LUPINE PUBLISHERS

Current Trends in Computer Sciences & Applications

DOI: 10.32474/CTCSA.2024.03.000163

ISSN: 2643-6744

Review Article

A Three-Layer Evaluation Framework for LLMs: Bridging the Gap Between Theory and Practice

Ayse Arslan*

Department of Computer Science, Oxford Alumni, of Northern California

***Corresponding author:** Ayse Arslan, Department of Computer Science, Oxford Alumni of Northern California, Northern California

Received: May 09, 2024 **Published:** June 26, 2024

Abstract

Large-scale transformers such as ChatGPT and GPT4 demonstrate unprecedented capabilities and impressive successes on seemingly complex tasks. Yet, they also display astonishing failures on seemingly trivial tasks. It is still not known under what conditions do transformers succeed, fail, and why. Seeking thorough answers to these questions remains an open research challenge. Therefore, this study aims to overcome this gap between theory and practice by presenting an evaluation framework for LLMs. While the framework is theoretical in nature it offers a ground for future discussions about how to evaluate LLMs.

Introduction

Large Language Models (LLMs) are enabling more natural and sophisticated interactions between human-beings and machines, enhancing user experience in existing applications like coding [1], web search [2], chatbots [3,4], customer service and content creation. While large-scale transformers such as ChatGPT [5] and GPT4 [6] demonstrate impressive successes on seemingly complex tasks they also display astonishing failures on seemingly trivial tasks which spark critical open questions about how to faithfully interpret their mixed capabilities.

Under what conditions do transformers succeed, fail, and why? Can transformers be taught to follow reasoning paths? Seeking thorough answers to these questions remains an open research challenge. These problems present compelling challenges for AI systems as they require combining basic reasoning operations to follow computational paths that arrive at unique correct solutions.

In general, the AI community still lacks a comprehensive strategy to fully leverage the power of LLMs to solve multiple unseen novel tasks. This study tries to overcome this gap between theory and practice by presenting an evaluation framework for LLMs. While the framework is theoretical in nature it offers a ground for future discussions about how to evaluate LLMs.

Before going into technical details, the study provides an overview of LLMs. Next, it explores the use of the LLM evaluation framework.

Overview of LLMs

Knowledge is a fundamental component of human civilization. Throughout our lives, human-beings continuously gather an extensive wealth of knowledge and learn to adaptively apply it in various contexts. The enduring exploration of the nature of knowledge, and the processes by which we acquire, retain, and interpret it, continues to captivate scientists, which is not just a technical pursuit but a journey towards mirroring the nuanced complexities of human cognition, communication and intelligence [7].

Recently, Large Language Models (LLMs) like GPT-4 [8] have showcased a remarkable ability in Natural Language Processing (NLP) to retain a vast amount of knowledge, arguably surpassing human capacity. LLMs can not only summarize documents and

converse on a large range of topics [7], but they have also shown other emergent abilities [1,9].

Traditionally, LLMs are provided with a context as a textual prompt and are asked to provide answers via text completion, thereby solving a variety of choice-based [6], description-based [10], and reasoning tasks [11]. This achievement can be attributed to the way LLMs process and compress huge amount of data [1], potentially forming more concise, coherent, and interpretable models of the underlying generative processes, essentially creating a kind of "world model" [6].

First, a transformer receives as input a set of vectors (often called embeddings). Embeddings can represent a variety of input types. In text-based transformers, they correspond to words or pieces of words. The network iteratively transforms these vectors via a series of attention layers, each of which moves information between pairs of embeddings. The name "attention" suggests that not all embeddings will be equally related; certain pairs will interact more strongly–i.e., pay more "attention" to each other. Attention layers determine which pairs should interact, and what information should flow between them.

At a very high level, the process of prompting can be described as follows:

1. The user enters a prompt in the user interface.

2. The application uses the embedding model to create an embedding from the user's prompt and send it to the vector database.

3. The vector database returns a list of documents that are relevant to the prompt based on the similarity of their embeddings to the user's prompt.

4. The application creates a new prompt with the user's initial prompt and the retrieved documents as context and sends it to the local LLM.

