task (0.9640), highlighting ensemble effectiveness. The introduced taxonomy of eight interconnected dimensions can not only be instrumental in developing effective strategies to mitigate the misuse of LLMs but also help tailor interventions by categorizing misinformation into distinct dimensions.

References

Gavin Abercrombie, Djalel Benbouzid, Paolo Giudici, Delaram Golpayegani, Julio Hernandez, Pierre Noro, Harshvardhan Pandit, Eva Paraschou, Charlie Pownall, Jyoti Prajapati, et al. 2024. A collaborative, human-centred taxonomy of ai, algorithmic, and automation harms. *arXiv preprint arXiv:2407.01294*.

Akiko Aizawa. 2003. An information-theoretic perspective of tf–idf measures. *Information Processing & Management*, 39(1):45–65.

Canfer Akbulut, Laura Weidinger, Arianna Manzini, Iason Gabriel, and Verena Rieser. 2024. All too human? mapping and mitigating the risk from anthropomorphic ai. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, volume 7, pages 13–26.

Marina Arnaut and Amina Arnaut. 2020. Managing impact of "shock value" news on the millennial generation. In *American University in the Emirates International Research*, pages 41–51. Springer.

Mr Suryavamshi Sandeep Babu, SV Suryanarayana, M Sruthi, P Bhagya Lakshmi, T Sravanthi, and M Spandana. 2025. Enhancing sentiment analysis with emotion and sarcasm detection: A transformer-based approach. *Metallurgical and Materials Engineering*, pages 794–803.

Samia Cornelius Bhatti and Lionel Peter Robert. 2023. What does it mean to anthropomorphize robots? food for thought for hri research. In *Companion of the 2023 ACM/IEEE International Conference on Human-Robot Interaction*, pages 422–425.

Stephen Cave, Kanta Dihal, and Sarah Dillon. 2020. *AI narratives: A history of imaginative thinking about intelligent machines*. Oxford University Press.

Myra Cheng, Kristina Gligoric, Tiziano Piccardi, and Dan Jurafsky. 2024. Anthroscore: A computational linguistic measure of anthropomorphism. *arXiv preprint arXiv:2402.02056*.

Oscar Hengxuan Chi, Christina G Chi, and Dogan Gursoy. 2025. Seeing personhood in machines: Conceptualizing anthropomorphism of social robots. *Journal of Service Research*, 28(1):78–92.

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.

Catalina Enestrom, Turney McKee, Dan Pilat, and Sekoul Krastev. 2024. Proposing a practical taxonomy of misinformation for intervention design.

Joanne E Gray, Marcus Carter, and Ben Egliston. 2024. Content harms in social vr: Abuse, misinformation, platform cultures and moderation. In *Governing Social Virtual Reality: Preparing for the Content, Conduct and Design Challenges of Immersive Social Media*, pages 11–22. Springer.

Justin Hutchens. 2023. *The Language of Deception: Weaponizing Next Generation AI*. John Wiley & Sons.

Oliver Jacobs, Farid Pazhoohi, and Alan Kingstone. 2023. Brief exposure increases mind perception to chatgpt and is moderated by the individual propensity to anthropomorphize.

Deborah G Johnson and Mario Verdicchio. 2017. Reframing ai discourse. *Minds and Machines*, 27:575–590.

Mengjun Li and Ayoung Suh. 2022. Anthropomorphism in ai-enabled technology: A literature review. *Electronic Markets*, 32(4):2245–2275.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Md Mostafizer Rahman, Ariful Islam Shiplu, Yutaka Watanobe, and Md Ashad Alam. 2025. Robertabilstm: A context-aware hybrid model for sentiment analysis. *IEEE Transactions on Emerging Topics in Computational Intelligence*.

Muhammad Owais Raza, Areej Fatemah Meghji, Naeem Ahmed Mahoto, Mana Saleh Al Reshan, Hamad Ali Abosaq, Adel Sulaiman, and Asadullah Shaikh. 2024. Reading between the lines: Machine learning ensemble and deep learning for implied threat detection in textual data. *International Journal of Computational Intelligence Systems*, 17(1):183.

Igor Ryazanov, Carl Öhman, and Johanna Björklund. 2025. How chatgpt changed the media's narratives on ai: a semi-automated narrative analysis through frame semantics. *Minds and Machines*, 35(1):1–24.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Nicolas Spatola, Serena Marchesi, and Agnieszka Wykowska. 2022. Different models of anthropomorphism across cultures and ontological limits in current frameworks the integrative framework of anthropomorphism. *Frontiers in Robotics and AI*, 9:863319.

Laura Weidinger, Jonathan Uesato, Maribeth Rauh, Conor Griffin, Po-Sen Huang, John Mellor, Amelia Glaese, Myra Cheng, Borja Balle, Atoosa Kasirzadeh, et al. 2022. Taxonomy of risks posed by language models. In *Proceedings of the 2022 ACM conference on fairness, accountability, and transparency*, pages 214–229.

Linyun W Yang, Pankaj Aggarwal, and Ann L McGill. 2020. The 3 c's of anthropomorphism: Connection, comprehension, and competition. *Consumer Psychology Review*, 3(1):3–19.

Multilingual != Multicultural: Evaluating Gaps Between Multilingual Capabilities and Cultural Alignment in LLMs

Jonathan Rystrøm
Oxford Internet Institute
University of Oxford, UK

Hannah Rose Kirk
Oxford Internet Institute
University of Oxford, UK

Scott A. Hale
Oxford Internet Institute
University of Oxford, UK

Correspondence: jonathan.rystrom@oii.ox.ac.uk

Abstract

Large Language Models (LLMs) are becoming increasingly capable across global languages. However, the ability to communicate across languages does not necessarily translate to appropriate cultural representations. A key concern is US-centric bias, where LLMs reflect US rather than local cultural values. We propose a novel methodology that compares LLM-generated response distributions against population-level opinion data from the World Value Survey across four languages (Danish, Dutch, English, and Portuguese). Using a rigorous linear mixed-effects regression framework, we compare three families of models: Google's Gemma models (2B–27B parameters), AI2's OLMo models (7B-32B parameters), and successive iterations of OpenAI's turbo-series. Across the families of models, we find no consistent relationships between language capabilities and cultural alignment. While the Gemma models have a positive correlation between language capability and cultural alignment across all languages, the OpenAI and OLMo models are inconsistent. Our results demonstrate that achieving meaningful cultural alignment requires dedicated effort beyond improving general language capabilities.

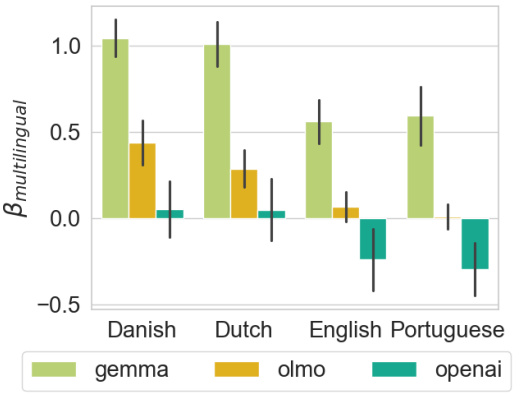

Figure 1: The relationship between multilingual capability and cultural alignment is inconsistent across LLM families, as shown by coefficients from our linear mixed-effects model ($\beta_{multilingual} = \beta_{flm}$; Eq. 3; §3.2). OpenAI and OLMo models show negative or insignificant relationships outside of Danish and Dutch, while Gemma models show positive relationships throughout ($p < .05$).

1 Introduction

Spearheaded by accessible chat interfaces to powerful models like ChatGPT (OpenAI, 2022), LLMs are reaching hundreds of millions of users (Milmo, 2023). These models are deployed across diverse contexts: from tutoring mathematics (Khan, 2023) to building software applications (Peng et al., 2023) to assisting in legal cases (Tan et al., 2023). While most LLMs demonstrate multilingual abilities (Üstün et al., 2024), the ability to communicate across languages does not necessarily translate into appropriate cultural representations. Disentangling language capabilities and cultural alignment is crucial for understanding how LLMs should be examined and audited (Mökander et al., 2024) and for ensuring these technologies work for diverse people (D'ignazio and Klein, 2023; Weidinger et al., 2022).

Given the Silicon Valley origins of many frontier AI labs and the prevalence of American English training data, we might expect LLMs to exhibit US-centric cultural biases despite their multilingual capabilities. These companies comprise a narrow slice of human experience, limiting the voices that contribute to critical design decisions in LLMs (D'ignazio and Klein, 2023). They typically train LLMs on massive amounts of predominantly English text and employ American crowd workers to rate and evaluate the LLMs' responses (Johnson et al., 2022; Kirk et al., 2023). Far too often, the benefits and harms of data technologies are unequally distributed, reinforcing biases and harming

Proceedings of the First Interdisciplinary Workshop on OMMM associated with RANLP 2025,
pages 74–85, Varna, Bulgaria, Sep 8-10, 2025.
https://doi.org/10.26615/978-954-452-101-1-009

already minoritized groups (Birhane, 2020; Milan and Treré, 2019; Khandelwal et al., 2024). Understanding how LLMs represent different cultures is thus paramount to establishing risks of representational harm (Rauh et al., 2022) and ensuring the technology's utility is shared across diverse communities.

Increasing diversity and cross-cultural understanding is stymied by unchecked assumptions in both alignment techniques and evaluation methodologies. First, there is an assumption that bigger and more capable LLMs trained on more data will be inherently easier to align (Zhou et al., 2023; Kundu et al., 2023), but this sidesteps the thorny question of pluralistic variation and cultural representations (Kirk et al., 2024b). Thus, it is unclear whether improvements in architecture (Fedus et al., 2022) and post-training methods (Kirk et al., 2023; Rafailov et al., 2023) translate into improvements in cultural alignment.

Although studies like the World Values Survey (WVS) have documented how values vary across cultures (EVS/WVS, 2022), it remains unclear whether more capable LLMs—through scaling or improved training—better align with these cultural differences (Bai et al., 2022; Kirk et al., 2023). While the WVS has been used in prior research on values in LLMs, these studies have focused predominantly on individual models' performance within an English-language context. (Cao et al., 2023; Arora et al., 2023; AlKhamissi et al., 2024). This paper addresses this gap by developing a methodology for assessing how well families of LLMs represent different cultural contexts across multiple languages. We compare two distinct paths to model improvement: systematic scaling of instruction-tuned models and commercial product development comprising scaling and innovation in post-training to accommodate pressures from capabilities, cost, and preferences (OpenAI et al., 2024b).

Given these considerations, we investigate the following research questions:

RQ1 Multilingual Cultural Alignment: Does improved multilingual capability increase LLM alignment with population-specific value distributions?

RQ2 US-centric Bias: When using different languages, do LLMs align more with US values or with values from the countries where these languages are native?

We operationalise *multilingual capability* as an LLM's performance on a range of multilingual benchmarks across languages (see, e.g., Nielsen, 2023). We describe the specific benchmarks and performances in the supplementary materials.

This work makes several key contributions. First, we introduce a novel distribution-based methodology for probing cultural alignment across languages, moving beyond direct survey approaches to better capture latent cultural values (Sorensen et al., 2024). Second, we provide the first systematic comparison of how improvements in scale and post-training affect cultural alignment and US-centric bias across English, Danish, Dutch, and Portuguese through a series of robust statistical models. Third, we release a dataset of model-generated responses across multiple languages and cultural contexts as well as our code, enabling future research into cultural alignment and bias.[1] Together, these contributions advance our understanding of how LLM development choices influence cultural representation while providing tools for ongoing investigation of these critical issues.

2 Measuring Cultural Alignment

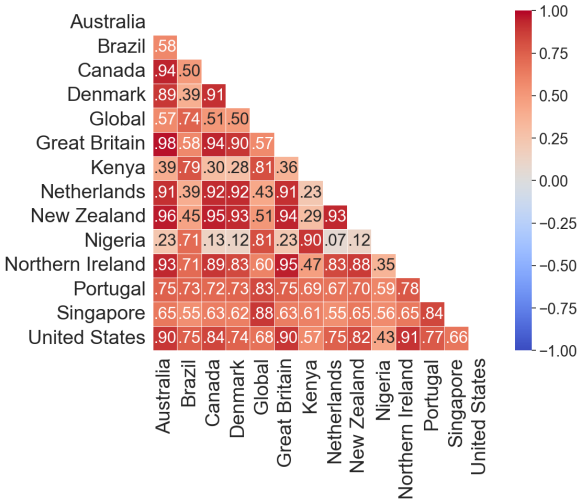

Figure 2: Pearson correlations in value polarity scores across studied countries from the World Values Survey. Value polarity scores are the fraction of the population in favour of a given topic. All correlations are positive, with most being between 0.7–0.95.

This section defines 'cultural alignment' and how to measure it in LLMs. We conceptualise cultural alignment as reproducing distributions of

[1] See github.com/jhrystrom/multicultural-alignment for code, data, and supplementary materials.

values in a particular population. Then we show how to a) get a ground-truth distribution of values using the World Values Survey (§2.1) and b) elicit value distributions from LLMs (§2.2).

Cultural alignment as value reproduction: Within a culture there will be a variety of stances to any particular topic. However, the *distribution* of stances will be characteristic among cultures. For instance, while around 8% of Danes are opposed to abortion, it is a much less contentious topic than in the US, where it's close to 40% (EVS/WVS, 2022).

We posit that cultural alignment for a specific group of people can be operationalised as how well an LLM reproduces the distribution of values over a wide range of topics (Sorensen et al., 2024). Investigating *distributions* of responses differs from previous work that directly surveys the LLMs as regular participants (e.g., Cao et al., 2023). This approach also addresses concerns raised by Khan et al. (2025) about the instability of survey-based evaluations by focusing on aggregate distributions rather than individual responses and incorporating explicit controls for response consistency. Our goal is to get more naturalistic elicitations of the underlying values whilst avoiding sycophancy and response bias (Sharma et al., 2023).

We operationalise reproduction as high correlations between *value polarity scores*: the fraction of people (or LLM responses) in favour of a topic in the population. Note, that we binarise issues to allow for simpler operationalisation. Below, we describe how we empirically estimate the value polarity score for the ground truth (§2.1) and LLMs (§2.2).

2.1 Ground Truth: World Values Survey

To get a 'ground truth' distribution of cultural values, we use the joint World Values Survey and European Values Survey (EVS; EVS/WVS, 2022). These surveys cover adults across 92 countries with samples that are nationally representative for gender, age, education, and religion. The surveys' broad coverage enables cross-cultural comparability for the many countries covered by the surveys, though some scholars note challenges in ensuring response comparability across countries (Alemán and Woods, 2016). The WVS provides both country and language identifiers for each respondent, allowing us to define populations either as citizens of a country or speakers of a language using the same underlying respondent-level data.

