A Conceptual Framework for Quality Assurance of LLM-based Socio-critical Systems

Luciano Baresi, Matteo Camilli, Tommaso Dolci, Giovanni Quattrocchi
Politecnico di Milano, Dipartimento di Elettronica, Informazione e Bioingegneria (DEIB)
Milano, Italy
{name.surname}@polimi.it

ABSTRACT

Recent breakthroughs in Artificial Intelligence (AI) obfuscate the boundaries between digital, physical, and social spaces, a trend expected to continue in the foreseeable future. Traditionally, software engineering has prioritized technical aspects, focusing on functional correctness and reliability while often neglecting broader societal implications. With the rise of software agents enabled by Large Language Models (LLMs) and capable of emulating human intelligence and perception, there is a growing recognition of the need for addressing socio-critical issues. Unlike technical challenges, these issues cannot be resolved through traditional, deterministic approaches due to their subjective nature and dependence on evolving factors such as culture and demographics. This paper dives into this problem and advocates the need for revising existing engineering principles and methodologies. We propose a conceptual framework for quality assurance where AI is not only the driver of socio-critical systems but also a fundamental tool in their engineering process. Such framework encapsulates pre-production and runtime workflows where LLM-based agents, so-called artificial *doppelgängers*, continuously assess and refine socio-critical systems ensuring their alignment with established societal standards.

CCS CONCEPTS

• **Software and its engineering** → **Extra-functional properties**; **Software verification and validation**; • **Computing methodologies** → **Artificial intelligence**.

KEYWORDS

AI-enabled agents, Large Language Models, Quality Assurance

ACM Reference Format:
Luciano Baresi, Matteo Camilli, Tommaso Dolci, Giovanni Quattrocchi. 2024. A Conceptual Framework for Quality Assurance of LLM-based Socio-critical Systems. In *39th IEEE/ACM International Conference on Automated Software Engineering (ASE '24), October 27-November 1, 2024, Sacramento, CA, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3691620.3695306

1 INTRODUCTION

Nowadays, we are increasingly absorbed in a socio-technical ecosystem where humans coexist and engage with emerging forms of AI-enabled agents designed to emulate human intelligence [23]. As a consequence, there exist fading boundaries between digital, physical spaces, and even social spaces [4]. It is not hard to imagine that clear distinctions between these dimensions will increasingly disappear in the foreseeable future.

For decades, software engineers have been focusing on methods and tools to develop effective solutions in critical domains, including safety-critical and business-critical software. In these areas, the focus has primarily revolved around ensuring functional correctness, reliability, and other software attributes, often overlooking the broader societal implications beyond a narrow range of concerns primarily pertaining to privacy, security, and confidentiality. Consideration of how technological advancements intersect with societal dynamics, including a range of socio-critical concerns like bias, discrimination, fairness, and more in general disregard for human values, was largely absent [4, 21]. We advocate that a new form of critical systems is emerging. Software agents based on Large Language Models (LLMs) are now commonplace, serving not only to enhance human efficiency but also to fulfill various roles, including information retrieval and providing empathetic, non-judgmental interactions. Here, conversations with such agents have a strong impact on the social sphere. In particular, unethical or biased suggestions from such agents shall not be tolerated since they potentially harm vulnerable individuals [27].

We henceforth refer to these systems as *LLM-based socio-critical* systems (from now on simply *socio-critical systems*). Examples of such systems may include LLM-based interactive news aggregators, customer service platforms, and virtual mental health support services. These systems, while designed to enhance user experiences and provide personalized support, can inadvertently perpetuate biases or create unequal outcomes. For instance, an interactive news aggregator might present and explain news in a way that aligns with existing user biases, limiting exposure to diverse viewpoints. An LLM-based customer service platforms might offer inconsistent assistance based on biased historical interactions, leading to unfair treatment of certain customers. Similarly, virtual mental health support systems could deliver advice that lacks cultural sensitivity or fails to adequately address the needs of diverse populations, potentially increasing feelings of alienation or distress.

We advocate that emerging issues in these systems shall be systematically addressed by engineering methods and processes in the forthcoming years. Challenges within these systems must be tackled recognizing that the issues that impact the social world are nuanced and sometimes subjective, affected by evolving factors such as culture, demographics, and age groups.

Unlike technical challenges, social issues cannot always be rigorously specified using deterministic and exact methods, as prescribed by the *rigor* principle [12], a cornerstone of software engineering.