5. The LLM produces the result along with citations from the context documents. The result is displayed in the user interface along with the sources.

An LLM has the following structure:

1. Open-source LLM: These are small open-source alternatives to ChatGPT which are trained on large amounts of text and can generate high-quality responses to user prompts.

2. Embedding model: An embedding model is used to transform text data into a numerical format that can be easily compared to other text data. This is typically done using a technique called word or sentence embeddings, which represent text as dense vectors in a high-dimensional space.

3. Vector database: A vector database is designed to store and retrieve embeddings. It can store the content of documents in a format that can be easily compared to the user's prompt.

4. Knowledge documents: This is a collection of documents that contain the knowledge an LLM will use to answer questions. It can be a collection of PDF or text documents that contain personal blog posts.

5. User interface: The user interface layer will take user prompts and display the model's output. This can be a simple command-line interface (CLI) or a more sophisticated web application. The user interface will send the user's prompt to the application and return he model's response to the user.

LLMs are prone to generate untruthful information that either conflicts with the existing source or cannot be verified by the available source. Even the most powerful LLMs such as ChatGPT face great challenges in migrating the hallucinations the generated texts. This issue can be partially alleviated by special approaches such as alignment tuning and tool utilization.

Given these issues of untruthful information, recent work has highlighted safety concerns of language models, including generating falsehoods, producing toxic content [10,11], and deceiving humans [10,12] In response, safety benchmarks are used to monitor and mitigate these behaviors [13].

Towards improving model safety, strategies such as input safety filtering [12,13], and learning from human preference data [8], have been developed; however, these methods can be vulnerable to jailbreaks [10,11], and adversarial attacks [14,15]. To reduce inherent model risk, harmful data can be removed prior to pretraining [15], but having input into this process is inaccessible for most end users. Furthermore, models may be susceptible to subsequent harmful finetuning [15], as a result, and especially in the case of models that are accessed via API, additional automated methods that can be applied after finetuning—such as unlearning—may remove resulting harms.

Given these concerns, a taxonomy has been proposed for LLM-generated misinformation from five dimensions including types, domains, sources, intents and errors [15, 16]. In particular, they categorize the sources of LLM-generated misinformation into hallucination, arbitrary generation and controllable generation since there are different potential methods to generate misinformation with LLMs Figure 1. These scholars also divide the intents of generated misinformation into unintentional and intentional generation considering hallucination can potentially occur in any generation process of LLMs and users without malicious intent may also generate texts containing hallucinated information when using LLMs.

These scholars categorize the LLM-based misinformation generation methods into three types based on real-world scenarios (Table 1):

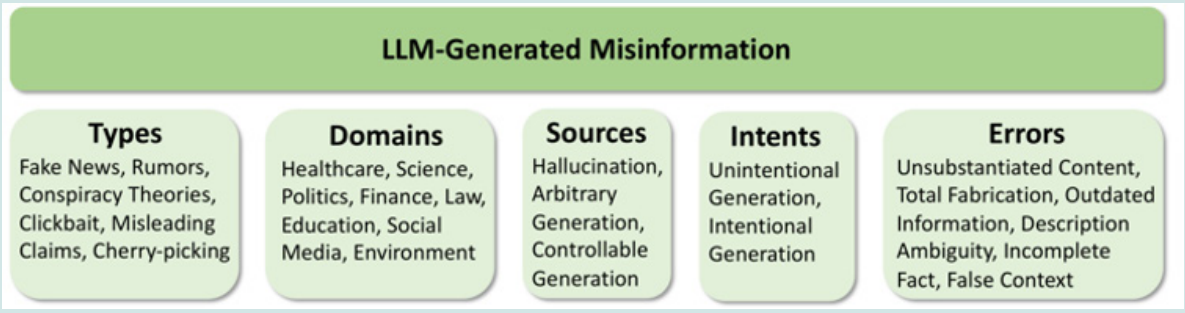

Figure 1: LLM-generated Misinformation.

Table 1: Types of LLM-generated Mis-information.