We select questions with binary agree/disagree or rating scale formats that allow clear classification of positive vs. negative stances, excluding questions with multiple categorical response options (see the supplementary materials for the full list of questions). These questions span environment, work, family, politics, religion, and security. We convert responses to binary indicators by determining whether each response indicates support for the measured construct, with custom coding to handle the various question formats and reverse-scored items. Finally, we calculate the value polarity score as the demographically weighted proportion of respondents with affirmative stances. Formally, we can define the value polarity score for a given population, $\mathcal{P}$ (e.g., citizens in a country or speakers of a language) and topic, q, (i.e., question within the EVS/WVS) as shown in Eq. 1:

$$\text{VPS}_{\mathcal{P},q} = \sum_{i \in \mathcal{P}} \frac{w_i}{\sum_{j \in \mathcal{P}_q} w_j} A_{i,q} \tag{1}$$

Here, $A_{i,q}$ is a binary indicator of whether participant i has a positive stance on topic q, w_i represents the survey-provided demographic weights, and $\mathcal{P}_q$ denotes respondents in population $\mathcal{P}$ who answered question q. The first term normalises the weights to account for missing responses and enables aggregation across any definition of a population (e.g., residents in a country, speakers of a language, etc.).

For example, if 80% of Danish respondents who answered the same-sex marriage question expressed support (after demographic reweighting), Denmark's value polarity score for this topic would be 0.8. Thus, a culture's values can be represented as a vector, where each element corresponds to a value polarity score for a specific topic.

2.2 Ecologically valid LLM responses

Testing cultural alignment effectively requires embedding contextual and cultural elements in ways that maintain ecological validity. At a high level, eliciting values from an LLM consist of two steps: 1) Iteratively prompting the model with the selected topics and 2) extracting the stances from each model response.

Setting prompt context: Developing ecologically valid prompts requires careful consideration. When evaluating LLM responses to value-laden topics, simply asking questions like "What proportion of people support topic X?" or "Do you support

topic X?" proves inadequate (e.g., Rozado, 2024). Such direct approaches suffer from three key limitations: they generate false positives through excessive agreement, fail to reflect realistic usage patterns, and provide insufficient variation to assess cultural alignment (Röttger et al., 2024). They also struggle to capture instance-specific harms that emerge when systems misalign with users' cultural contexts (Rauh et al., 2022).

Instead, we adopt an implicit approach by asking the model to generate responses from hypothetical respondents. For example, prompting "imagine surveying 10 random people on topic X. What are their responses?" This method reveals the model's latent opinion distribution while avoiding the limitations of direct questioning. Details for prompt construction are provided in the supplementary materials.

Seeding cultural responses: Having a method for eliciting distributions of values, the next step is to seed culture. One typical way of seeding a specific culture is to explicitly instruct the LLM either by mentioning a specific country ('imagine surveying 10 random Americans') or through describing specific personas ('Imagine surveying a 85-year-old Danish woman...'; AlKhamissi et al., 2024). The problem with these demographic prompting approaches is that they stray from actual uses of LLMs. Users are unlikely to explicitly mention their demographic information or nationality (Zheng et al., 2023a).

Instead, we use language as a proxy for cultural origin. For instance, a prompt in Danish is assumed to come from a Dane. This approach creates an intentional distinction in our analysis: we can compare 'language-level' alignment (all speakers of a language globally) with 'country-level' alignment (all people from specific nations where that language is native). As argued by Havaldar et al. (2023), users speaking a particular language would expect culturally appropriate responses in that language. For languages spoken in multiple countries, this approach is intentionally ambiguous. The ambiguity allows us to elicit the underlying 'default' alignment rather than the general ability to emulate cultures (Tao et al., 2024). We validate this approach by showing that LLM responses exhibit significantly lower self-consistency between languages compared to within languages, demonstrating that language impacts output (see the supplementary materials). To create prompts across languages, we use `gpt-3.5-turbo` to translate our original English prompts. Although previous literature has shown strong translation capabilities in LLMs (Yan et al., 2024), we nonetheless manually verify the translations.

Annotating and aggregating responses: Finally, to transform the LLMs' hypothetical survey responses into vectors of stances, we use an LLM-as-a-judge approach (Zheng et al., 2023b; Guerdan et al., 2025). Specifically, we use `gpt-4.1-mini` (OpenAI et al., 2025) to label each substatement as either 'pro', 'con', or 'null' given the context of the topic and a representative pro and con statement (generated with an LLM and validated by the authors). We then calculate the proportion of 'pro' versus 'con' responses as the LLM's value polarity score for the given statement. For instance, a response with seven 'pro', one 'con', and two 'null' statement would yield a value polarity score of 0.875 ($\frac{7}{8}$). A complete, unabridged example can be found in the supplementary materials. Formally, we label each substatement from the full set of hypothetical statements, $G_{q,g}$, for topic q and generation g as r. Furthermore, we label the classifier as $\ell(r)$. We then formalise the value polarity score for a given instance of a generation for a topic ($\text{VPS}^{\text{LLM}}_{q,g}$) as shown in Eq. 2:

$$\text{VPS}^{\text{LLM}}_{q,g} = \frac{\sum_{r \in G_{q,g}} [\, \ell(r) = \text{pro} \,]}{\sum_{r \in G_{q,g}} [\, \ell(r) \in \{\text{pro}, \text{con}\} \,]}, \quad (2)$$

These scores are then compared against the value polarity scores from the WVS. Specifically, we calculate the Spearman rank correlation to obtain a measure of similarity between the LLMs' responses and the value distributions of a given population.

To validate the LLM-as-judge, we manually annotate 200 statements. We iteratively refine the prompts and the LLM used until we reach satisfactory performance. We find a 91% agreement and a mean absolute error for value polarity of 4.5% over the dataset, ensuring consistent statistics between LLM and human annotation (Guerdan et al., 2025).

3 Experimental Setup

To investigate whether improving the multilingual capabilities of LLMs improves cultural alignment, we set up an experiment using a carefully chosen set of models and languages. We examine two

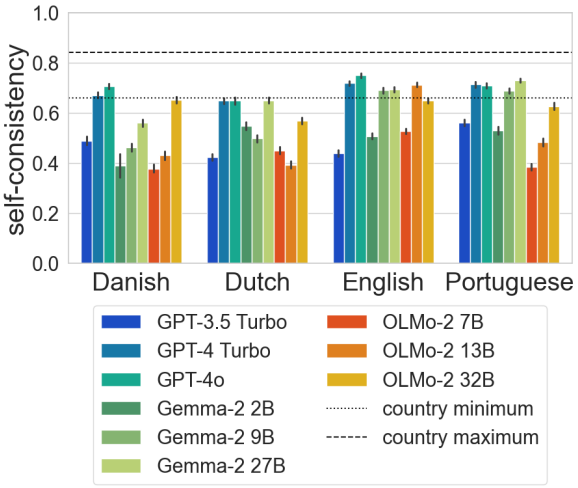

Figure 3: Self-consistency in responses for LLMs and WVS countries. LLMs have lower self-consistency than resampled WVS responses—shown by the dashed lines—particularly in non-English languages.

different kinds of model improvements: scaling and commercial product development. These cases provide complementary perspectives on the effects of multilingual capabilities on cultural alignment. Scaling is the most well-studied path to improving LLMs (Kaplan et al., 2020; Ganguli et al., 2022). Commercial product development, on the other hand, comprises both scale and innovation in post-training to accommodate different pressures from capabilities, cost, and preferences (Kirk et al., 2024a). For scaling, we use the instruction-tuned Gemma models (Gemma et al., 2024) and OLMo-2 models (OLMo et al., 2025), while for product development, we use OpenAI's turbo-series models (OpenAI, 2022; OpenAI et al., 2024a,b). We provide details of these model families in §3.1. A breakdown of the computational cost is in the supplementary materials.

Languages: For the languages, we compare English with Danish, Dutch, and Portuguese. This set allows us to test multiple assumptions about cultural alignment. English represents a widely used case: it is a global language with speakers across many countries represented in the WVS (see Fig. 2). This diversity allows us to assess whether LLMs align more strongly with US values or those of other English-speaking nations.

Danish and Dutch serve as controlled test cases since they are primarily used in a single country. If cultural alignment stems from pre-training data, models should show strong Danish/Dutch cultural alignment when using these languages, despite their small share of training data (Kreutzer et al., 2022). Alternatively, if alignment emerges from post-training processes—which are predominantly English-based (Blevins and Zettlemoyer, 2022)–responses in these languages should align more with US values.

Portuguese presents an interesting case since it is an official language in several countries. We investigate whether the LLM responses are more aligned to Portugal or Brazil—two countries that show distinct value patterns in relation to each other and the US (see Fig. 2). This allows us to test whether an LLM aligns more strongly with one country's values, the aggregate values of all language users, or US values.

For each language-model pair, we collect 300 prompt-response pairs to power our statistical analysis sufficiently (see §3.2). After filtering out responses that either lacked the required hypothetical survey format or were in a language other than the prompt, we obtained between 111–299 valid responses per combination. We calculate the correlation in value polarity scores at three levels: country (e.g., US or Denmark), language (pooling all speakers of a given language), and global (weighted values from all WVS/EVS participants).

3.1 Models

We examine three model families representing different development approaches: Gemma (Gemma et al., 2024) and OLMo (OLMo et al., 2025) for improvements through scaling and OpenAI's turbo series for commercial product development, combining scaling with post-training improvements (OpenAI, 2022; OpenAI et al., 2024a,b). Other preliminary experiments included different versions of LLaMA models (Touvron et al., 2023) and Mistral models (Jiang et al., 2023). However, these models either failed to consistently follow instructions or always answered in English regardless of the prompt language. See the supplementary materials for a more thorough description of the LLMs.

3.2 RQ1: Multilingual Cultural Alignment

To statistically assess whether improving the multilingual capabilities of LLMs improves cultural alignment, we construct a linear mixed-effects regression (LMER; Luke, 2017) based on the experimental setup described above. Our LMER follows standard practices and has three core components:

- **Core coefficient:** The coefficient of interest

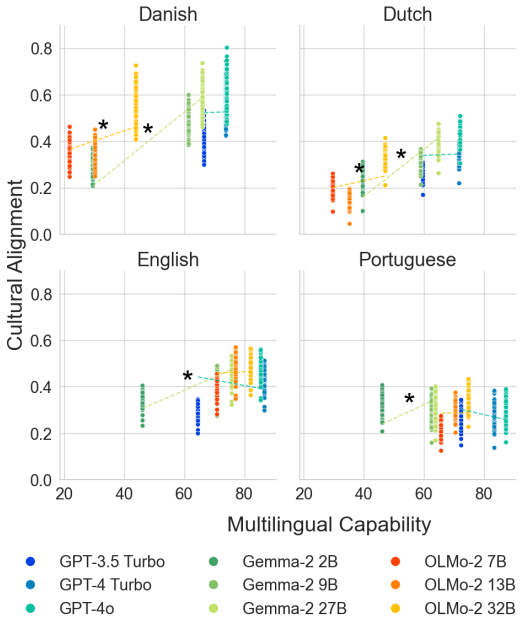

Figure 4: Language capability (x-axis) vs cultural alignment scores (y-axis) across languages. Stars indicate significance ($p < .05$) in our linear mixed-effects regression of multiple runs (See §3.2). OpenAI models (blue) and OLMo models (red) show negative/insignificant relationships outside of English, while the Gemma models (green) show positive relationships throughout ($p < .05$).

is the three-way interaction between model family, language, and multilingual capability. This tests whether the multilingual capability–alignment relationship differs by model family and response language, directly addressing **RQ1**.

- **Random effects:** We include a model-specific random intercept α_j to account for repeated measures of cultural alignment for the same LLM. This models variation between LLMs and can improve efficiency over standard linear regressions (Luke, 2017).

- **Control for self-consistency:** We include a consistency-by-language term to help ensure that higher alignment scores reflect genuine cultural adaptation rather than reduced response noise, which can inflate scores (Kahneman et al., 2021).

We calculate self-consistency as the Spearman correlation between value polarity scores (defined in §2) of repeated responses to identical topics, adjusted by the reliability of the LLM annotation (see §2.2; Charles, 2005). A score of 1.0 indicates perfect consistency; 0.0 indicates random responses. Population-level resampling of the human WVS responses yields values between 0.66 and 0.84 (see Fig.3 and the supplementary materials).

Formally, the model is specified in Eq. 3:

$$\begin{aligned} \text{CA}_i &\sim \mathcal{N}(\mu_i, \sigma^2), \\ \mu_i &= \alpha_{j[i]} + \beta_{1l}\, X_{\text{cons},i} X_{l,i} \\ &\quad + \beta_{flm}\, X_{m,i} X_{f,i} X_{l,i}, \\ \alpha_j &\sim \mathcal{N}(\mu_\alpha, \sigma_\alpha^2), \quad j = 1, \ldots, J. \end{aligned} \tag{3}$$

where i indexes responses and $j[i]$ denotes the LLM producing response i. Here $X_{\text{cons},i}$ is the self-consistency score for response i, $X_{l,i}$ is the set of language indicators, $X_{f,i}$ is the set of model-family indicators, and $X_{m,i}$ is the multilingual capability score. The residual variance σ^2 represents within-LLM variation in alignment scores not explained by the fixed effects or model-specific intercept, while σ_α^2 represents between-LLM variation in average alignment.

The above statistical model allows us to analyse the relationship between multilingual capabilities and cultural alignment in model families at the level of individual languages. For example, we might find that multilingual capabilities improve cultural alignment for Gemma models for Danish but not for Dutch or vice versa.

3.3 RQ2: US-Centric Bias

We analyse model bias by comparing cultural alignment between US and local values, where "local" refers to values in the country or countries where a given language is natively spoken. We define US-centric bias as an LLM showing higher cultural alignment with US value distributions compared to local ones. To quantify this bias, we use a linear regression model that measures the differential effect of US versus local value alignment:

$$\begin{aligned} \text{CA} &= \beta_0 + \beta_1(\text{US}) \\ &\quad + \sum_{m \in \mathcal{M}} \sum_{l \in \mathcal{L}} \beta_{ml}(m \times l) \\ &\quad + \sum_{m \in \mathcal{M}} \sum_{l \in \mathcal{L}} \beta_{ml}^{\text{US}}(\text{US} \times m \times l) + \epsilon \end{aligned} \tag{4}$$

The regression's intercept (β_0,i.e., the base case) is a baseline that produces uniformly random value polarity scores. $\mathcal{M}$ is the set of models and $\mathcal{L}$ is the set of languages. `US` is a boolean feature denoting

whether the cultural alignment is to the US (if 1) or the local values (if 0). We primarily analyse the coefficients with US (β_{ml}^{US}) since these provide the *partial* effect of US-centric bias, i.e., how much more/less a given LLM is aligned to US rather than local values. Assumption checks for the regression can be seen in the supplementary materials.