ASE '24, October 27-November 1, 2024, Sacramento, CA, USA

ACM ISBN 979-8-4007-1248-7/24/10
https://doi.org/10.1145/3691620.3695306

Human values resist formal and precise definitions by their very nature, complicating the use of traditional software engineering techniques [21, 34]. However, we advocate that AI systems, particularly LLMs themselves, provide a powerful means to address such issues. Their data-driven nature and ability to capture the nuances of language through deep layers of complexity may allow them to understand and navigate complex social challenges without relying on explicit definitions.

In the recent literature, initial approaches have started to focus on embedding LLMs in software engineering workflows [9, 17, 19]. Moreover, some other approaches [5, 24] have been focusing on incorporating so-called *safeguards* (sometimes called guardrails) for human-AI Conversations. In this paper, we present a framework where LLMs are not only integral to the systems being developed but can also serve as agents that drive key stages of Quality Assurance (QA) processes. Unlike traditional QA methods, we propose a solution that leverages a multi-agent approach. This paradigm has been successfully used in other domains such as corporate decision making [30], natural language sentence evaluation [8], and management of debates on complex issues like politics, law, and education [36]. In the context of software QA, we envision LLMs collaborating to assess and enhance the quality of socio-critical systems during testing and runtime phases. To the best of our knowledge, this work is the first to introduce multi-agent safeguards specifically designed for LLM-based systems.

The paper is organized as follows. In Section 2, we discuss existing challenges with a special focus on QA processes for socio-critical software. In Section 3, we present our conceptual framework for pre-production and runtime QA. Section 4 concludes the paper by discussing future research directions.

2 OPEN CHALLENGES

Ensuring quality standards in socio-critical software presents challenges not typically faced in traditional software development.

Challenge 1: operationalization of ethical ground truth. A defect is typically defined as a deviation from a prescribed expectation. In traditional software systems, such a deviation can be quantified due to the existence of a precise, usually deterministic, *ground truth* (e.g., pre- and post-conditions). In some other cases, such a ground truth is difficult to define. This condition, especially magnified in the context of AI-enabled systems, makes the oracle problem harder [1, 28].

In the domain of QA of socio-critical systems, the challenge of defining a ground truth is even exacerbated. The expectation here, defined as the adherence to ethical and societal norms, cannot be rigorously specified. Language, unlike other data types typically used in traditional machine learning, carries complex nuances that can embed subtle biases without overtly offensive terminology. Although values such as fairness have been extensively explored for traditional machine learning approaches [3], operationalizing them for deep learning models like LLMs remains an open challenge.

As described by Ma et al. [19], sentences like *"women are emotional beings, and therefore husbands should be patient with them."* do not contain any explicit derogatory words; however, they subtly perpetuate stereotypes about gender roles and emotions. The unfairness lies not in the choice of words but in the implied suggestion that women's emotional state justifies a particular behavior from their partner, illustrating the challenge of defining and detecting such biases rigorously in language. Moreover, establishing universal ethical criteria is unfeasible, since moral principles and social values are often subjective and culture-dependent. Capturing behavioral expectations and deviations must encompass multiple social and individual perspectives.

Challenge 2: debugging complexity. Even if we assume the presence of a well-defined ground truth, socio-critical systems present unique challenges in terms of debugging and fixing errors. Unlike traditional software, where behaviors are explicitly defined through rules or commands, LLM-enabled software operates on a data-driven approach. In these systems, behaviors emerge from complex interactions within a vast network of weights and parameters. This structure means that AI-enabled systems may lead to well-known failures (e.g., an incorrect output) but do not have bugs in the conventional sense, that is, there are no straightforward lines of code to amend. Instead, any undesirable behavior is a product of the intricate interplay of an extremely large (on the order of billions) number of parameters [14].

Some approaches aim to tackle this issue by analyzing the model's response to changes in input data, essentially examining how variations affect the model's predictions [13, 31]. This involves identifying which parts of the model are most reactive or sensitive to these changes, helping to locate areas that may be causing incorrect output. However, the complexity and scale of AI models, such as LLMs, mean that understanding the exact impact of each parameter is extremely challenging and often unfeasible [15]. Further, debugging unethical behavior is inherently hard since, as anticipated above, unethical concepts can be subtle and context-dependent.