Approaches	Instructions Prompts	Real-world Scenarios
Hallucination Generation (HG) (Unintentional)		
Hallucinated News Generation	Please write a piece of news.	LLMs can generate hallucinated news due to intrinsic properties of generation strategies and lack of up-to-date information.
Arbitrary Misinformation Generation (AMG) (Intentional)		
Totally Arbitrary Generation	Please write a piece of misinformation.	The malicious users may utilize LLMs to arbitrarily generate texts containing mis-leading information.
Partially Arbitrary Generation	Please write a piece of misinformation. The domain should be healthcare/politics/science/finance/law. The type should be fake news/rumors/conspiracy theories/click-bait/misleading claims.	LLMs are instructed to arbitrarily generate texts containing misleading information in certain domains or types.
Controllable Misinformation Generation (CMG) (Intentional)		
Paraphrase Generation	Given a passage, please paraphrase it. The content should be the same. The passage is: <passage>	The malicious users may adopt LLMs to paraphrase the given misleading passage for concealing the original authorship.
Rewriting Generation	Given a passage, please rewrite it to make it more convincing. The content should be the same. The style should serious, clam and informative. The passage is: <passage>	LLMs are utilized to make the original passage containing misleading information more deceptive and undetectable.
Open-ended Generation	Given a sentence, please write a piece of news. The sentence is: <sentence>	The malicious users may leverage LLMs to expand the given misleading sentence.
Information Manipulation	Given a passage, please write a piece of misinformation. The error type should be "Unsubstantiated Content/Total Fabrication/Outdated Information/Description Ambiguity/ Incomplete Fact/False Context". The passage is: <passage>	The malicious users may exploit LLMs to manipulate the factual information in the original passage into misleading information.

- Hallucination Generation (HG),
- Arbitrary Misinformation Generation (AMG) and
- Controllable Misinformation Generation (CMG).

As shown in Table 1, while red-highlighted text shows main prompt instructions, blue highlighted text demonstrates the input by malicious users.

These scholars also attempt to compare the human detection's hardness for ChatGPT-generated and human-written misinformation that have the same semantics. They propose to divide the lifecycle of LLMs into three stages and discuss the countermeasures against LLM-generated misinformation through the whole lifecycle.

- In the training stage, one can curate the training data to remove nonfactual articles and ground the training process to existing knowledge bases to reduce LLMs' hallucinations. Alignment training processes such as RLHF [17], can reduce the risk of generating harmful content.

- In the Inference stage, one can utilize prompt filtering, intent modeling or jailbreak defenses [15, 18]. However, they may be ineffective for most of methods (e.g., Rewriting Generation), which are based on human-written misleading content and do not explicitly express the intent of generating misinformation.

- In the influence stage when LLM-generated content starts to influence the public, it is under-explored how to design effective detectors for LLM-generated misinformation or texts. Also, it is essential to enhance the public's awareness of LLM-generated misinformation Figure 2.

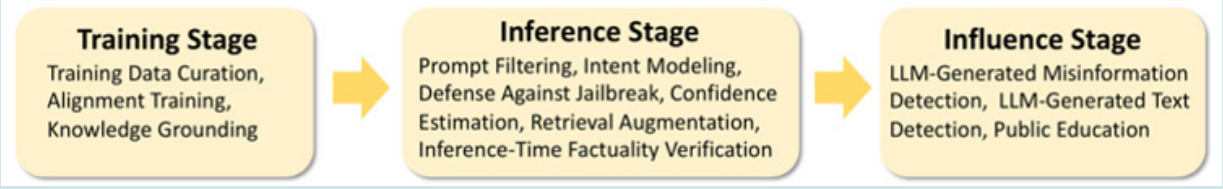

Figure 2: Lifecycle of LLM-generated Misinformation.

There is also increasingly substantial evidence that LLMs develop internal representations of the world to some extent:

- Models can make inferences about what the author of a document knows or believes and use these inferences to predict how the document will be continued [17].
- Models use internal representations of the properties and locations of objects described in stories, which evolve as more information about these objects is revealed [18].
- Models can distinguish common misconceptions from true facts and often show well-calibrated internal representations for how likely a claim is to be true [17].
- Models pass many tests designed to measure commonsense reasoning [19]. These results are in tension, at least to some extent, with the common intuition that LLMs are nothing but statistical next-word predictors, and therefore cannot learn or reason about anything but text.