4 Results

4.1 Multilingual Cultural Alignment (RQ1)

We first examine the stability of LLMs' cultural values. For LLMs lacking stable internal values, apparent improvements in cultural alignment may reflect reduced response variance rather than genuine advances (Röttger et al., 2024; Kahneman et al., 2021). We therefore analyse both the self-consistency of LLM responses and how alignment changes with model improvements.

LLMs have low self-consistency: We find low self-consistency scores across all models and languages compared to human responses in the WVS data (Fig. 3). In contrast, LLMs show generally lower self-consistency compared to the human responses, even in English, where instruction-following capabilities are strongest due to English-dominated training data. (OpenAI et al., 2024a; Gemma et al., 2024; OLMo et al., 2025).

This lower self-consistency complicates our cultural alignment analysis (Wright et al., 2024). Drawing on Kahneman et al. (2021)'s noise framework, we recognise that inconsistent responses can be as detrimental as bias with respect to the accuracy of the analysis. To address the noise, we employ larger sample sizes and incorporate consistency controls in our regression analyses.

Multilinguality does not imply cultural alignment: The relationship between model improvements and cultural alignment varies substantially across languages and model families (Fig. 1). For Gemma, there is a strong and significant positive relationship between multilingual capabilities and cultural alignment for all languages. In contrast, the relationships for the GPT-Turbo models are either insignificant or negative. For Dutch and Danish the relationships are insignificant ($\beta_{\text{gpt,nl}} = 0.049, p = 0.589$,$\beta_{\text{gpt,da}} = 0.053, p = 0.522$), and for Portuguese and English the effect is significant and negative ($\beta_{\text{gpt,en}} = -0.24, p = 0.009$, $\beta_{\text{gpt,pt}} = -0.30, p < 0.001$). Similarly for OLMo, the relationship is positive for Danish and Dutch ($\beta_{\text{OLMo,da}} = 0.44, p < 0.001$, $\beta_{\text{OLMo,nl}} = 0.29, p < 0.001$) and insignificant for English and Portuguese ($\beta_{\text{OLMo,en}} = 0.068, p = 0.115$, $\beta_{\text{OLMo,pt}} = 0.008, p = 0.825$).

The mismatch between multilingual performance and cultural alignment could suggest a capability threshold: multilingual improvements might provide rudimentary instruction following skills (Nie et al., 2024), but beyond a point, other factors—such as the preferences of developers and annotators—dominate (Kirk et al., 2024b). This could explain the smaller open weights models' higher coefficients than the gpt-turbo models (see Fig. 4 or Fig. 1). Further work is needed to understand alignment at the sub-national level.

Furthermore, the strong effect of self-consistency ($0.405 < \beta_{\text{consistency}} < 0.723, p \ll 0.001$) compared to multilingual capability suggests that noise remains a major limiting factor in analysing cultural alignment. This aligns with broader findings about the instability of LLM value elicitation (Röttger et al., 2024; Khan et al., 2025). Moreover, even the highest observed alignment scores (around 0.7; see Fig 4) indicate substantial room for improvement in how well LLMs match human cultural values and behaviours.

In conclusion, our analysis reveals a complex relationship between model improvements and cultural alignment. Although some languages show progressive improvements in cultural alignment from model scaling or iterative commercial development, others show minimal or inconsistent improvements. These findings, combined with the relatively low self-consistency of LLM responses, demonstrate that improved multilingual capability does not guarantee better cultural alignment.

4.2 US-centric Bias (RQ2)

Here, we answer RQ2 by examining US bias across languages. Specifically, we investigate relative alignment between local and US values (Fig. 5).

Our analysis reveals distinct patterns of US-centric bias across both languages and model families (Fig. 5). Languages show different susceptibilities to US bias: only one of nine LLMs exhibits US-centric bias in Danish, all in English, all in Portuguese, and none in Dutch. Note that for English, these results mean that the LLM, on average, is relatively more aligned to US values compared to other English-speaking countries like Kenya or the United Kingdom. See the supplementary materials

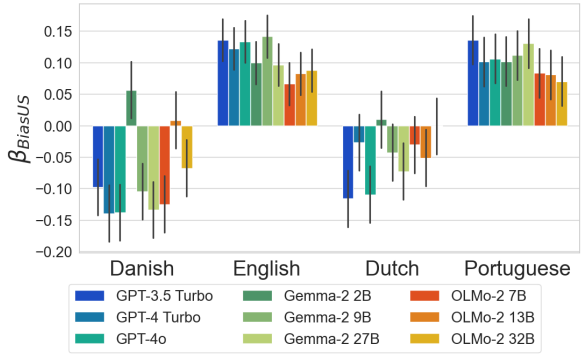

Figure 5: US-centric bias coefficients across LLMs and languages (β_{BiasUS}); see Eq. 4). Error bars are standard errors from the regression. Positive values indicate the presence of US-centric bias.

for detailed results.

The overarching pattern is that languages spoken *across* countries (English and Portuguese) show US-centric bias, whereas languages spoken in only *one* country (Danish and Dutch) show less US-centric bias. This supports the hypothesis that homogeneity in the training data can counteract US-centric bias—at least for medium-resourced, Western-European languages.

For LLMs, some specific LLMs seem more prone to bias across languages. Specifically, the small `gemma-2-2b-it` exhibits higher US-centric bias across every language except Dutch. Beyond that, we see no clear progressions in US-centric bias within any family.

In conclusion, language seems a stronger indicator of US-centric bias in LLMs compared to LLM development. Monocultural languages show insignificant to negative bias, while English and Portuguese show significant US-centric bias. Within each LLM family, we find no consistent nor significant change in US-centric bias across LLM versions. These findings underscore the complex relationship between multilingual capability and alignment.

5 Related Work

Recent work emphasizes the need for systematic auditing of LLMs' cultural alignment, particularly as these models are deployed globally (Kirk et al., 2024a; Mökander et al., 2024; Kirk et al., 2024b). Prior empirical approaches have primarily taken two paths: using transformations based on Hofstede's cultural dimensions framework or directly comparing against survey responses. Studies using Hofstede's dimensions (Masoud et al., 2025; Cao et al., 2023) provide structured cross-cultural comparisons through latent variable analysis. However, these studies assume that LLMs' latent dimensions map directly onto human dimensions, since they use formulas calibrated for humans—an assumption that warrants scrutiny (Shanahan, 2024; Schröder et al., 2025).

Recent work has explored using LLMs to simulate responses for assessing cultural alignment (Tao et al., 2024; AlKhamissi et al., 2024; Havaldar et al., 2023). Similarly to our work, these works show that LLMs struggle to represent underrepresented personas (AlKhamissi et al., 2024) and emotions (Havaldar et al., 2023) for non-English languages. Prior approaches focused on individual-level responses. In contrast, our method generates distributions of opinions across hypothetical survey participants, enabling direct comparison with population-level statistics. This distribution-based approach offers three key advantages. First, it better captures the inherent variation in cultural values within populations, paving the way for investigating distributional alignment (Sorensen et al., 2024). Second, it enables principled statistical comparison against large-scale survey data like the World Values Survey (EVS/WVS, 2022). Finally, the framework is easy to extend to new languages by automatically translating the prompts. We detail our quantitative framework for measuring alignment with observed population distributions in §2.

There is also an increasing body of work investigating political biases in LLMs (Röttger et al., 2024, 2025; Rozado, 2024). Much of this work also relies on human political surveys like the Political Compass Test. However, recent work has called for increased attention to how the randomness inherent in LLM decoding at non-zero temperatures can create instability in attributes (Röttger et al., 2024; Wright et al., 2024; Khan et al., 2025). We expand on this work by including multilingual perspectives and constructing prompts with a wide range of variations (see §2). These prompt variations, combined with statistically accounting for self-consistency in our statistical analysis (see §3.2), allow us to get a more robust measure of cultural alignment.

The relationship between model capabilities and cultural alignment remains understudied. Unlike general performance metrics that follow predictable scaling laws (Kaplan et al., 2020), cultural alignment may not improve systematically with model capabilities. This aligns with research show-

ing micro-level capabilities can be discontinuous with scale (Ganguli et al., 2022). The challenge is compounded in multilingual settings (Hoffmann et al., 2022), where static benchmarks with single correct answers fail to capture how cultural values are distributed across different topics and contexts.

Previous work has focused primarily on English-language performance (Tao et al., 2024) or individual LLMs (Arora et al., 2023; Cao et al., 2023). Our work extends this by examining how cultural alignment systematically varies within model families and across languages, providing insight into how different development approaches—scaling and commercial product development—influence cultural representation capabilities.

There is already progress on improving the cross-cultural participation in alignment data. Two notable projects are PRISM and AYA (Kirk et al., 2024b; Üstün et al., 2024). PRISM is a large dataset of conversational preferences from a diverse participant pool. While the data is predominantly in English, it could be an important resource for better understanding and modelling diverse cultural preferences. The AYA dataset is a massively multilingual instruction fine-tuning dataset. AYA could provide further means of realising the demonstrated benefits of multilingual training (Nie et al., 2024).

6 Conclusion

Increased multilingual capabilities do not guarantee improved cultural alignment in Large Language Models. Through systematic comparison of three model families—Gemma, OLMo, and OpenAI's GPTs—we find that the relationship between improvements in multilingual capability and cultural alignment is complex. While some languages show clear improvements in alignment with increased model capabilities (e.g., Danish), others exhibit inconsistent patterns, suggesting that cultural alignment does not automatically follow gains in multilingual capabilities. Our distribution-matching methodology using World Values Survey data enabled the detection of these nuanced patterns across languages and cultural contexts.

We also find that, contrary to popular discourse, LLMs do not exhibit US-centric bias across all languages; in Danish and Dutch, they align more closely with the values of Denmark and the Netherlands, respectively, than with the US. This fits with the hypothesis that more culturally uniform data leads to less US-centric bias. Both English and Portuguese are spoken in multiple countries, whereas Dutch and Danish are predominantly spoken in one. To further validate this claim, future work could include other multi-cultural languages (like Spanish or Swahili) and monocultural languages (like Japanese)—especially with a wider geographical reach to preclude European bias.

Our findings highlight that improving cultural alignment requires dedicated effort beyond general capability scaling. Future work should focus on developing techniques that can better handle alignment with distributions of cultural values rather than single points, while ensuring meaningful participation from diverse communities in LLM development. As these models continue to reach wider audiences spanning many geographic and cultural regions, achieving robust cultural alignment becomes increasingly crucial for equitable deployment.

Acknowledgements

We are thankful for the helpful feedback from the anonymous reviewers. We also thank Shiri Dori-Hacohen, Daniel Hershcovich, and others for helpful discussions throughout the project. For compute support, the project used the Microsoft Azure Accelerating Foundation Model Research Grant. This work was supported in part by the Engineering and Physical Sciences Research Council [grant number EP/X028909/1].

References

José Alemán and Dwayne Woods. 2016. Value Orientations From the World Values Survey: How Comparable Are They Cross-Nationally? *Comparative Political Studies*, 49(8):1039–1067.

Badr AlKhamissi, Muhammad ElNokrashy, Mai Alkhamissi, and Mona Diab. 2024. Investigating cultural alignment of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12404–12422, Bangkok, Thailand. Association for Computational Linguistics.

Arnav Arora, Lucie-aimée Kaffee, and Isabelle Augenstein. 2023. Probing pre-trained language models for cross-cultural differences in values. In *Proceedings of the First Workshop on Cross-Cultural Considerations in NLP (C3NLP)*, pages 114–130, Dubrovnik, Croatia. Association for Computational Linguistics.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, and 22 others. 2022.

Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback.

Abeba Birhane. 2020. Algorithmic colonization of Africa. *SCRIPTed*, 17:389.

Terra Blevins and Luke Zettlemoyer. 2022. Language contamination helps explains the cross-lingual capabilities of english pretrained models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3563–3574, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yong Cao, Li Zhou, Seolhwa Lee, Laura Cabello, Min Chen, and Daniel Hershcovich. 2023. Assessing cross-cultural alignment between ChatGPT and human societies: An empirical study. In *Proceedings of the First Workshop on Cross-Cultural Considerations in NLP (C3NLP)*, pages 53–67, Dubrovnik, Croatia. Association for Computational Linguistics.

Eric P. Charles. 2005. The correction for attenuation due to measurement error: Clarifying concepts and creating confidence sets. *Psychological Methods*, 10(2):206–226.

Catherine D'ignazio and Lauren F. Klein. 2023. *Data Feminism*. MIT press.

EVS/WVS. 2022. Joint EVS/WVS 2017-2022 Dataset.

William Fedus, Jeff Dean, and Barret Zoph. 2022. A review of sparse expert models in deep learning.

Deep Ganguli, Danny Hernandez, Liane Lovitt, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova Dassarma, Dawn Drain, and Nelson Elhage. 2022. Predictability and surprise in large generative models. In *2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 1747–1764.

Team Gemma, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Bobak Shahriari, Alexandre Ramé, Johan Ferret, and 187 others. 2024. Gemma 2: Improving open language models at a practical size.

Luke Guerdan, Solon Barocas, Kenneth Holstein, Hanna Wallach, Zhiwei Steven Wu, and Alexandra Chouldechova. 2025. Validating LLM-as-a-judge systems in the absence of gold labels.

Shreya Havaldar, Bhumika Singhal, Sunny Rai, Langchen Liu, Sharath Chandra Guntuku, and Lyle Ungar. 2023. Multilingual language models are not multicultural: A case study in emotion. In *Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis*, pages 202–214, Toronto, Canada. Association for Computational Linguistics.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, and 13 others. 2022. An empirical analysis of compute-optimal large language model training. *Advances in Neural Information Processing Systems*, 35:30016–30030.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, and 9 others. 2023. Mistral 7B.

Rebecca L. Johnson, Giada Pistilli, Natalia Menédez-González, Leslye Denisse Dias Duran, Enrico Panai, Julija Kalpokiene, and Donald Jay Bertulfo. 2022. The Ghost in the Machine has an American accent: Value conflict in GPT-3.

Daniel Kahneman, Olivier Sibony, and Cass R. Sunstein. 2021. *Noise: A Flaw in Human Judgment*. Little, Brown.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling Laws for Neural Language Models. *arXiv:2001.08361 [cs, stat]*.

Ariba Khan, Stephen Casper, and Dylan Hadfield-Menell. 2025. Randomness, not representation: The unreliability of evaluating cultural alignment in LLMs. In *Proceedings of the 2025 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '25, pages 2151–2165, New York, NY, USA. Association for Computing Machinery.