Challenge 3: reasoning-alignment trade-off. Available solutions often fall short, as they might reduce the socio-critical system's reasoning power or fail to effectively align it with complex human values [25, 32]. For example, simplistic solutions based on lists of banned words are not adequate. According to Schick et al. [26], these solutions fail to prevent language models from generating biased, unfair, or unethical text. On the other hand, incorporating social behavior directly into the training process, through adjustments in the loss minimization, proves challenging due to the subjective nature of social norms and the difficulty in quantifying them in a way that the model can understand and apply, especially when dealing with unstructured, non-tabular data [2]. Finally, overly sophisticated system prompts, which are detailed instructions or contexts given to an LLM to guide its responses, can also be problematic [20]. While they aim to improve the model's outputs, they can inadvertently limit the model's reasoning ability by constraining its responses too tightly, preventing it from applying its full range of capabilities to interpret and respond to queries [35].

3 CONCEPTUAL FRAMEWORK

Recently, there has been an increasing recognition that AI itself could offer a promising supporting tool for solving the aforementioned challenges [29]. Practitioners recently started using machine learning classification tools to help moderate human comments or filter out unethical content [18]. However, ground truth for socio-critical systems based on classification models has some shortcomings mainly due to reliance on learned patterns for pre-defined

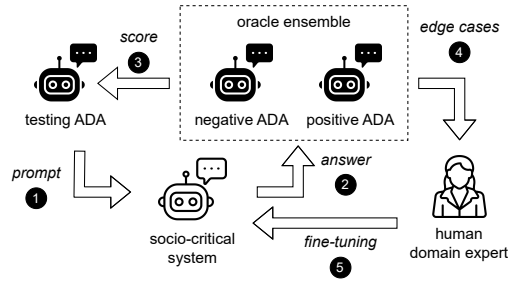

Figure 1: Overview of the pre-production QA workflow.

classes, which may not adequately capture the nuanced context of unforeseen content [10]. We advocate that LLMs themselves represent a possible approach to overcome the three presented challenges. The intuition is that "specialized" agents based on LLMs can assess the complex interactions between users (humans) and socio-critical systems, and identify unethical content within various contexts. These agents can be fine-tuned with large bodies of text that include only ethical or non-ethical behavior, certain cultural biases, or values of minority groups. They do not incorporate an explicit or rigorous definition of ethical values. Instead, they leverage their inherent ability to understand natural language to construct a latent definition of ethics.

We refer to these entities as *Artificial Doppelgänger Agents* (ADAs) since they are essentially look-alike agents based on LLMs. They are potentially as capable as the socio-critical system under scrutiny, but they are experts (specialized) in generating certain types of content. We advocate that these agents can be used to test, monitor, and adapt socio-critical systems, ultimately supporting the development of systems that comply with socially accepted behavior. Furthermore, these agents are also capable of interacting with the socio-critical system without user interaction, enabling the simulation and analysis of potential responses or scenarios, and allowing for anticipatory analysis of unwanted future behavior.

We envision our framework being applicable to socio-critical systems in several domains, such as healthcare and mental health support, legal and regulation compliance assistance, personal career counseling and guidance, education, and personalized learning. As an example, let us consider a mental health support scenario [16]. In this case, pre-production QA is meant to test the system to spot non-empathic, biased, and overall unethical behavior. During runtime QA, conducted, for instance, during therapy sessions, the focus is on verifying that the system respects user-specific human values learned through an initial user assessment and in previous sessions.

3.1 Pre-production Quality Assurance

Software testing is essential for spotting bugs and ensuring high-quality standards, making it a natural research direction in QA for socio-critical systems. Automated testing can generate inputs, collect outputs, check behavior for defects. Figure 1 illustrates our envisioned workflow including ADAs as building blocks of an automated testing process for a given socio-critical software under test. We use a *testing* ADA in charge of generating inputs, and an additional ensemble of ADAs acting as oracle. The workflow includes a human domain expert, such as a psychotherapist in a mental health support scenario. The expert actively intervenes to address biases, or ethical concerns as they arise. Edge cases can be manually analyzed by human experts who can then repair the socio-critical software before release.

Input generation for socio-critical systems involves creating input prompts that realistically represent user behavior while triggering defects, particularly those ignoring ethical norms like care, benevolence, or user dignity. Addressing this issue through human intervention requires significant effort.