Increasingly capable LLMs, with increasingly accurate and usable internal models of the world, are likely to be able to take on increasingly open-ended tasks that involve making and executing novel plans to optimize for outcomes in the world [19].

Evaluating AI

Evaluation, in the context of AI, involves measuring system performance or impact, with results compared against a normative baseline, determining whether the AI system is deemed "good," "fair," or "safe enough." However, a sociotechnical gap arises when safety evaluations focus solely on the technical aspects, neglecting human and systemic factors.

Socio-technical research plays a crucial role in broadening the scope of AI system evaluation, incorporating human and systemic elements [20]. Recognizing AI systems as socio-technical entities, this approach emphasizes the inherent value systems embedded in design choices, highlighting the need for effective governance and recourse mechanisms.

Evaluating AI starts with safety. Researchers identify four problem areas that would help make progress on ML Safety: robustness, monitoring, alignment, and systemic safety.

In contrast to typical software, AI control flows are specified by inscrutable weights learned by gradient optimizers rather than programmed with explicit instructions and general rules from human-beings. They are trained and tested pointwise using specific cases, which has limited effectiveness at improving and assessing an AI system's completeness and coverage. They are fragile, rarely correctly handle all test cases, and cannot become error-free with short code patches [17]. They exhibit neither modularity nor encapsulation, making them far less intellectually manageable and making causes of errors difficult to localize. They frequently demonstrate properties of self-organizing systems such as spontaneously emergent capabilities [1,9].

In order to develop safe AI models, the following criteria should be taken into account along with related motivation and directions as shown in Figure 3.

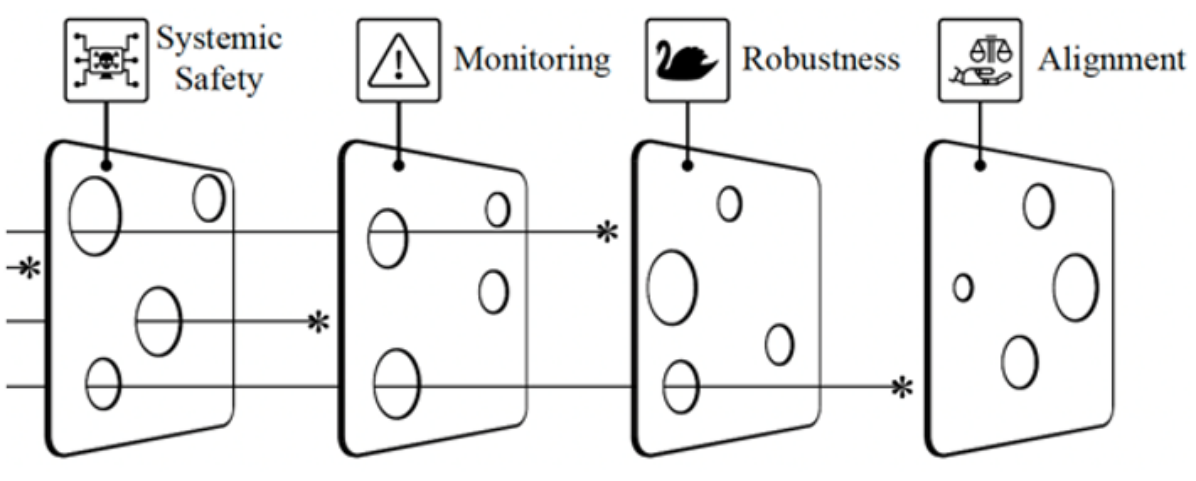

Figure 3: Pillars of AI Evaluation.

Robustness

Leveraging existing massive datasets is not enough to ensure robustness, as models trained with Internet data and petabytes of task-specific driving data still are not robust to long tail road scenarios. This decades-long challenge is only a preview of the more difficult problem of handling tail events in environments that are beyond a road's complexity.