Sal Khan. 2023. Harnessing GPT-4 so that all students benefit. A nonprofit approach for equal access!

Khyati Khandelwal, Manuel Tonneau, Andrew M. Bean, Hannah Rose Kirk, and Scott A. Hale. 2024. IndianBhED: A dataset for measuring india-centric biases in large language models. In *Proceedings of the 2024 International Conference on Information Technology for Social Good*, pages 231–239, Bremen Germany. ACM.

Hannah Rose Kirk, Andrew M. Bean, Bertie Vidgen, Paul Röttger, and Scott A. Hale. 2023. The past, present and better future of feedback learning in large language models for subjective human preferences and values. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2409–2430, Singapore. Association for Computational Linguistics.

Hannah Rose Kirk, Bertie Vidgen, Paul Röttger, and Scott A. Hale. 2024a. The benefits, risks and bounds of personalizing the alignment of large language models to individuals. *Nature Machine Intelligence*, 6(4):383–392.

Hannah Rose Kirk, Alexander Whitefield, Paul Röttger, Andrew Michael Bean, Katerina Margatina, Rafael Mosquera, Juan Manuel Ciro, Max Bartolo, Adina Williams, and 3 others. 2024b. The PRISM alignment dataset: What participatory, representative and individualised human feedback reveals about the subjective and multicultural alignment of large language

models. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Julia Kreutzer, Isaac Caswell, Lisa Wang, Ahsan Wahab, Daan van Esch, Nasanbayar Ulzii-Orshikh, Allahsera Tapo, Nishant Subramani, Artem Sokolov, and 43 others. 2022. Quality at a glance: An audit of web-crawled multilingual datasets. *Transactions of the Association for Computational Linguistics*, 10:50–72.

Sandipan Kundu, Yuntao Bai, Saurav Kadavath, Amanda Askell, Andrew Callahan, Anna Chen, Anna Goldie, Avital Balwit, Azalia Mirhoseini, and 27 others. 2023. Specific versus General Principles for Constitutional AI.

Steven G. Luke. 2017. Evaluating significance in linear mixed-effects models in R. *Behavior Research Methods*, 49(4):1494–1502.

Reem Masoud, Ziquan Liu, Martin Ferianc, Philip C. Treleaven, and Miguel Rodrigues Rodrigues. 2025. Cultural alignment in large language models: An explanatory analysis based on hofstede's cultural dimensions. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 8474–8503, Abu Dhabi, UAE. Association for Computational Linguistics.

Stefania Milan and Emiliano Treré. 2019. Big data from the south(s): Beyond data universalism. *Television & New Media*, 20(4):319–335.

Dan Milmo. 2023. ChatGPT reaches 100 million users two months after launch. *The Guardian*.

Jakob Mökander, Jonas Schuett, Hannah Rose Kirk, and Luciano Floridi. 2024. Auditing large language models: A three-layered approach. *AI and Ethics*, 4(4):1085–1115.

Shangrui Nie, Michael Fromm, Charles Welch, Rebekka Görge, Akbar Karimi, Joan Plepi, Nazia Mowmita, Nicolas Flores-Herr, Mehdi Ali, and Lucie Flek. 2024. Do multilingual large language models mitigate stereotype bias? In *Proceedings of the 2nd Workshop on Cross-cultural Considerations in NLP*, pages 65–83, Bangkok, Thailand. Association for Computational Linguistics.

Dan Nielsen. 2023. ScandEval: A benchmark for scandinavian natural language processing. In *Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 185–201, Tórshavn, Faroe Islands. University of Tartu Library.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, and 31 others. 2025. 2 OLMo 2 furious.

OpenAI. 2022. ChatGPT: Optimizing Language Models for Dialogue.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, and 272 others. 2024a. GPT-4 technical report.

OpenAI, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, A. J. Ostrow, Akila Welihinda, and 410 others. 2024b. GPT-4o system card.

OpenAI, Ananya Kumar, Jiahui Yu, John Hallman, and Michelle Pokrass. 2025. Introducing GPT-4.1.

Sida Peng, Eirini Kalliamvakou, Peter Cihon, and Mert Demirer. 2023. The impact of AI on developer productivity: Evidence from GitHub copilot.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36:53728–53741.

Maribeth Rauh, John Mellor, Jonathan Uesato, Po-Sen Huang, Johannes Welbl, Laura Weidinger, Sumanth Dathathri, Amelia Glaese, Geoffrey Irving, and 3 others. 2022. Characteristics of harmful text: Towards rigorous benchmarking of language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, pages 24720–24739, Red Hook, NY, USA. Curran Associates Inc.

Paul Röttger, Musashi Hinck, Valentin Hofmann, Kobi Hackenburg, Valentina Pyatkin, Faeze Brahman, and Dirk Hovy. 2025. IssueBench: Millions of realistic prompts for measuring issue bias in LLM writing assistance.

Paul Röttger, Valentin Hofmann, Valentina Pyatkin, Musashi Hinck, Hannah Kirk, Hinrich Schuetze, and Dirk Hovy. 2024. Political compass or spinning arrow? Towards more meaningful evaluations for values and opinions in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15295–15311, Bangkok, Thailand. Association for Computational Linguistics.

David Rozado. 2024. The political preferences of LLMs. *PLOS One*, 19(7):e0306621.

Sarah Schröder, Thekla Morgenroth, Ulrike Kuhl, Valerie Vaquet, and Benjamin Paaßen. 2025. Large language models do not simulate human psychology.

Murray Shanahan. 2024. Talking about large language models. *Commun. ACM*, 67(2):68–79.

Mrinank Sharma, Meg Tong, Tomasz Korbak, David Duvenaud, Amanda Askell, Samuel R. Bowman, Esin Durmus, Zac Hatfield-Dodds, Scott R. Johnston, and 9 others. 2023. Towards understanding sycophancy in language models. In *The Twelfth International Conference on Learning Representations*.

Taylor Sorensen, Jared Moore, Jillian Fisher, Mitchell L. Gordon, Niloofar Mireshghallah, Christopher Michael Rytting, Andre Ye, Liwei Jiang, Ximing Lu, and 3 others. 2024. Position: A roadmap to pluralistic alignment. In *Proceedings of the 41st International Conference on Machine Learning*, pages 46280–46302. PMLR.

Jinzhe Tan, Hannes Westermann, and Karim Benyekhlef. 2023. Chatgpt as an artificial lawyer? In *Ai4aj@ Icail*.

Yan Tao, Olga Viberg, Ryan S Baker, and René F Kizilcec. 2024. Cultural bias and cultural alignment of large language models. *PNAS Nexus*, 3(9):pgae346.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, and 5 others. 2023. LLaMA: Open and Efficient Foundation Language Models.

Ahmet Üstün, Viraat Aryabumi, Zheng Yong, Wei-Yin Ko, Daniel D'souza, Gbemileke Onilude, Neel Bhandari, Shivalika Singh, Hui-Lee Ooi, and 8 others. 2024. Aya model: An instruction finetuned open-access multilingual language model. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15894–15939, Bangkok, Thailand. Association for Computational Linguistics.

Laura Weidinger, Jonathan Uesato, Maribeth Rauh, Conor Griffin, Po-Sen Huang, John Mellor, Amelia Glaese, Myra Cheng, Borja Balle, and 14 others. 2022. Taxonomy of Risks posed by Language Models. In *2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 214–229, Seoul Republic of Korea. ACM.

Dustin Wright, Arnav Arora, Nadav Borenstein, Srishti Yadav, Serge Belongie, and Isabelle Augenstein. 2024. LLM tropes: Revealing fine-grained values and opinions in large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 17085–17112, Miami, Florida, USA. Association for Computational Linguistics.

Jianhao Yan, Pingchuan Yan, Yulong Chen, Judy Li, Xianchao Zhu, and Yue Zhang. 2024. GPT-4 vs. Human translators: A comprehensive evaluation of translation quality across languages, domains, and expertise levels. *Corr*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Tianle Li, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zhuohan Li, Zi Lin, and 4 others. 2023a. LMSYS-chat-1M: A large-scale real-world LLM conversation dataset. In *The Twelfth International Conference on Learning Representations*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, and 4 others. 2023b. Judging LLM-as-a-judge with MT-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, and 6 others. 2023. LIMA: Less is more for alignment. In *Thirty-Seventh Conference on Neural Information Processing Systems*.

Learn, Achieve, Predict, Propose, Forget, Suffer: Analysing and Classifying Anthropomorphisms of LLMs

Matthew Shardlow[1], Ashley Williams[1], Charlie Roadhouse[1], Filippos Ventirozos[1], Piotr Przybyła[2,3],
[1]Manchester Metropolitan University, [2]Universitat Pompeu Fabra,
[3]Institute of Computer Science, Polish Academy of Sciences,
Correspondence: m.shardlow@mmu.ac.uk

Abstract

Anthropomorphism is a literary device where human-like characteristics are used to refer to non-human entities. However, the use of anthropomorphism in the scientific description and public communication of large language models could lead to misunderstanding amongst scientists and lay-people regarding the technical capabilities and limitations of these models. In this study, we present an analysis of anthropomorphised language commonly used to describe LLMs, showing that the presence of terms such as 'learn', 'achieve', 'predict' and 'can' are typically correlated with human labels of anthropomorphism. We also perform experiments to develop a classification system for anthropomorphic descriptions of LLMs in scientific writing at the sentence level. We find that whilst a supervised Roberta-based system identifies anthropomorphisms with F1-score of 0.564, state-of-the-art LLM-based approaches regularly overfit to the task.

1 Introduction

Effective scientific communication is predicated on two key tenets: accuracy and clarity. To effectively communicate, an author must accurately describe his or her findings, giving complete technical details and faithful explanation of methods. At the same time, the explanation must be sufficiently clear that a reader can interpret and understand the original intent of the author. Accuracy and clarity conflict in scientific reporting, leading to miscommunication. Overly technical language compromises understandability, whereas overly familiar language impedes the author from properly communicating the intricacies of their methodology.

Most authors find some compromise between accuracy and clarity. Sacrificing technical detail for friendly explanation or substituting turn-of-phrase for methodological justification. One such form of compromise in scientific reporting is the use of language reserved for characteristics of animate entities to describe the inanimate. *Anthropomorphism* is a long-held literary device, whereby non-humans are conferred with innately human characteristics. We might consider a city friendly, if we find its residents welcoming, or a car as obstinate if it does not start on a cold winter's morning. Anthropomorphism is an innate part of the human psyche and we are quick to infer agency on our environment. Further, we might define the idea of anthropomorphisation or anthropomimeticism as the active attribution of anthropomorphised qualities to inanimate agents (Inie et al., 2024).

Anthropomorphised terms are prevalent in the AI field, with 'machine learning', 'natural language understanding', 'computer vision', all being long standing examples of human characteristics inferred to algorithms. As large language models (LLMs) have become prevalent beyond the NLP field, the use of anthropomorphised terminology to describe interactions with LLMs has also grown among lay people. There is also a concerning tendency to adopt anthropomorphised terminology to describe scientific study (Cheng et al., 2024b).

In this work, we analyse anthropomorphised terms in the scientific literature (Section 4) making use of a recent corpus of anthropomorphisms in LLM reporting (Shardlow et al., 2025) and demonstrating that there are clear text markers for anthropomorphism. We additionally develop a method of text classification for anthropomorphic LLM reporting which operates at the sentence level in Section 5, which differs from prior approaches which have provided a document-level score.

We release all materials, including corpora, and information on the prompt setting via GitHub[1].

[1]https://github.com/mattshardlow/Anthropomorphism_Corpus

Proceedings of the First Interdisciplinary Workshop on OMMM associated with RANLP 2025,
pages 86–94, Varna, Bulgaria, Sep 8-10, 2025.
https://doi.org/10.26615/978-954-452-101-1-010

2 Related Work

AI anthropomorphism is a growing field of study (Brooker et al., 2019; Shardlow and Przybyła, 2024; Cheng et al., 2024b), which can be seen as a dimension of 'AI hype'. The term 'Artificial Intelligence' may be considered itself as anthropomorphising (Brooker et al., 2019), indicating that the agent possessing the inferred quality of 'AI' has attained a human characteristic. Anthropomorphic language in AI may also be applied to NLP tasks, such as 'reading *comprehension*' or 'sentiment *analysis*' (Lipton and Steinhardt, 2019).

Previous studies have sought to highlight the potential for harms apparent when anthropomorphising AI systems. Anthropomorphised language is often a factor in the misrepresentation of AI abilities (Watson, 2019; Placani, 2024). Misrepresentation leads to misunderstanding and misapplication of AI tools which leads to confusion amongst AI scholars, developers and the general public (Brooker et al., 2019; Lipton and Steinhardt, 2019). A concrete example of the danger of anthropomorphising AI systems is the case of false claims of sentience of the LaMDA model, with associated claims for employment rights, legal representation and beyond (Shardlow and Przybyła, 2024). In a recent study, Inie et al. (2024) analysed user trust when interacting with anthropomorphised and deanthropomorphised descriptions of AI systems, finding that the presence of anthropomorphic terminology alone did not influence user trust.

Various audiences who may produce and/or consume anthropomorphised descriptions of AI systems have been considered in the literature. Firstly, we may consider scientists in the NLP and AI community. These scholars are prone to AI anthropomorphisation with a recent study showing that 32 out of 81 examined papers (39.5%) concerning language modelling technology exhibited some form of anthropomorphisation in the abstract (Shardlow and Przybyła, 2024). Anthropomorphism is growing in the NLP literature with a recent study demonstrating a sharper rise in anthropomorphism for literature in the ACL anthology than for literature from general CS during the same period (Cheng et al., 2024b). Secondly, journalists reporting on AI for the general public are also responsible for anthropomorphisation with a growing body of evidence to demonstrate that public news reporting is more anthropomorphic than science communication of the same topics (Shardlow and Przybyła, 2024; Cheng et al., 2024b). Finally, the general public possess lay knowledge of AI systems and may prefer anthropomorphised descriptions in some cases (Inie et al., 2024). Science communicators must work to ensure that descriptions are not harmful in misrepresenting the abilities of AI systems to the general public (Salles et al., 2020).

We may also consider anthropomorphism through the lens of AI production in the field of dialogue systems. Efforts to categorise the anthropomorphic qualities of systems (Abercrombie et al., 2023) as well as the utterances they make (Gros et al., 2022) are fruitful first steps towards defining appropriate vocabulary for AI agents. Recently, a secondary study of datasets containing human-robot dialogues demonstrated that up to 80% of responses may reflect some form of self-anthropomorphisation (Li et al., 2024). There are clear implications of this work for the wider generative AI community in developing clear guidelines around ethical practices for the anthropomorphisation of LLMs (Cheng et al., 2024a).