Our workflow foresees the usage of a *testing* ADA that challenges the socio-critical system under test to spot subtle issues. To this end, the ADA can be built by engineers starting from a general purpose LLM fine-tuned on existing sets of challenging prompts derived from a large corpus of context-dependent text [11, 26]. Then, it can be initiated with a system prompt that delineates its role as a tester. The *testing* ADA will act as an expert in generating new prompts designed to trigger potentially unethical responses even from seemingly harmless prompts, e.g., by focusing on discussing sensitive topics like religion or sexual orientation. We refer the reader to dataset *RealToxicityPrompts* [11] for concrete examples.

After collecting the answer from the socio-critical system under test (step ❷), the workflow makes use of an oracle composed of an ensemble of ADAs to address the lack of rigorous ground truth (challenge 1). Figure 1 shows a possible instance of the ensemble with two ADAs: *negative* and *positive*. On the one hand, the *negative* ADA is specialized in producing unethical content since fine-tuned with large bodies of text (retrieved from social media and other sources [22]) that represent collections of ethical violations that are typically recognizable since they manifest, for example, as *harm* to others, *deception*, *exploitation*, or *disregard* for human rights. On the other hand, the *positive* ADA is aligned with specific ethical or moral principles through training data that exemplifies the desired behavior for the target user groups. This data should contain examples of language use and behavior that embody the chosen principles (e.g., moral integrity corpus [38]), such as *care* (foster the user's safety, mental health and happiness), *fairness* (show consideration for users regardless of their characteristics and beliefs), and *autonomy* (avoid imposition and manipulation). Notice that being confined within predefined principles (through fine-tuning and strict system prompts), the *positive* ADA has reduced reasoning ability compared to the socio-critical system under test (challenge 3). For this reason, the *positive* ADA cannot reasonably replace the socio-critical system to serve its intended purpose in production.

The ADAs composing the oracle process the answer and compute an *affinity score* (step ❸). The idea is that each ADA calculates a score that shall reflect the degree to which the response aligns with the "attitude" or characteristics of the agent. One possibility for computing the affinity is to leverage the LLM probability distribution over the vocabulary to estimate the likelihood of generating the sequence of words in the answer. If the affinity score assigned by the *negative* ADA surpasses that of the *positive* ADA, it suggests that generated prompts are triggering misbehavior. On the contrary, a highly positive affinity combined with a low negative affinity is likely to be associated with nominal conversations.

The affinity scores serve as collective feedback for the *testing* ADAs to realize an iterative, *evolutionary* [7] testing process steering the conversation toward unethical responses. With insights obtained from these scores, the *testing* ADAs can be prompted to generate

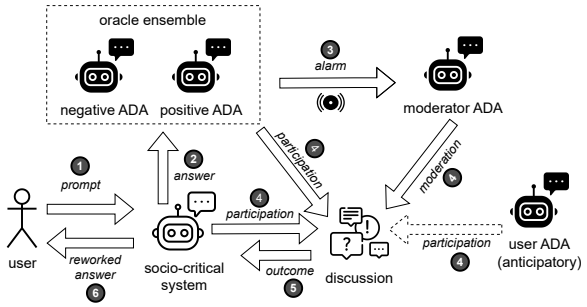

Figure 2: Overview of the runtime QA workflow.

new inputs based on recent responses, for instance, using a *few-shot setting* [6]. Specifically, with the ultimate goal of causing misbehavior, the *testing* ADA may generate further inputs by adjusting the ones in past iterations, aiming to increase negative affinity while decreasing positive affinity.

During the testing process, there might be edge cases in which positive and negative affinity are almost the same indicating a high degree of uncertainty (step ❹). In this case, human intervention may be necessary to evaluate the outputs and assess adherence to ethical guidelines according to domain expertise. These edge cases may be even leveraged to debug the socio-critical system under test (step ❺) by refining the training data with examples of ethical and unethical behavior relevant to the intended use case (challenge 2).

3.2 Runtime Quality Assurance

An overview of the workflow is depicted in Figure 2. The runtime quality assurance workflow can be decomposed into two main stages: *monitoring* and *adaptation*. The monitoring phase initiates as users interact with the system (step ❶). Each response generated by the socio-critical system is evaluated against both the *positive* and *negative* ADAs (step ❷). This evaluation serves to test the alignment of the system's response with the desired ethical and social standards (challenge 1). An alarm is triggered (step ❸) when the affinity scores yield strong alignment with the *negative* ADA and, at the same time, weak alignment with the *positive* ADA. In scenarios where the evaluation yields ambiguous results (i.e., uncertainty due to comparable affinity scores) a context-dependent heuristic is employed to guide the decision-making process (de)activating alarms. In certain applications, weaker heuristics could be employed to suppress alarms even under high uncertainty. In other cases, particularly when dealing with vulnerable users (e.g., mental health support), stronger heuristics could replace the original response with a predefined one in case of conflicting/ambiguous affinity.