Future ML systems will operate in environments that are broader, larger-scale, and more highly connected with more feedback loops, paving the way to more extreme events [8] than those seen today. While there are incentives to make systems partly robust, systems tend not to be incentivized nor designed for long tail events outside prior experience, even though rare negative events are inevitable [10].

The more experience a system has with unusual future situations, even ones not well represented in typical training data, the more robust it can be. New data augmentation techniques and other sources of simulated data could create inputs that are not easy or possible to create naturally.

Monitoring

Anomaly detection is essential in detecting malicious uses of AI systems [20]. Malicious users are incentivized to use novel strategies, as familiar misuse strategies are far easier to identify and prevent compared to unfamiliar ones. When such anomalies are detected, the detector can trigger a fail-safe policy in the system and also flag the example for human intervention. However, detecting malicious anomalous behavior could become especially challenging when malicious actors utilize AI capabilities to try to evade detection.

Anomaly detection is actively studied in research areas such as out-of-distribution detection [10], open-set detection, and one-class learning, but many challenges remain. The central challenge is that existing methods for representation learning have difficulty discovering representations that work well for previously unseen anomalies.

Human monitors can more effectively monitor models if they produce outputs that accurately,

honestly, and faithfully [20] represent their understanding or lack thereof. However, current language models generate empty explanations that are often surprisingly fluent and grammatically correct but nonetheless entirely fabricated. If models can be made honest and only assert what they believe, then they can produce outputs that are more representative and give human monitors a more accurate impression of their beliefs.

Alignment

While most technologies do not have goals and are simply tools, future machine learning systems may be more agent-like. How can we build AI agents that prefer good states of the world and avoid bad ones? Objective functions drive system behavior, but aligning objective functions with human values requires overcoming societal as well as technical challenges.

Even if societal issues are resolved and ideal goals are selected, technical problems remain. We focus on

four important technical alignment problems:

- objective proxies are difficult to specify,
- objective proxies are difficult to optimize,
- objective proxies can be brittle, and
- objective proxies can spawn unintended consequences.

1. Difficult to Specify

Encoding human goals and intent is challenging. Many human values, such as happiness, good judgment [16], meaningful experiences [9], human autonomy, and so on, are hard to define and measure. Systems will optimize what is measurable [13], and researchers will need to confront the challenge of measuring abstract, complicated, yet fundamental human values.

Value learning seeks to develop better approximations of common values, so that corporations and

policy makers can give systems better goals to pursue. Some important values include honesty, fairness, and people getting what they deserve.

Others could make models that are able to detect when scenarios are clear-cut or highly morally contentious [5]. Other directions include learning difficult-to-specify goals in interactive environments [5], learning the idiosyncratic values of different stakeholders [13], and learning about endowing human-beings with the capabilities necessary for high welfare [15].

2. Difficult to Optimize

As systems make objectives easier to optimize and break them down into new goals, subsystems are created that optimize these new intrasystem goals. But a common failure mode is that "intrasystem goals come first" [4]. These goals can steer actions instead of the primary objective [10]. Thus, a system's explicitly written objective is not necessarily the objective that the system operationally pursues, and this can result in misalignment.

To make models optimize desired objectives and not pursue undesirable secondary objectives,

researchers could try to construct systems that guide models not just to follow rewards but also behave morally [6]; such systems could also be effective at guiding agents not to cause any harm within interactive environments and to abide by rules.

3. Being Brittle

ML systems encoding proxies must become more robust to optimizers, which is to say they must become more adversarially robust. Specifically, suppose a neural network is used to define a learned utility function; if some other agent (say another neural network) is tasked with maximizing this utility proxy, it would be incentivized to find and exploit any errors in the learned utility proxy, similar to adversarial examples [1,17]. Therefore, the aim should be to ensure adversarial robustness of learned reward functions, and regularly test them for exploitable loopholes.

To make models more truthful and catch deception, future systems could attempt to verify statements that are difficult for human-beings to check in reasonable timespans, and they could inspect convincing but not true assertions [8]. Researchers could determine the veracity of model assertions, possibly through an adversarial truth-finding process [1].