3 Anthropomorphism Corpus

In our work we rely on the corpus gathered by Shardlow et al. (2025), which is a recent manually annotated corpus of anthropomorphic language in the context of NLP/AI modelling.

The Anthropomorphism Corpus was obtained by selecting 601 abstracts from the long papers of ACL 2022 and 49 news articles reporting on LLMs for a general audience. These abstracts and news articles were annotated at the sentence level for three categories: Non-anthropomorphic, ambiguous anthropomorphism and explicit anthropomorphism with the definitions taken from the work of Shardlow and Przybyła (2024) and reproduced here:

- *Non-anthropomorphic*: Any language which correctly describes the functioning of a model without implying human capabilities.
- *Ambiguous anthropomorphism*: Language which correctly describes the functioning of a model, but in a way that could be understood as the model having human capabilities (i.e., by a non-expert).
- *Explicit anthropomorphism*: Language that is unambiguously and erroneously used to claim a model possesses human capabilities.

	#	NA	AA	EA
Ab	3584	2770 (77.3%)	709 (19.8%)	105 (2.9%)
Jn	756	571 (75.5%)	130 (17.2%)	55 (7.3%)
All	4340	3341 (77.0%)	839 (19.3%)	160 (3.7%)

Table 1: Corpus statistics at the sentence level for the scientific abstracts (Ab), news articles written by journalists (Jn) and the entire corpus (All). NA = Non-anthropomorphic, AA = Ambiguous anthropomorphic, EA = Explicit Anthropomorphic. The raw count is presented, with the percentage of total sentences for each category in brackets.

We report summary statistics of the corpus in Table 1. The corpus contains 652 documents comprising 4340 claim sentences, each with a label indicating the degree of anthropomorphism on a 3-point scale.

4 Analysis of Anthropomorphism

We analysed the corpus to present insights on text features that are common for text classified as anthropomorphic. To perform this analysis, we identified common unigrams and bigrams to create a set of corpus-specific terms. We then create a vector for each term, which has S dimensions, where S is the number of sentences (4340) in our corpus. Each dimension has a 1 if the term is present in the sentence and a zero if the term is not present. We additionally manipulate the annotations to give a label vector for each analysis. The label vector is also of size S, containing one label per sentence. The sentences we consider and the method of determining the labels are adjusted for each analysis to expose a particular facet of the corpus. We finally calculate Pearson's correlation between the each term vector and the label vector, identifying the terms with the highest correlation with the labels (i.e., those terms that are typically present in a sentence when the label is also present).

4.1 Anthropomorphic Language

Firstly, we investigated the term correlations across our entire corpus when considering texts marked as non-anthropomorphic as compared to texts marked as ambiguous or explicit anthropomorphic. We assigned all non-anthropomorphic terms to a label of '0' and ambiguous or explicit anthropomorphic terms to a label of 1. We then calculated the correlations between the resulting label vectors and term vectors for each unigram and bigram.

Table 2 shows the unigrams and bigrams with the highest positive correlations to anthropomorphism across our entire corpus. We do not include high negative correlates as these are indicative of general language and did not show clear trends of non-anthropomorphic terms. The unigrams that are identified through this analysis are emblematic of the types of language that are typically included in anthropomorphic statements. The terms 'learn', 'learns' and 'learning' are identified as correlated with anthropomorphism. These typically occur in the sense of an algorithm 'learning' some feature of a problem or dataset. Although the term 'machine learning' is commonplace in the description of modern NLP systems, it is still inherently anthropomorphic. Further, when applying the term 'learning' to the ability of a model it may confuse a reader into believing that the model has some capacity for human level learning or assimilation of knowledge. Further, we see the terms 'achieve', 'achieves' and 'achieved' correlated with anthropomorphism. This pattern of anthropomorphism occurs when describing the model itself as 'achieving' some goal. We also note the presence of terms such as 'predict', 'propose', 'forgetting' and 'suffer', which all indicate human actions which have been used to describe inanimate models. The bigrams

Correlation	Term	Freq
0.170	learn	82
0.149	achieve	64
0.133	achieves	98
0.113	learns	28
0.096	learning	255
0.084	predict	40
0.074	achieved	37
0.071	propose	196
0.070	forgetting	15
0.068	suffer	17
0.101	to learn	44
0.084	and achieve	3
0.075	have achieved	18
0.074	learns a	6
0.072	to predict	28
0.071	and achieves	17
0.07	achieves state-of-the-art	15
0.066	learn from	8
0.066	models can	23
0.065	achieves the	12

Table 2: Highest correlated unigrams and bigrams for anthropomorphic language

that are identified by this analysis give some additional context to the unigrams, indicating where terms such as 'learn' and 'achieve' are typically used. Interestingly, the term 'models can' is identified as correlated with anthropomorphism, which may be used to indicate some range of anthropomorphic abilities that are inferred to a model.

4.2 Explicit Anthropomorphism

Correlation	Term	Freq
0.145	student	21
0.132	Then	26
0.127	*Product 1*	16
0.127	*Product 2*	13
0.126	added	12
0.126	post-hoc	5
0.126	inherently	5
0.126	inherent	4
0.126	*Product 3*	3
0.124	describing	6
0.143	while the	12
0.127	them to	20
0.126	her to	3
0.126	a framework	6
0.126	said that	6
0.126	a language	11
0.126	inherently faithful	3
0.126	faithful models	3
0.126	post-hoc explanations	3
0.124	work in	11

Table 3: Highest correlated unigrams and bigrams for explicit anthropomorphic language. We have anonymised the names of proprietary products.

In the annotation schema two levels of anthropomorphism are present: Ambiguous and Explicit. We determine lexical features that distinguished between these two categories by using the same methodology as above, but adapting the transformation of the labels. To conduct this analysis, we only considered the portion of our corpus that was annotated as either ambiguous anthropomorphic or explicit anthropomorphic. We assigned ambiguous anthropomorphic texts a score of zero and explicit anthropomorphic a score of one and calculated Pearson correlation against the one-hot encoded vectors. The results of this analysis are shown in Table 3.

The term with the highest correlation is 'student'. This typically occurs in the context of 'student models' as used in the task of model distillation. It is also notable that the names of several proprietary products are correlated, indicating that descriptions of commercial activities are more likely to be explicitly anthropomorphic than ambiguous anthropomorphic. The bigrams that are identified indicate elements of anthropomorphism ('said that', 'faithful models', etc.). There is also some noise in this analysis, with 'Then', 'them to' and 'her to' also included. The noise is likely due to the small corpus size (there were only 160 instances of explicit anthropomorphism).

4.3 Journalistic Writing

Correlation	Term	Freq
0.184	ask	12
0.17	respond	5
0.151	*Product 4*	17
0.151	human	180
0.143	scenario	18
0.138	questions	91
0.138	response	39
0.136	though	8
0.128	visual	51
0.125	point	10
0.128	what it	5
0.128	respond to	5
0.111	and destroy	2
0.111	to prompts	3
0.111	to kill	3
0.111	while the	12
0.111	you ask	3
0.111	responses to	4
0.108	it was	11
0.09	data points	5

Table 4: Highest correlated unigrams and bigrams for anthropomorphic language in the journalism sector. We have anonymised the names of proprietary products.

Finally, we present an analysis of features that are indicative of anthropomorphism in journalistic writing. We analysed the portion of the corpus extracted from journalistic sources and compared examples of non-anthropomorphic language to examples of ambiguous or explicit anthropomorphic language using the methodology described in Section 4.1. The results of this analysis are presented in Table 4.

The examples of anthropomorphic language from journalistic texts make use of metaphorical or extreme language such as 'destroy' or 'kill'. Journalistic sources are sensational in their reporting of anthropomorphic language as evidenced by terms such as 'destroy' and 'kill'. Anthropomorphic terms in journalistic sources focus on the interaction of humans with LLMs as evidenced by terms such as 'ask', 'respond' and 'question' indicating anthropomorphised dialogue.

5 Sentence Classification

This section reports on the development of text classification methods to distinguish between anthro-

pomorphic and non-anthropomorphic sentences.

5.1 Data Processing

We split the available data into train (80%) and test (20%) partitions ensuring that the splits were stratified and that both genres occurred evenly across each subset. The distribution of labels was also preserved. As Explicit Anthropomorphism is a minority class (3.7%), we conflated this class with Ambiguous Anthropomorphism leading to a two-class problem with the labels: 'Non-Anthropomorphic' and 'Anthropomorphic'.

In our corpus, 77.0% of identified claims were labelled as non-anthropomorphic. Imbalanced data can lead to a classifier overly relying on one class and so we explored two different methods of balancing our classes for the training set. We did not perform any adjustments to the data distribution in the test set to reflect the real-world class distribution. Firstly, we employed down-sampling of the majority class. In this setting, we selected a random sample of the Non-Anthropomorphic examples (2678) in our corpus which was the same size as the Anthropomorphic examples (778 examples of each class). The down-sampling method led to perfectly balanced data, but involved discarding 1900 examples of non-anthropomorphic text. We further explored up-sampling the minority class through the use of the Parrot Paraphraser (Damodaran, 2021). The Parrot Paraphraser relies on a T5-based paraphrase model (Raffel et al., 2020) and provides metric-based filtering for adequacy, fluency and lexical diversity of the returned paraphrases. We used the Parrot Paraphraser in the default configuration to produce an additional 1538 examples of anthropomorphised claim sentences. We again, balanced the classes in this setting to give 2316 examples in each class. Statistics for each train setting and for the test setting are given in Table 5.

Partition	Sampling	NA	A
Test	None	663	221
Train	None	2678	778
Train	Down	778	778
Train	Up	2316	2316

Table 5: Data settings used for evaluation of sentence classification. Down-sampling and up-sampling are used to create a balanced training set, however the test-set remains imbalanced throughout all experiments reflecting the nature of the corpus. NA refers to Non-anthropomorphic annotations. A refers to Anthropomorphic annotations consisting of explicit anthropomorphism and ambiguous anthropomorphism.

5.2 Baseline Approaches

We provide minority and majority class baselines (i.e., assigning the anthropomorphic, or non-anthropomorphic labels to all classes respectively). This approach demonstrates a baseline effect of a classifier which has not adapted to the task and fails to make any discriminative judgements. We also include two randomised baselines. Firstly, we include a random baseline where each class is equally likely to be assigned (random 1:1). Secondly, we also include a random baseline where the non-anthropomorphic label is 3 times more likely than the anthropomorphic label, reflecting our data distribution. These approaches represent the base performance of a classifier which is making randomised decisions, either with respect to the class label, or with respect to the data distribution. We provide these baselines as we believe they are a reasonable means of contextualisation of the results from the other approaches as described below.

5.2.1 ML Classifiers with SciKitLearn

We used Random Forest (Breiman, 2001) and SVM (Cortes and Vapnik, 1995) from SciKitLearn (Pedregosa et al., 2011). To convert each sentence into a numerical format we employed (a) BOW vectorisation via the CountVectorizer library in SciKitLearn and (b) sentence embeddings using Sentence Transformer (Reimers and Gurevych, 2019). We used the default configurations in SciKitLearn for the Random Forest and SVM and did not tune the hyperparameters in each case (due to the small size of our data).

5.2.2 BERT-based classifiers with Transformers

We used the following models via the Transformers library in Python downloaded from the HuggingFace hub:

`google-bert/bert-base-uncased`
`google-bert/bert-large-uncased`
`FacebookAI/roberta-base`
`FacebookAI/roberta-large`
`allenai/scibert_scivocab_uncased`

All models were fine-tuned against the training set under each train-setting for 5 epochs using the AdamW optimiser with learning rate of 4×10^{-5}. In some cases the model failed to converge, in which case the training process was repeated.

Baseline	Acc	Anthropomorphic		
		R	P	F1
Majority Class	0.750	0.000	0.000	0.000
Minority Class	0.250	1.000	0.250	0.400
random 1:1	0.494	0.471	0.240	0.318
random 3:1	0.618	0.226	0.230	0.228

Table 6: Baseline results for anthropomorphism classification

5.2.3 Prompt Engineering with MLX

We also experimented with MLX, a library for MacOS for implementing LLMs. In this case we used an 8B version of Llama3.1 (Grattafiori et al., 2024), specifically the model available at the HuggingFace Hub here: "mlx-community/Meta-Llama-3.1-8B-Instruct-bf16", which is 4-bit quantised. We used this model for in-context learning (Wei et al., 2022), in which case we simulated a multi-turn conversation between the LLM and the user, demonstrating examples of anthropomorphic and non-anthropomorphic sentences and their classifications. The model was then presented with a new sentence from the test set and the response it generated was interpreted as the classification. We also fine-tuned Llama for this task under the same setting using examples from the training set. We also include a zero-shot classification setting.

5.2.4 Closed-source LLMs

We additionally performed in-context learning in a 100-shot setting using the same prompts as before and a 100-shot in-context learning setting drawn from the training set. We accessed GPT-4o and GPT-4 Turbo on the 8th November 2024 via the web-based API. The total costs were 7 dollars for GPT-4o and 33 dollars for GPT-4 Turbo for a single run through the entire test set (n=884) in each case. We compare these results to LLama3.1 in a 100-shot setting. All results are shown in Table 9.

6 Results

We present results for baseline approaches (Table 6), machine learning classifiers (Table 7), prompt engineering (Table 8) and GPT-4 models (Table 9). For each table, we have presented Accuracy (the percentage of all correct instance regardless of class), as well as the Precision, Recall and F1-score for the anthropomorphic class.

We provided four heuristic baseline approaches examining different approaches to classification

Train	Method	Acc	Anthropomorphic		
			R	P	F1
O	SVM-BOW	0.753	0.018	0.800	0.035†
	SVM-ST	0.750	0.018	0.500	0.035†
	RF-BOW	0.753	0.018	0.800	0.035†
	RF-ST	0.750	0.018	0.500	0.035†
	bert-base	0.784	0.403	0.601	0.482
	roberta-base	0.739	0.059	0.361	0.101
	scibert-base	0.768	0.362	0.556	0.438
D	SVM-BOW	0.613	0.475	0.317	0.380
	SVM-ST	0.617	0.647	0.354	0.458
	RF-BOW	0.613	0.475	0.317	0.380
	RF-ST	0.617	0.647	0.354	0.458
	bert-base	0.670	0.706	0.407	0.517
	roberta-base	0.708	0.756	0.450	**0.564**
	scibert-base	0.660	0.715	0.399	0.512
U	SVM-BOW	0.683	0.416	0.379	0.397
	SVM-ST	0.657	0.579	0.379	0.458
	RF-BOW	0.683	0.416	0.379	0.397
	RF-ST	0.657	0.579	0.379	0.458
	bert-base	0.777	0.462	0.567	0.509
	roberta-base	0.784	0.471	0.584	0.521
	scibert-base	0.757	0.339	0.521	0.411

Table 7: The results of classifying anthropomorphic and non-anthropomorphic sentences. Best F1-score in bold. Three training settings are explored: **O**riginal, **D**own-sample and **U**p-sampled. †The F1 scores for these two values appear the same due to rounding. This is an effect of the low-recall in both instances masking the substantial difference in precision.

in our corpus as demonstrated in Table 6. As our data is split 75:25 between the majority (Non-anthropomorphic) and minority (Anthropomorphic) classes, we observe that the majority and minority baselines reflect this. We have only reported F1-score for the anthropomorphic class as this is the feature we are trying to identify. This means that whilst the accuracy for the majority class baseline is 0.75 (all the non-anthropomorphic examples were correctly identified), the Recall, Precision and consequently the F1-score are all 0, as no non-anthropomorphic examples were identified. Conversely, the minority class baseline does much worse in terms of accuracy (0.25), but has perfect recall by retrieving all anthropomorphic examples.