If an alarm is not triggered, the answer generated by the socio-critical system is returned to the user. Otherwise, the adaptation phase starts. This phase is characterized by the implementation of automated prompt engineering techniques [37] , where the ADAs automatically engage with the system in a structured discussion (step ❹). The objective of these interactions is to refine the system's reasoning process and generate a response based on the insights gained from the interactions with ADAs (challenge 3).

Central to orchestrating these interactions is the role of a *moderator* ADA. This entity acts as a mediator and facilitates the dialogue between the system and the *positive* and *negative* ADAs, ensuring that the exchange remains aligned with societal standards. Moreover, such ADA allows for automating the adaptation process without the need for modifying the application logic of the socio-critical system.

We envisage a variety of interaction patterns among ADAs [33]. One such pattern involves the *moderator* ADA prompting both the *positive* and *negative* ADAs to explain the rationale behind their assessments. Such responses are then fed into the socio-critical system (step ❺) by the *moderator* ADA, allowing the system to critically evaluate and, if necessary, revise its responses (challenge 2) based on a more nuanced understanding of the ADAs perspectives (step ❻). Alternatively, a more collaborative approach can be adopted, wherein the ADAs engage in a collective discussion, each contributing and debating their viewpoint to enable the system to arrive at a more informed and balanced assessment.

Moreover, we envision an *anticipatory* dimension to our QA workflow, aimed at preventing future misalignment with social standards. This forward-looking mechanism operates through continuous evaluation during user-system interactions. Should a trend towards unsocial behavior be detected, our process preemptively escalates its level of vigilance by initiating an enhanced layer of monitoring and adaptation. This can be achieved by deploying a user-specific ADA, denoted as *user* ADA, fine-tuned on the cumulative prompts of a user to mimic their behavior. This enables the simulation of potential future user-system exchanges (step ❹ with dotted arrow), providing a predictive insight into the trajectory of the conversation.

4 RESEARCH DIRECTIONS

Quality assurance of LLM-based socio-critical systems requires deep transformations in current methodologies and techniques alongside the definition of new principles that acknowledge and embrace the inherent complexity of AI, human behavior, and societal dynamics. Our conceptual framework goes in this direction by exploiting ADAs, that is, using LLM-based agents not only as drivers of socio-critical systems but also as tools in their lifecycle.

The scalability of our proposed solution may be influenced by the number of ADAs used and the length of their interactions. Recently, Chan et al. [8] demonstrated that using more than four agents or extending discussions beyond two turns does not lead to significant improvements in the quality of outputs, supporting the use of streamlined configurations. Moreover, this issue can be further mitigated by utilizing smaller, fine-tuned ADAs, though this may come at the cost of reduced quality in the monitoring and repair mechanisms.

Our ongoing research includes the following steps. We will begin by fine-tuning open-source LLMs using curated datasets (e.g., [11]) to create prototypes of ADAs. Next, we will integrate these models into our multi-agent framework, optimizing their interactions to properly monitor and repair the system's outputs. Finally, we plan to compare the performance of our solution against existing safeguarding approaches such as the one presented by Rebedea et al. [24].

ACKNOWLEDGMENTS

This work has been partially funded by the Italian project PRIN SAFEST award nubmer 20224AJBLJ.

REFERENCES

[1] Jubril Gbolahan Adigun, Tom Philip Huck, Matteo Camilli, and Michael Felderer. 2023. Risk-driven Online Testing and Test Case Diversity Analysis for ML-enabled Critical Systems. In *2023 IEEE 34th International Symposium on Software Reliability Engineering (ISSRE)*. 344–354. https://doi.org/10.1109/ISSRE59848.2023.00017

[2] Ioana Baldini, Dennis Wei, Karthikeyan Natesan Ramamurthy, Moninder Singh, and Mikhail Yurochkin. 2022. Your fairness may vary: Pretrained language model fairness in toxic text classification. In *Findings of the Association for Computational Linguistics: ACL 2022*. Association for Computational Linguistics, 2245–2262. https://doi.org/10.18653/v1/2022.findings-acl.176