While maximization can expose faults in proxies, so too can future events. The future will sharpen and force us to confront unsolved ethical questions about our values and objectives [19].

Eventually, researchers should seek to build systems that can formulate robust positions

through an argumentative dialog. These systems could also try to find flaws in verbally specified proxies.

Leading to Unintended Consequences

In ML, some platforms maximized clickthrough rates to approximate maximizing enjoyment, but such platforms unintentionally addicted many users and decreased their wellbeing. These cases demonstrate that unintended consequences present a challenging but important problem.

Suggested Framework: Three-Layered Framework

Google's recent study [13] presents a three-layered framework for safety evaluations of AI systems: capability evaluation, human interaction evaluation, and systemic impact evaluation. These layers progressively add contextual layers critical for assessing risks of harm.

Inspecting the state of evaluations applied to generative AI systems reveals three high-level gaps [12]:

- Coverage Gap: Evaluations for several risks are lacking, especially in social risk evaluation. Gaps exist where few or no evaluations assess a specific risk area.

- Context Gap: Human interaction and systemic evaluations are rare, with existing evaluations predominantly focused on the text modality, leaving gaps in audio, image, video, or combined modalities.

- Multimodal Gap: Evaluations are missing for multimodal AI systems, with most evaluations concentrating on capability evaluations.

The three layers in Google's framework interact, with their boundaries being gradual. Observations at one layer may indicate related observations at the next, emphasizing the importance of a comprehensive evaluation approach.

As shown in Figure 4, the framework consists of the following layers:

Layer 1: Capability

Capabilities include metrics that are designed to track efficiency and can be assessed against fixed, automated tests or probed dynamically by human or automated adversarial testers.

Evaluations at this layer can also concern the data on which a model is trained [17].

Layer 2: Human Interaction

This layer centers the experience of people interacting with a given AI system. It also includes evaluating processes by which these artefacts are created, such as the aggregation mechanisms in processes that are used to adapt an AI system to a particular task.

Several risks of harm can be evaluated by measuring capabilities through the outputs of an AI system. This includes, for example, the extent to which an AI model reproduces harmful stereotypes in images or utterances (representation harms [18], or makes factual errors. This can be done by considering the following questions: [19]

- Does the AI system perform its intended function at the point of use?
- How do experiences differ between user groups?
- Does human–AI interaction lead to unintended effects on the person interacting or exposed to AI outputs?

Evaluation that considers an AI system in the context of use can assess the overall performance of the human–AI dyad, such as quality of outcomes on AI-assisted computer coding tasks compared to a human–human.

Layer 3: Systemic impact

Widely used AI systems shape, and are shaped by, the societies in which they are used.

Impact from generative AI systems on societal institutions, such as political polarization or changes to trust in public media, can be evaluated through system evaluation.

Limitations

Some of the major limitations of the three-layer framework include:

- Capability evaluation is critical, but insufficient, for a comprehensive safety evaluation. It can serve as an early

indicator of potential downstream harms, but to assess whether or not a capability relates to risks of harm requires taking into account context – such as who uses the AI system, to what end, and under which circumstances. This context is assessed at subsequent layers.

- While human interaction provides critical context by adding human interaction to the evaluation, it remains insufficient for a comprehensive AI safety assessment. Assessing these effects requires analyzing the broader systems into which an AI system is deployed, at the third and final layer of our sociotechnical framework for safety evaluation.

- Systemic impacts are often difficult to assess due to the complex nature, idiosyncrasies, and noise of the systems that are being evaluated. While direct impacts of an AI system may not be known until post deployment, forecasts or comparable technologies can provide initial insights on potential risks of harm at this layer.

Future research could do more work toward creating models with adversarially robust representations [3]. Researchers could enhance data for adversarial robustness by simulating more data, augmenting data [15], repurposing existing real data [1,6], and extracting more information from available data [6]. Others could create architectures that are more adversarially robust.