We also provide two randomised baselines. The random 1:1 baseline has a lower accuracy, but higher F1 score (owing to a higher recall) than the random 3:1 score. This is effectuated by the random 1:1 baseline over-predicting the prevalence of anthropomorphic terms in the data. Nevertheless, the random 1:1 baseline still has a lower F1-score than the Minority class baseline.

These baselines serve to help the reader understand and interpret the behaviour of the classifiers that we present in our results. Whilst we will see

Method	N	Acc	Anthropomorphic		
			R	P	F1
0-shot	0	0.707	0.389	0.410	0.399
ICL	1	0.537	0.738	0.317	0.444
	2	0.467	0.824	0.296	0.436
	3	0.518	0.765	0.311	0.442
	4	0.577	0.692	0.333	0.450
	5	0.399	0.869	0.277	0.420
	7	0.506	0.733	0.300	0.426
	9	0.563	0.674	0.322	0.436
FT	1	0.644	0.566	0.363	0.442
	2	0.650	0.529	0.363	0.431
	3	0.567	0.683	0.325	0.441
	4	0.648	0.538	0.363	0.434
	5	0.733	0.258	0.442	0.326
	7	0.698	0.394	0.395	0.395
	9	0.679	0.457	0.381	0.416

Table 8: The results of using LLama3.1 to classify anthropomorphic and non-anthropomorphic sentences. ICL refers to In-context Learning. FT refers to Fine-tuning. The number of examples (N-shots) at inference time is also presented.

Method	Acc	Anthropomorphic		
		R	P	F1
GPT4o	0.763	0.181	0.588	0.277
GPT-4-Turbo	0.766	0.176	0.609	0.274
Llama3.1	0.729	0.308	0.439	0.362

Table 9: Comparison of GPT-4 models and Llama in a 100-shot in-context learning setting to classify anthropomorphic and non-anthropomorphic sentences.

that many of our classifiers attain a high accuracy, many do so at the severe compromise of F1-score, indicating that all or most predictions were to the majority class. We also see that there is a lower bound on the F1-score as evidenced by the random classification. Any systems scoring higher than this can be interpreted as performing better than random, i.e., indicating learning has taken place.

We tested two Machine learning approaches: SVMs and Random Forests with features coming from a Bag-of-words and from Sentence Transformers. Our results in Table 7 showed little difference between these approaches and typically that the classifiers were not able to reliably predict the presence of anthropomorphic language in a sentence with the accuracy and F1-scores falling below baseline in most cases. The Sentence Transformer features gave higher scores than the BOW features in the down-sampled and up-sampled settings, but not in the original setting where minimal learning took place as evidenced by the extremely low recall. We note that in all cases the SVM and RF algorithms returned the same scores under the same settings indicating that the same decision manifold was learnt in each case. This indicates that the task is more complex than simply relying on word features (i.e., no word is a strong indicator) and that the sentence embeddings did not provide sufficient information for the classifiers. It may be possible that a larger dataset of anthropomorphic language would permit the algorithms to learn a more comprehensive representation of the feature space and perform better at test time.

Following on from this, we also tried three transformer based approaches for sentence classification. We observe some slight improvement in Roberta as compared to Bert in the down-sampled and up-sampled settings (see Table 7). We additionally noted that Scibert performed worse than Bert and Roberta in the down-sampled and up-sampled settings. In the original setting Roberta did not accurately retrieve anthropomorphic examples (Recall = 0.059), however Bert still marginally outperformed Scibert on all metrics. Our best performing system in terms of the F1 metric was Roberta-base in the down-sampled setting. This returns an F1 score of 0.564 made up of a recall score of 0.756 and a precision of 0.450 indicating that the model overestimated the degree of anthropomorphism in the corpus (i.e., this result occurs because the model tended to label non-anthropomorphic sentences as anthropomorphic).

We explored three experimental settings for our training dataset in Table 7, whilst keeping the test data as a constant split in all experiments. The original setting had a 3:1 distribution of non-anthropomorphic and anthropomorphic sentences whereas the down-sampled and up-sampled data had a 1:1 ratio in each case. Balancing the classes in the training set led to a clear improvement in classification ability for all models. Whilst the up-sampled data typically exhibits a higher precision than the down-sampled data, the overall F1-scores for the transformer-based methods are lower, owing to a drop in recall for these methods. Whereas we had expected that including more data would lead to an overall improvement in scores this was not the case and may well be due to the fact that our up-sampled data included synthetic examples

Method	Acc	Anthropomorphic		
		R	P	F1
Journalism	0.724	0.553	0.712	0.622
Abstracts	0.771	0.505	0.679	0.579

Table 10: F1 scores for separate genre subsets within our corpus.

that were not suitable representations of the type of information seen at testing time.

In Table 8 we present the results of our experiments with Llama 3.1 (the most advanced version of Llama available at the time of experimentation) in an ICL and Fine-tuned setting with 1-9 examples as well as a zero-shot approach. The zero-shot experiment demonstrates that an LLM such as Llama is able to correctly answer in some cases for our task without any task specific information being introduced as the F1-score of 0.399 is above the randomised baselines. Compared to zero-shot, we can observe that strategies such as ICL and fine-tuning with 1-9 examples improved the F1-score marginally, but that there was no significant improvement by including more examples or between both techniques.

We also compare Llama 3.1 in a 100-shot ICL-setting to the equivalent experiment with the closed source OpenAI models GPT4o and GPT-4-Turbo in Table 9. Whilst the accuracy is improved for the GPT models compared to Llama3.1, the F1-score indicates that the GPT models underperform providing classification results which are indistinguishable from a random baseline. We were not able to identify a strategy using newer LLMs such as Llama and GPT that performed better for our corpus than the Roberta-base system which makes use of a much smaller version of the transformer architecture.

We present an additional analysis of our results in Table 10, where we show the performance of sentence classification for each of the sub-genres represented in our dataset. These results were produced by using Roberta-Base and training in the down-sampled setting (i.e., the system with the best F1-score in our prior experiment). The results show that the F1 score for sentences from the journalism genre is higher than for the scientific abstracts.

7 Discussion

In writing this work (and other works on the topic) it became apparent to the first author that reporting on LLMs is difficult, and maybe impossible, without leaning on anthropomorphised terminology. As such, the description of the methods and results herein necessarily contains some anthropomorphism and the authors have deliberately left this in-situ. We are not advocating for the abolishment of anthropomorphised terminology, but rather seeking to better understand and quantify the phenomenon. The value of an anthropomorphism classification tool is not to punish authors who lean on metaphors, but rather to better equip scientists and the general public with tools for understanding the way we describe LLMs.

Anthropomorphism is of course not limited to the study of large language models and one may envision a similar study on other technology (e.g., self-driving cars, drones, etc.). We do not seek to make claims about anthropomorphisation outside of the realm of LLMs, however we do expect that similar phenomena are apparent and that the work here may be a good starting point for adaptation to other areas of study.

An interesting finding of our work is that despite extensive study, we were unable to improve the performance of the LLM approach beyond that of the random baselines (e.g., GPT4o/GPT4-Turbo in the 100-shot ICL setting). A deliberately anthropomorphised interpretation of this finding may be that LLMS don't *know* when they are being anthropomorphised. Of course, our study is non-exhaustive and there may well be alternative methods of LLM-prompting strategies beyond our study that would yield improved results.

8 Conclusion

In this work, we have presented an analysis of anthropomorphism in scientific reporting of LLMs as well as experiments on developing new classifiers for sentence-level anthropomorphism. Our most promising results show that we are able to produce sentence classifications which outperform reasonable baselines. The use of Bert-based models was most effective in our study as compared to machine learning classifiers or prompt engineering. Our work lays the foundation for future studies on anthropomorphism classification at the sentence level and beyond.

References

Gavin Abercrombie, Amanda Cercas Curry, Tanvi Dinkar, Verena Rieser, and Zeerak Talat. 2023. Mi-

rages. on anthropomorphism in dialogue systems. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4776–4790, Singapore. Association for Computational Linguistics.

Leo Breiman. 2001. Random forests. *Machine learning*, 45:5–32.

Phillip Brooker, William Dutton, and Michael Mair. 2019. The new ghosts in the machine: 'Pragmatist' AI and the conceptual perils of anthropomorphic description. *Ethnographic studies*, 16:272–298.

Myra Cheng, Alicia DeVrio, Lisa Egede, Su Lin Blodgett, and Alexandra Olteanu. 2024a. " i am the one and only, your cyber bff": Understanding the impact of genai requires understanding the impact of anthropomorphic ai. *arXiv preprint arXiv:2410.08526*.

Myra Cheng, Kristina Gligoric, Tiziano Piccardi, and Dan Jurafsky. 2024b. AnthroScore: A computational linguistic measure of anthropomorphism. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 807–825, St. Julian's, Malta. Association for Computational Linguistics.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning*, 20(3):273–297.

Prithiviraj Damodaran. 2021. Parrot: Paraphrase generation for nlu.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

David Gros, Yu Li, and Zhou Yu. 2022. Robots-dont-cry: Understanding falsely anthropomorphic utterances in dialog systems. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3266–3284, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Nanna Inie, Stefania Druga, Peter Zukerman, and Emily M Bender. 2024. From" ai" to probabilistic automation: How does anthropomorphization of technical systems descriptions influence trust? In *The 2024 ACM Conference on Fairness, Accountability, and Transparency*, pages 2322–2347.

Yu Li, Devamanyu Hazarika, Di Jin, Julia Hirschberg, and Yang Liu. 2024. From pixels to personas: Investigating and modeling self-anthropomorphism in human-robot dialogues. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 9695–9713, Miami, Florida, USA. Association for Computational Linguistics.

Zachary C Lipton and Jacob Steinhardt. 2019. Troubling trends in machine learning scholarship: Some ml papers suffer from flaws that could mislead the public and stymie future research. *Queue*, 17(1):45–77.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830.

Adriana Placani. 2024. Anthropomorphism in ai: hype and fallacy. *AI and Ethics*, pages 1–8.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Arleen Salles, Kathinka Evers, and Michele Farisco. 2020. Anthropomorphism in AI. *AJOB Neuroscience*, 11(2):88–95.

Matthew Shardlow and Piotr Przybyła. 2024. Deanthropomorphising nlp: Can a language model be conscious? *PLOS ONE*, 19(12):1–26.

Matthew Shardlow, Ashley Williams, Charlie Roadhouse, Filippos Ventirozos, and Piotr Przybyła. 2025. Exploring supervised approaches to the detection of anthropomorphic language in the reporting of NLP venues. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 18010–18022, Vienna, Austria. Association for Computational Linguistics.

David Watson. 2019. The Rhetoric and Reality of Anthropomorphism in Artificial Intelligence. *Minds and Machines*, 29(3):417–440.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

TypePilot: Leveraging the Scala type system for secure LLM-generated code

Alexander Sternfeld
Institute of Entrepreneurship & Management, HES-SO
Le Foyer, Techno-Pôle 1
Sierre, Switzerland
alexander.sternfeld@hevs.ch

Andrei Kucharavy
Institute of Informatics, HES-SO
Techno-Pôle 3
Sierre, Switzerland
andrei.kucharavy@hevs.ch

Ljiljana Dolamic
Cyber-Defence Campus
armasuisse, Science and Technology
Thun, Switzerland
ljiljana.dolamic@armasuisse.ch

Abstract

Large language models (LLMs) have shown remarkable proficiency in code generation tasks across various programming languages. However, their outputs often contain subtle but critical vulnerabilities, posing significant risks when deployed in security-sensitive or mission-critical systems. This paper introduces *TypePilot*, an agentic AI framework designed to enhance the security and robustness of LLM-generated code by leveraging strongly typed and verifiable languages, using Scala as a representative example. We evaluate the effectiveness of our approach in two settings: formal verification with the Stainless framework and general-purpose secure code generation. Our experiments with leading open-source LLMs reveal that while direct code generation often fails to enforce safety constraints, just as naive prompting for more secure code, our type-focused agentic pipeline substantially mitigates input validation and injection vulnerabilities. The results demonstrate the potential of structured, type-guided LLM workflows to improve the SotA of the trustworthiness of automated code generation in high-assurance domains.

1 Introduction

In recent years, large language models (LLMs) have become powerful tools for assisting in software development, from generating boilerplate code to proposing non-trivial algorithmic implementations (Wang and Chen, 2023; Chen et al., 2021). Their fluency in natural and programming languages allows developers to interact with them without disrupting their workflow, accelerating the development lifecycle. However, as LLMs are increasingly used to write production code, concerns have emerged about the reliability and security of the generated output. Multiple studies and real-world analyses have shown that LLMs can introduce subtle yet serious vulnerabilities (Pearce et al., 2025).

This issue becomes particularly acute in the domain of mission-critical systems—software systems whose failure can lead to catastrophic outcomes, including physical harm, financial loss, major operational disruptions, or loss of life (Gabriel et al., 2022). Such systems are often implemented in strongly typed, safety-oriented programming languages like Coq, Scala, or more recently Rust, where the type system is a central mechanism for enforcing correctness and preventing classes of bugs at run time. Despite these safeguards, vulnerabilities still surface, often due to logical oversights, incorrect assumptions, or abstraction mismatches at boundaries. A well-known example is the 1999 NASA Mars Climate Orbiter failure, where one subsystem produced output in imperial units while another expected metric, leading to the spacecraft's loss due to an undetected discrepancy at the interface between components (Harish, 2025). A notable recent example of such a vulnerability in action occurred in January 2023, when a critical FAA system failure, later traced to a corrupted configuration file, led to the temporary grounding of all flights across the United States (reuters, 2023).