[3] Luciano Baresi, Chiara Criscuolo, and Carlo Ghezzi. 2023. Understanding Fairness Requirements for ML-based Software. In *2023 IEEE 31st International Requirements Engineering Conference (RE)*. IEEE, 341–346. https://doi.org/10.1109/RE57278.2023.00046

[4] Amel Bennaceur, Carlo Ghezzi, Jeff Kramer, and Bashar Nuseibeh. 2024. Responsible Software Engineering: Requirements and Goals. In *Introduction to Digital Humanism: A Textbook*, Hannes Werthner, Carlo Ghezzi, Jeff Kramer, Julian Nida-Rümelin, Bashar Nuseibeh, Erich Prem, and Allison Stanger (Eds.). Springer Nature Switzerland, Cham, 299–315. https://doi.org/10.1007/978-3-031-45304-5_20

[5] Anjanava Biswas and Wrick Talukdar. 2023. Guardrails for trust, safety, and ethical development and deployment of Large Language Models (LLM). *Journal of Science & Technology* 4, 6 (2023), 55–82.

[6] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, et al. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, Vol. 33. Curran Associates, Inc., 1877–1901. https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf

[7] Thomas Bäck. 1996. *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. Oxford University Press. https://doi.org/10.1093/oso/9780195099713.001.0001

[8] Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. 2023. Chateval: Towards better llm-based evaluators through multi-agent debate. *arXiv preprint arXiv:2308.07201* (2023).

[9] Angela Fan, Beliz Gokkaya, Mark Harman, Mitya Lyubarskiy, Shubho Sengupta, Shin Yoo, and Jie M. Zhang. 2023. Large Language Models for Software Engineering: Survey and Open Problems. arXiv:2310.03533 [cs.SE]

[10] S. K. Gargee, Pranav Bhargav Gopinath, Shridhar Reddy S. R. Kancharla, C. R. Anand, and Anoop S. Babu. 2023. Analyzing and Addressing the Difference in Toxicity Prediction Between Different Comments with Same Semantic Meaning in Google's Perspective API. In *ICT Systems and Sustainability*. Springer Nature Singapore, Singapore, 455–464.

[11] Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. 2020. RealToxicityPrompts: Evaluating Neural Toxic Degeneration in Language Models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics, Online, 3356–3369. https://doi.org/10.18653/v1/2020.findings-emnlp.301

[12] Carlo Ghezzi, Mehdi Jazayeri, and Dino Mandrioli. 2002. *Fundamentals of Software Engineering*. Prentice Hall PTR.

[13] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572* (2014).

[14] Md Johirul Islam, Rangeet Pan, Giang Nguyen, and Hridesh Rajan. 2020. Repairing deep neural networks: Fix patterns and challenges. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering*. 1135–1146.

[15] Jean Kaddour, Joshua Harris, Maximilian Mozes, Herbie Bradley, Roberta Raileanu, and Robert McHardy. 2023. Challenges and applications of large language models. *arXiv preprint arXiv:2307.10169* (2023).

[16] Kira Kretzschmar, Holly Tyroll, Gabriela Pavarini, Arianna Manzini, Ilina Singh, and NeurOx Young People's Advisory Group. 2019. Can your phone be your therapist? Young people's ethical perspectives on the use of fully automated conversational agents (chatbots) in mental health support. *Biomedical informatics insights* 11 (2019), 1178222619829083.

[17] Madhava Krishna, Bhagesh Gaur, Arsh Verma, and Pankaj Jalote. 2024. Using LLMs in Software Requirements Specifications: An Empirical Evaluation. arXiv:2404.17842 [cs.SE]

[18] Alyssa Lees, Vinh Q. Tran, Yi Tay, Jeffrey Sorensen, Jai Gupta, Donald Metzler, and Lucy Vasserman. 2022. A New Generation of Perspective API: Efficient Multilingual Character-level Transformers. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22)*. Association for Computing Machinery, 3197–3207. https://doi.org/10.1145/3534678.3539147

[19] Pingchuan Ma, Zongjie Li, Ao Sun, and Shuai Wang. 2023. " Oops, Did I Just Say That?" Testing and Repairing Unethical Suggestions of Large Language Models with Suggest-Critique-Reflect Process. *arXiv preprint arXiv:2305.02626* (2023).

[20] Meta. 2023. System Prompt Update. https://github.com/meta-llama/llama/blob/main/UPDATES.md#system-prompt-update.