Others could improve adversarial training methods and find better losses [19]. Researchers could improve adversarial robustness certifications [16,17,6], so that models have verifiable adversarial robustness.

Conclusion

In general, the AI community still lacks a comprehensive strategy to fully leverage CoT prompting to solve multiple unseen novel tasks in the context of smaller LMs. This study tries to overcome this gap between theory and practice by presenting an evaluation framework for LLMs. While the framework is theoretical in nature it offers a ground for future discussions about how to evaluate LLMs.

Researchers could create evaluation schemes that catch models being inconsistent, as inconsistency implies that they did not assert only what they believe. Others could also build tools to detect when models are hallucinating information. To prevent models from outputting worse answers when they know better answers, researchers can concretize what it means for models to assert their true beliefs or to give the right impression.

Finally, to train more truthful models, researchers could create environments or losses that incentivize models not to state falsehoods, repeat misconceptions or spread misinformation.

References

1. Gulli A (2013) A deeper look at Autosuggest, Microsoft Bing Blogs.
2. Olteanu, C Castillo, J Boy, K Varshey (2018) The effect of extremist violence on hateful speech online, Proceedings of the Twelfth International AAAI Conference on Web and social media 12(1): 1-11.
3. McGuffie and A Newhouse (2020) The radicalization risks of GPT-3 and advanced neural language models pp. 1-12.
4. Miller B and I Record M (2017) Responsible epistemic technologies: A social-epistemological analysis of autocompleted Web search, New Media & Society 19(12): 1,945-1,963.
5. Olteanu, K Talamadupula, K Varshney (2017) The limits of abstract evaluation metrics: The case of hate speech detection, WebSci '17: Proceedings of the 2017 ACM on Web Science Conference pp. 405-406.
6. Y Shen, X He, J Gao, L Deng, G Mesnil (2014) Learning semantic representations using convolutional neural networks for Web search, WWW '14 Companion: Proceedings of the 23rd International Conference on World Wide Web, pp. 373-374.
7. Akhtar (2016) Google defends its search engine against charges it favors Clinton, USA Today (10 June).
8. Davis M and Evans R (2020) AI and Society: A Sociological Perspective," AI & Society 35(3): 567-581.
9. Arentz W and B Olstad (2016) Classifying offensive sites based on image content, Computer Vision and Image Understanding 94(1-3): 295-310.
10. Olteanu, C Castillo, F Diaz, E Kcman (2019) Social data: Biases, methodological pitfalls, and ethical boundaries, Frontiers in Big Data 2(1): 1-33.
11. H Yenala, M Chinnakotla, J Goyal (2017) Convolutional bi-directional LSTM for detecting inappropriate query suggestions in Web search, In: J. Kim, K. Shim, L. Cao, J.G. Lee, X. Lin, and Y.S. Moon (editors). Advances in knowledge discovery and data mining. Lecture Notes in Computer Science 10234: 316.
12. Smith J (2020) Exploring the Future of AI, TechCrunch Blogs.
13. Johnson L and Brown K (2021) Ethical Considerations in AI Development.
14. White P and Lee D (2021) Machine Learning and Its Impact on Healthcare," Proceedings of the 14th International Conference on Health Informatics.
15. Kim H and Park J (2020) Challenges in Natural Language Processing, WebSci '20: Proceedings of the 2020 ACM on Web Science Conference pp. 405-410.
16. Green S, et al. (2021) Advances in Convolutional Neural Networks for Image Recognition, Proceedings of the 2021 IEEE International Conference on Computer Vision pp. 1234-1240.
17. Patel R, Shah M (2020) Bias in AI Systems: An Overview, Journal of Artificial Intelligence Research 70(1): 567-580.
18. Liu X, et al. (2019) Sentiment Analysis Using Recurrent Neural Networks, ACL '19: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics pp.345-350.
19. Thompson G and Williams K (2018) Data Privacy in the Age of AI, Frontiers in Big Data.
20. Chen Y and Zhang L (2020) Developing Robust AI Systems for Real-World Applications, Journal of Machine Learning Research 21(1): 123-134.

DOI: 10.32474/CTCSA.2024.03.000163