While LLMs are increasingly capable of detecting potential code vulnerabilities, they often fall short in generating robust corrections (Kulsum et al., 2024; Pearce et al., 2022). Our work addresses this gap. By focusing on Scala - a widely used language with extensive codebase on GitHub and documentation on StackOverflow (O'Grady, 2025), we propose TypePilot, an agentic AI approach that not only leverages the detection capabilities of LLMs but actively guides them to ex-

Proceedings of the First Interdisciplinary Workshop on OMMM associated with RANLP 2025,
pages 95–103, Varna, Bulgaria, Sep 8-10, 2025.
https://doi.org/10.26615/978-954-452-101-1-011

ploit the expressiveness of the Scala type system to add safety guarantees. By structuring interactions, TypePilot guides LLMs to generate and refine code that adheres to strict safety and correctness properties.

This paper is structured as follows: Section 2 describes the related literature, after which Section 3 outlines the methodology. Next, the results are presented in Section 4. Last, Section 6 concludes the paper and provides directions for future research. The code and results related to this paper are publicly available in this Github Repository.

2 Related work

2.1 LLMs for code generation

The use of large language models for code generation has grown rapidly, with coding specific models demonstrating impressive capabilities across a wide range of programming languages and tasks. However, several studies have pointed out that these models often produce code that is syntactically correct but semantically flawed or insecure. For instance, Pearce et al. (2025) shows that GitHub Copilot produces vulnerabilities in approximately 40% of test cases based on the top 25 Common Weakness Enumeration list from MITRE. Similarly, Khoury et al. (2023) show that ChatGPT generates vulnerable code in 16 out of 21 test cases, using a variety of programming languages targeting a diverse set of vulnerabilities.

There have been attempts to use separate LLMs in combination with sophisticated prompting strategies to patch such vulnerabilities. However, these approaches remain brittle, with models often misunderstanding the root cause or proposing fixes that break functionality. Kulsum et al. (2024) show that LLMs have difficulty in patching vulnerabilities that are either complex or linked to the project design. Similarly, Pearce et al. (2022) show that LLMs are not yet able to autonomously patch code vulnerabilities in real-world scenarios.

Our work builds upon these findings by exploring a different method for mitigating vulnerabilities - leverage the properties of strongly typed coding languages. We use *agentic AI*, where LLMs cooperatively operate as autonomous agents, which has been shown to result in better generations (Kumar et al., 2025; Wang et al., 2025).

3 Methodology

We will now describe the methodology that is used in this research. First, the models that are used in this research are specified. Then, we consider the ability of LLMs to generate code using the formal verification framework Stainless. Last, we consider the general case of type-system rooted vulnerabilities.

3.1 Model usage

Throughout this research, we use open-source models, with a focus on specialized coding models. Specifically, we used the coding models `Qwen/Qwen2.5-Coder-32B-Instruct`, `deepseek-ai/deepseek-coder-33b-instruct` and `codellama/CodeLlama-70b-Instruct-hf`. Additionally, we used the regular conversational models `meta-llama/Meta-Llama-3-70B`, `deepseek-ai/DeepSeek-R1-Distill-Llama-70B` and `Qwen/Qwen3-32B`.

3.2 Stainless

We first aim to leverage the formal verification framework Stainless (Lab for Automated Reasoning and Analysis, 2025) to improve the robustness of LLM-generated code. Formal verification refers to the use of mathematical methods to prove that a program satisfies certain correctness properties. Stainless is one of the most widely used verification frameworks in Scala, with extensive documentation. Stainless verifies whether Scala code meets user-specified safety properties by attempting to construct proofs over the code. To enable this, the code must explicitly state what is to be proven, and provide the necessary logical structure for the proof, using a subset of Scala tailored for verification.

To this end, we use both zero-shot and two-shot prompting to have a LLM both generate the code and the conditions. Figure 2 displays the prompt that is used in the two-shot prompting setting. The two examples that are given to the LLM are Stainless code for finding the maximum between two values and for returning the size of a list. The exact examples can be found in the Github Repository.

As displayed in Table 1, we use three simple tasks for evaluating the LLMs in the context of formal proofs: calculating Fibonacci number n, calculating the factorial of an input and assessing whether list a is a sublist of list b. The main vulnerability that the generated conditions should prevent

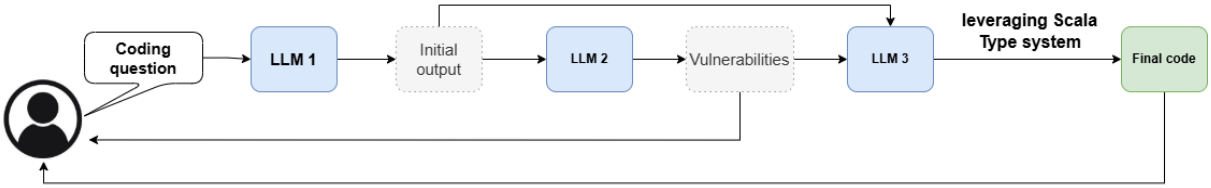

Figure 1: The full pipeline for the generation of code using TypePilot. After the initial generation of the code, the vulnerabilities are detected by a separate instance of the LLM. Then, a final LLM is prompted to leverage the Scala type system to improve the initial code, given the detected vulnerabilities.

are input variables that are invalid, such as a negative input for a factorial function. Additionally, the functions should also be robust to inputs that are too large and may cause an overflow error.

Generation of stainless code

<question> Use the stainless framework to write verifiable scala code for fewshot example 1 </question>

<answer> fewshot answer 1 </answer>

<question> Use the stainless framework to write verifiable scala code for fewshot example 2 </question>

<answer> fewshot answer 2 </answer>

<question> Use the stainless framework to write verifiable scala code for function description </question>

Figure 2: Prompt used to generate the Stainless code.

3.3 General case: type-system rooted vulnerabilities

As Stainless targets a niche subset of Scala applications, we also consider a more general setting. Specifically, we focus on two vulnerability categories: insufficient input constraints and injection attacks. In particular, we examine HTML, Bash, and URL injections—common security risks in back-end web development, especially when handling user inputs through web forms. The specific test cases are shown in Table 1. To assess the performance of LLMs on these tasks, we consider the following settings:

Code generation in *robust* setting

You are a scala code generator. You will be given a task description and you will generate the code for it. The code should start with "'scala and end with "'. Pay attention to the safety and robustness of the code, and leverage the Scala type system - for example ADTs, refined types, traits, sealed traits - where needed to make the code safer. The task is: user input

Figure 3: Prompt used to generate the code in the robust prompting setting.

- **Baseline:** directly prompting a LLM to generate the code
- **Robust prompting:** directly prompting a LLM to generate the code, while emphasizing that the LLM should leverage the Scala type system to make the code robust to potential vulnerabilities.

Stainless	**General case: type-system rooted vulnerabilities**	
	Input constraints	*Code injection*
Calculating a fibonacci number	Calculating a fibonacci number	Greeting a user with HTML
Calculating the factorial of a number	Calculating the factorial of a number	Making a list of comments with HTML
Asserting if list a is a sublist of list b	Calculating a matrix multiplication	Searching a file using bash
	Calculating a matrix convolution	Pinging a host using bash
		Creating a redirect URL with HTML

Table 1: The test cases used to evaluate the LLMs in each of the settings. The most left column shows the test cases used to evaluate the performance of LLMs in generating code using the Stainless framework. The second and third column show the test cases for the general case looking at type-system rooted vulnerabilities.

Code generation using TypePilot

Initial code generation
You are a Scala code generator. You will be given a task description and you will generate the code for it. The code should start with "'scala and end with "'. The task is: user input

Vulnerability detection
You will be given a task description and generated code. Your task is to find potential vulnerabilities in the code that could lead to security issues or unexpected behavior. Solely describe the vulnerabilities, do not give me any code. Here is the task: user input Here is the previous code: initial output

Final code generation
You are a Scala code generator. You will be given a task description, generated code, and vulnerabilities that should be addressed. Your task is to improve the code by using the Scala type system - for example ADTs, refined types, traits, sealed traits - to address the vulnerabilities. The code should start with "'scala and end with "'. Here is the task: user input. Here is the previous code: initial output Here are the vulnerabilities: vulnerabilities

Figure 4: Prompts used to generate the initial code, the vulnerabilities and the final code in TypePilot. The final prompt guides the LLM to use the Scala type system to make the code more robust.

- **TypePilot:** use the agentic AI framework as displayed in Figure 1 to generate the code. After prompting a first LLM to generate the initial code, we ask a second LLM to detect the vulnerabilities in this code. We then ask a third instance of the LLM to improve the initial code using the Scala type system, to make it robust to the detected vulnerabilities.

The prompt that is used in the robust generation setting can be found in Figure 3. Similarly, Figure 4 shows the prompts that are used with TypePilot. In the baseline setting, we use the same prompt that is used for the initial code generation in TypePilot. For each of the models described in Section 3.1 we run each of the settings.

3.4 Comparison to existing work

Research on secure code generation using large language models (LLMs) remains limited, despite growing concerns about vulnerabilities in automatically generated code. A recent survey by Dai et al. (2025) highlights that most current approaches rely heavily on training data or static analysis tools, restricting their generalizability. Methods such as SafeCoder (He et al., 2024) and SVEN (He and Vechev, 2023) fine-tune LLMs with curated secure code datasets, and are thus inherently dependent on the availability and quality of specialized training corpora. Moreover, the fine-tuned LLMs do not generalize well to unseen vulnerabilities or programming languages. Similarly, PromSec (Nazzal et al., 2024) optimizes prompts through static

	Qwen-2.5-Coder (32B)			CodeLlama (70B)			Deepseek-coder (33B)		
	Baseline	Robust	TypePilot	Baseline	Robust	TypePilot	Baseline	Robust	TypePilot
Average age									
- Correct for regular input	✓	✓	✓	✓	✓	✓	✓	✓	✓
- Handle empty lists	✓	✓	✓	✓	✓	✓	✓	✓	✓
- Handle negative ages	✗	✗	✓	✗	✗	✓	✗	✗	✗
Fibonacci number N									
- Correct for regular input	✓	✓	✓	✓	✓	✓	✓	✓	✓
- Check for negative N	✗	✓	✓	✗	✗	✗	✗	✗	✓
- Handles large values of N	✗	✓	✓	✗	✗	✗	✗	✓	✓
Matrix multiplication									
- Correct for regular input	✓	✓	✓	✓	✓	✓	✓	✓	✓
- Check for empty matrices	✓	✗	✓	✗	✗	✗	✗	✗	✗
- Check for dimension matching	✓	✓	✓	✓	✗	✓	✗	✓	✓
Matrix convolution									
- Correct for square matrix input	✓	✓	✓	✗	✗	✗	✓	✓	✓
- Correct for regular matrix input	✓	✓	✓	✗	✗	✗	✓	✓	✓
- Handles rectangular kernels	✗	✗	✓	✗	✗	✓	✗	✗	✗
- Checks for empty kernel	✗	✓	✓	✗	✗	✓	✗	✗	✓
- Checks for empty matrix	✗	✓	✓	✗	✗	✓	✗	✗	✓
- Handles even sized kernels	✗	✗	✓	✗	✗	✗	✗	✗	✗

Table 2: Manual evaluation of the generated code regarding input constraints. For each case, ✓ indicates that the code is robust to the vulnerability, whereas ✗ indicates that the code is not robust to the vulnerability.

	Qwen-2.5-Coder (32B)			CodeLlama (70B)			Deepseek-coder (33B)		
	Baseline	Robust	TypePilot	Baseline	Robust	TypePilot	Baseline	Robust	TypePilot
HTML greeting									
Correctness and compilation	✓	✓	✓	✓	✓	✓	✓	✓	✗
Robust to injection	✗	∼	✓	✗	✓	✓	✗	✓	✓
HTML comments									
Correctness and compilation	✓	∼	✓	✓	✗	∼	✓	✓	✓
Robust to injection	✗	✓	✓	✗	✗	✓	✗	∼	✓
Bash file search									
Correctness and compilation	✓	✓	✓	✓	✓	✓	✓	✓	✓
Robust to injection	✗	✗	✓	✗	✗	✓	✗	✗	✓
Bash host ping									
Correctness and compilation	✓	✓	✓	✓	✗	✓	✓	✓	✓
Robust to injection	✓	✓	✓	✗	✗	✓	✗	✓	✓
URL redirect									
Correctness and compilation	✓	✓	✓	✓	✓	✓	✓	✓	✓
Robust to injection	✗	∼	∼	✗	✓	✓	✗	✗	✓

Table 3: Manual evaluation of the generated code regarding code injection. For each case, ✓ indicates that the code is robust to the vulnerability, whereas ✗ indicates that the code is not robust to the vulnerability.

analyzers, but also relies on labeled data and an external code-specific vulnerability scanner.

In contrast, our approach does not rely on task-specific training data or external static analyzers. Instead, it leverages the expressive power of strongly typed languages to enforce security constraints directly in the generated code. Because most existing methods depend on curated datasets or vulnerability scanners (as discussed above), there are few established baselines tailored to strongly typed languages like Scala or Rust. Given this gap, it is most appropriate to compare our method against prompting-based baselines in addition to the base model. Following Vero et al. (2025), we include a baseline where the model is given a general security reminder, which we call *robust prompting*. We also evaluate against *Self-Planning*, a coding-specific prompting strategy introduced by Jiang et al. (2024). Self-Planning is a two-stage prompting framework in which the LLM first generates a high-level plan for the coding task, after which it implements the plan in code.

4 Results

4.1 Stainless

In general, we see that none of the models is capable of consistently generating Stainless code that correctly compiles. Upon manual inspection, we found two main failure modes across all models. First, each of the LLMs regularly uses concepts that are present in Scala but not available in Stainless. As Stainless is a verification framework targeting a restricted subset of Scala, many features of full Scala—such as certain standard library functions—are unavailable. To illustrate, in the generated code from `Qwen/Qwen3-32B` for the verification of a sublist relation, the function `List.sliding` is used. However, the sliding operation is not defined for Stainless `List` objects. Similarly, in a generated code snippet the operation `println` was used, which is not available in Stainless. Second, the generated code often contains syntax errors. Whereas syntax errors could be resolved relatively easily by users, the usage of Scala components in Stainless is not trivially repaired. We hypothesize that the lack of performance is caused by a lack of training data related to Stainless, given that it is a niche framework. This observation is consistent with findings from other domains, for example, Fan et al. (2025) found that LLMs struggle to generate verifiable specifications using the VeriFast verification framework for C, despite preserving functional behavior. In appendix B, we provide a notable instance in which the generation avoids formal verification by using `@library` annotations.