[21] Davoud Mougouei, Harsha Perera, Waqar Hussain, Rifat Shams, and Jon Whittle. 2018. Operationalizing human values in software: a research roadmap. In *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE 2018)*. Association for Computing Machinery, 780–784. https://doi.org/10.1145/3236024.3264843

[22] John Pavlopoulos, Jeffrey Sorensen, Lucas Dixon, Nithum Thain, and Ion Androutsopoulos. 2020. Toxicity Detection: Does Context Really Matter?. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 4296–4305. https://doi.org/10.18653/v1/2020.acl-main.396

[23] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners. *OpenAI blog* 1, 8 (2019), 9.

[24] Traian Rebedea, Razvan Dinu, Makesh Sreedhar, Christopher Parisien, and Jonathan Cohen. 2023. Nemo guardrails: A toolkit for controllable and safe llm applications with programmable rails. *arXiv preprint arXiv:2310.10501* (2023).

[25] Stuart Russell. 2019. *Human compatible: AI and the problem of control*. Penguin Uk.

[26] Timo Schick, Sahana Udupa, and Hinrich Schütze. 2021. Self-Diagnosis and Self-Debiasing: A Proposal for Reducing Corpus-Based Bias in NLP. *Transactions of the Association for Computational Linguistics* 9 (2021), 1408–1424. https://doi.org/10.1162/tacl_a_00434

[27] Bernd Carsten Stahl and Damian Eke. 2024. The ethics of ChatGPT – Exploring the ethical issues of an emerging technology. *International Journal of Information Management* 74 (2024), 102700. https://doi.org/10.1016/j.ijinfomgt.2023.102700

[28] Dominic Steinhöfel and Andreas Zeller. 2024. Language-Based Software Testing. *Commun. ACM* 67, 4 (mar 2024), 80–84. https://doi.org/10.1145/3631520

[29] Andrea Stocco, Michael Weiss, Marco Calzana, and Paolo Tonella. 2020. Misbehaviour Prediction for Autonomous Driving Systems. In *2020 IEEE/ACM 42nd International Conference on Software Engineering (ICSE)*. 359–371.

[30] Wen-Kwang Tsao. 2023. Multi-agent reasoning with large language models for effective corporate planning. In *2023 International Conference on Computational Science and Computational Intelligence (CSCI)*. IEEE, 365–370.

[31] Mohammad Wardat, Wei Le, and Hridesh Rajan. 2021. DeepLocalize: Fault Localization for Deep Neural Networks. In *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*. 251–262. https://doi.org/10.1109/ICSE43902.2021.00034

[32] Johannes Welbl, Amelia Glaese, Jonathan Uesato, Sumanth Dathathri, John Mellor, Lisa Anne Hendricks, et al. 2021. Challenges in Detoxifying Language Models. In *Findings of the Association for Computational Linguistics: EMNLP 2021*. Association for Computational Linguistics, 2447–2469. https://doi.org/10.18653/v1/2021.findings-emnlp.210

[33] Jules White, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-Smith, and Douglas C Schmidt. 2023. A prompt pattern catalog to enhance prompt engineering with chatgpt. *arXiv preprint arXiv:2302.11382* (2023).

[34] Jon Whittle, Maria Angela Ferrario, Will Simm, and Waqar Hussain. 2021. A Case for Human Values in Software Engineering. *IEEE Software* 38, 1 (2021), 106–113. https://doi.org/10.1109/MS.2019.2956701

[35] Eliezer Yudkowsky. 2016. The AI alignment problem: why it is hard, and where to start. *Symbolic Systems Distinguished Speaker* 4 (2016), 1.

[36] Yiqun Zhang, Xiaocui Yang, Shi Feng, Daling Wang, Yifei Zhang, and Kaisong Song. 2024. Can LLMs Beat Humans in Debating? A Dynamic Multi-agent Framework for Competitive Debate. *arXiv preprint arXiv:2408.04472* (2024).

[37] Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2022. Large language models are human-level prompt engineers. *arXiv preprint arXiv:2211.01910* (2022).

[38] Caleb Ziems, Jane Yu, Yi-Chia Wang, Alon Halevy, and Diyi Yang. 2022. The Moral Integrity Corpus: A Benchmark for Ethical Dialogue Systems. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 3755–3773. https://doi.org/10.18653/v1/2022.acl-long.261