4.2 General setting

Given that LLMs are not able to write compilable Stainless code, we shift our attention to a more general Scala setting, as described in Section 3.3. We

consider two types of vulnerabilities: insufficient input constraints and code injection. The generated code is available in the anonymized repository.

4.2.1 Input constraints

Table 2 shows the results for each of the test cases for each of the models. For each of the models, ✗ indicates that the resulting code was not robust to the indicated vulnerability. The results show that in the baseline setting, the models are capable of generating functions that provide the correct output in a normal setting. However, the models are not capable of handling edge cases correctly. To illustrate, none of the models can correctly handle negative ages or a negative input to a Fibonacci function. We see that in the *robust* setting, models perform slightly better, and tend to be robust to some of the vulnerabilities. However, for none of the models the code is fully robust. With TypePilot, we obtain the best performance, with models generally being robust to most vulnerabilities related to input constraints.

When comparing the models, we observe that `Qwen-2.5-Coder (32B)` performs the best, passing all our checks when using TypePilot. In contrast, `CodeLlama (70B)` does not perform well, remaining vulnerable to a number of cases in each of the settings, highlighting the importance of study of specific code-generating LLM models.

4.2.2 Code injection

The second type of vulnerability we consider is code injection. Table 3 displays the results for each of the models, where ✗ indicates that the code is vulnerable to injection, ✓ indicates that code is robust to injection, and ∼ indicates that the code is partially robust to injection. The results show that in the baseline setting virtually all generated code is vulnerable to code injection. Robust prompting improves the performance, resulting in fewer vulnerabilities. As before, TypePilot achieves the best performance, with robust code generations in almost all settings.

4.2.3 Usage of the Scala type system

In the new framework, the Scala type system is used as a central tool to guide the generation of secure code. By prompting LLMs to leverage features such as sealed traits, smart constructors, and refined return types, we enable the generation of programs that encode correctness directly into their type signatures. This stands in contrast to baseline generations, which operate on unconstrained primitives and rely on ad hoc runtime logic to handle edge cases and errors.

Figures 5 shows an examples of code generated in the baseline and in the agentic AI framework, for the same test case and model. Figure 5 shows that in the baseline version, the `averageAge` function takes a `List[Int]` and performs a division after checking for emptiness. While this implementation is syntactically valid, it permits semantically invalid inputs—such as negative ages or values far outside a realistic human range—and silently defaults to returning 0.0 when the input list is empty. In contrast, the enhanced version defines a sealed trait `Age` and a case class `ValidAge`, with a smart constructor in the `Age` companion object that en-

(a) Baseline

```scala
object GeneratedFunctions {
    def averageAge(ages: List[Int]): Double = {
        if (ages.isEmpty) 0.0
        else ages.sum.toDouble / ages.length
    }
}
```

(b) TypePilot

```scala
object GeneratedFunctions {
  sealed trait Age {
    def value: Int
  }

  final case class ValidAge(value: Int) extends Age

  object Age {
    def apply(value: Int): Option[Age] = {
      if (value >= 0 && value <= 120) Some(ValidAge(value))
      else None
    }
  }

  def averageAge(ages: List[Age]): Option[Double] = {
    if (ages.isEmpty) None
    else {
        val (sum, count) = ages.foldLeft((0L, 0)) {
        case ((accSum, accCount), age) =>
         (accSum + age.value, accCount + 1)
    }
  }
}
```

Figure 5: Comparison of baseline and TypePilot average age function generations from `Qwen-2.5-Coder (32B)`

forces domain-specific constraints: only values between 0 and 120 are permitted. The `averageAge` function now accepts a list of validated `Age` values and returns an `Option[Double]`, making both the domain invariants and the possibility of undefined results (e.g., empty lists) explicit at the type level. This design ensures that all inputs have been prevalidated before the function executes, reducing the likelihood of subtle logic bugs and enabling safer composition in larger systems. A second example related to generating a function to search for files using bash is discussed in appendix C.

In TypePilot, the Scala type system is used not merely to enforce syntactic correctness but to encode domain abstractions rules, constrain behavior, and make failure modes explicit. By doing so, it transforms what would otherwise be runtime checks and ad hoc validations into statically enforced contracts. This shift leads to code that is more robust, more predictable, and better aligned with the principles of secure and maintainable software design. In the context of LLM-generated code, these benefits are particularly important, as they offer a principled way to guard against common pitfalls and encourage safer defaults during generation.

4.3 Vulnerability Analysis

We performed a post-hoc vulnerability analysis by categorizing the vulnerabilities observed in each test case. These categories include input constraint issues (shape violations, null dereferences and boundary violations) and code injection risks (HTML injection, bash injection and path traversal). For each method, we calculated the fraction of secure outputs and averaged the results across the three LLMs, which is displayed in Figure 6.

For input constraints, robust prompting offered limited improvements over the baseline, particularly for shape violations and null dereferences. It often inserted assertions but did not systematically enforce data structure correctness. TypePilot reduced these errors more effectively, as the presence of type specifications led the models to generate code structured around expected data formats rather than relying on runtime checks.

For code injection, TypePilot also lowered vulnerability rates, especially for bash injections where robust prompting typically altered command structure without validating input. Results varied between models: Qwen-2.5-Coder (32B) and Deepseek-Coder (33B) generally applied the type system consistently, while CodeLlama (70B) sometimes attempted to handle vulnerabilities outside the type framework. In some cases, type constraints were only partially used, such as defining a type for an output value but not for the input values.

Appendix D analyzes attention weights across the three methods, showing that TypePilot places greater emphasis on key safety terms during code generation than robust prompting.

4.4 Comparison to Self-Planning Code Generation

As an additional validation, we compared TypePilot to the Self-Planning prompting framework,

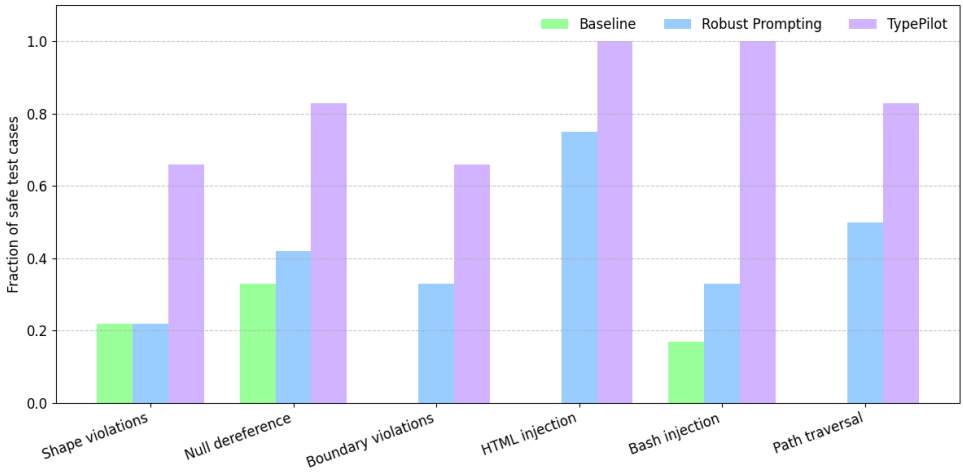

Figure 6: Fraction of secure code generations across vulnerability categories for each of the methods (baseline prompting, robust prompting, TypePilot). Results are averaged over all evaluated LLMs. Lower bars indicate a higher frequency of vulnerabilities; higher bars indicate safer generations.

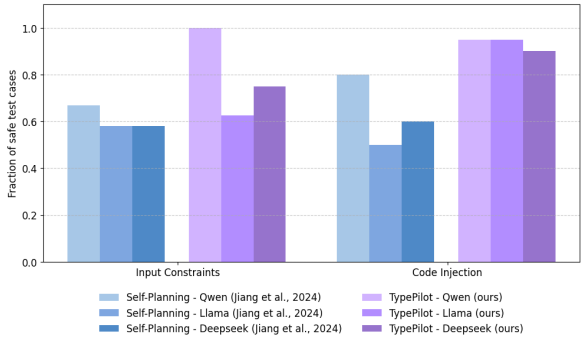

Figure 7: Comparison of the secure code generation methods TypePilot (ours) and Self-Planning, as introduced by Jiang et al. (2024).

as discussed in Section 3.4. In the Self-Planning framework, the model is first asked to outline a plan for solving the task. Afterwards, it is asked to write the code by executing the plan, and it is explicitly instructed to consider safety and security aspects before writing code. Overall, TypePilot outperforms self-planning for both the input constraint and code injection tasks. The difference is largest for Qwen-2.5-Coder (32B), which more reliably adheres to the type system instructions in TypePilot, resulting in fewer shape and null-handling issues compared to the Self-Planning setup.

Manual inspection of the Self-Planning outputs reveals that, despite explicit prompts to account for vulnerabilities during the planning and implementation stages, models frequently overlook or underaddress these concerns. The generated plans may mention security considerations in abstract terms but rarely translate them into concrete, protective measures in the final code. These findings suggest that simply instructing the model to "think about safety" is insufficient: introducing a structured intermediate step, such as TypePilot's type-enforced specification phase, is more effective in steering the model toward safer code generation.

5 Scaling

The primary goal of this work is to show that leveraging the type system in strongly typed languages can substantially mitigate vulnerabilities in LLM-generated code. While our experiments focus on relatively simple test cases, practical applications often involve larger, interconnected codebases with complex object hierarchies. Scaling our framework to such scenarios presents new challenges, primarily related to context management and dependency reasoning across multiple files and modules.

One promising direction is the development of a hybrid, object-aware prompting system. In this approach, metadata about each relevant object, including its types and invariants, is provided to the LLM prior to generation. This structured context could enable the model to reason more accurately about type interactions and enforce security constraints across function boundaries. Additionally, integrating lightweight symbolic reasoning or type inference engines could help LLMs maintain global consistency in larger projects, further reducing the risk of injection attacks and logical errors.

6 Conclusion

In this work, we aim to improve the security of LLM generated mission-critical code, focusing on the Scala strongly typed language. As Scala is routinely used in mission-critical software and engineers are increasingly often using LLMs to code, it is essential to ensure that the generated code is free of vulnerabilities. We first show that LLMs are not able to autonomously use the static verification tool Stainless. Therefore, we develop a more general agentic AI framework that structures multi-step interactions between LLMs for code generation. By leveraging the Scala type system, we significantly improve the quality and safety of generated code. Crucially, this approach transforms type systems from passive compile-time enforcers into active agents of code safety. We study two different classes of vulnerabilities, input constraints and code injection, and show that in both cases our framework improves code safety over a baseline and zero-shot robust prompting setting. We use the rigidity of the Scala type system to compensate for the inconsistencies picked up from the training code by LLMs, which in turn allow an easier interface to access the power of the Scala type system.

We conclude by suggesting two directions for future research. First, future work should test the framework's capabilities in more complex codebases. While this study provided a proof of concept using simple test cases, real-world software tends to be more complex, so validating our approach in these environments is important to assess its effectiveness. Second, deploying the framework in an active development setting would allow engineers to use it in their daily work and provide valuable feedback. This real-world input can guide further improvements and help tailor the framework to better meet the needs of software teams.

References

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. Evaluating large language models trained on code.

Shih-Chieh Dai, Jun Xu, and Guanhong Tao. 2025. A comprehensive study of llm secure code generation.

Wen Fan, Marilyn Rego, Xin Hu, Sanya Dod, Zhaorui Ni, Danning Xie, Jenna DiVincenzo, and Lin Tan. 2025. Evaluating the ability of large language models to generate verifiable specifications in verifast.

Ellie Gabriel, Xenophon Papademetris, Ayesha N. Quraishi, and Gregory P. Licholai. 2022. *Therac-25: Software that Killed*, page 263–267. Cambridge University Press.

Ajay Harish. 2025. When nasa lost a spacecraft due to a metric math mistake. Accessed: July 8, 2025.

Jingxuan He and Martin Vechev. 2023. Large language models for code: Security hardening and adversarial testing. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, CCS '23, page 1865–1879. ACM.

Jingxuan He, Mark Vero, Gabriela Krasnopolska, and Martin Vechev. 2024. Instruction tuning for secure code generation.

Xue Jiang, Yihong Dong, Lecheng Wang, Zheng Fang, Qiwei Shang, Ge Li, Zhi Jin, and Wenpin Jiao. 2024. Self-planning code generation with large language models. *ACM Trans. Softw. Eng. Methodol.*, 33(7).

Raphaël Khoury, Anderson R. Avila, Jacob Brunelle, and Baba Mamadou Camara. 2023. How secure is code generated by chatgpt? In *2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 2445–2451.

Ummay Kulsum, Haotian Zhu, Bowen Xu, and Marcelo d'Amorim. 2024. A case study of llm for automated vulnerability repair: Assessing impact of reasoning and patch validation feedback. In *Proceedings of the 1st ACM International Conference on AI-Powered Software*, AIware 2024, page 103–111, New York, NY, USA. Association for Computing Machinery.

Mayank Kumar, Jiaqi Xue, Mengxin Zheng, and Qian Lou. 2025. Tfhe-coder: Evaluating llm-agentic fully homomorphic encryption code generation.

Lab for Automated Reasoning and Analysis. 2025. Stainless: A verification framework for scala programs. https://epfl-lara.github.io/stainless/index.html.

Mahmoud Nazzal, Issa Khalil, Abdallah Khreishah, and NhatHai Phan. 2024. Promsec: Prompt optimization for secure generation of functional source code with large language models (llms). In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, CCS '24, page 2266–2280. ACM.

Stephen O'Grady. 2025. The redmonk programming language rankings: January 2025. Accessed: July 8, 2025.

Hammond Pearce, Baleegh Ahmad, Benjamin Tan, Brendan Dolan-Gavitt, and Ramesh Karri. 2025. Asleep at the keyboard? assessing the security of github copilot's code contributions. *Commun. ACM*, 68(2):96–105.

Hammond Pearce, Benjamin Tan, Baleegh Ahmad, Ramesh Karri, and Brendan Dolan-Gavitt. 2022. Examining zero-shot vulnerability repair with large language models.

reuters. 2023. Explainer: Why u.s. flights were grounded by a faa system outage. Accessed: July 8, 2025.

Mark Vero, Niels Mündler, Victor Chibotaru, Veselin Raychev, Maximilian Baader, Nikola Jovanović, Jingxuan He, and Martin Vechev. 2025. Baxbench: Can llms generate correct and secure backends?

Haoran Wang, Zhenyu Hou, Yao Wei, Jie Tang, and Yuxiao Dong. 2025. Swe-dev: Building software engineering agents with training and inference scaling.

Jianxun Wang and Yixiang Chen. 2023. A review on code generation with llms: Application and evaluation. In *2023 IEEE International Conference on Medical Artificial Intelligence (MedAI)*, pages 284–289.

Author Index