Chatbot Evaluation Is (Sometimes) Ill-Posed: Contextualization Errors in the Human-Interface-Model Pipeline

Aspen Hopkins*
MIT CSAIL
Cambridge, MA, USA
dataspen@mit.edu

Angie Boggust*
MIT CSAIL
Cambridge, MA, USA
aboggust@csail.mit.edu

Harini Suresh*
Brown University
Providence, RI, USA
harini_suresh@brown.edu

Abstract

Large language models (LLMs) — and in particular, their user-facing manifestation as chatbots — are known to produce erroneous outputs, often misinterpreting user intent and generating incorrect or problematic responses. These errors have led to calls for more comprehensive, human-centered, and context-sensitive evaluation mechanisms. To inform such evaluation mechanisms, we argue that these failures are not solely the result of "hallucinations," but also of *contextualization errors* — systematic mistakes that arise from the interaction between users, models, and interfaces. Contextualization errors are not just a product of a model's limitations in isolation, but stem from failures in the user-model *interaction* process. To mitigate these issues, it is necessary to evaluate not only the correctness of the model's responses, but also the specificity of user inputs and the affordances of the interactive interfaces that mediate human-model communication. We introduce a framework for understanding the sources of contextualization errors, identifying three primary sources: (1) *semantic underspecification*, where user prompts contain inherent ambiguity; (2) *missing information*, where key contextual details are absent despite an explicit prompt; and (3) *insufficient user constraints*, where users do not properly direct or constrain the expected set of outputs. Additionally, we explore (4) *output ambiguity*, where factors such as non-deterministic behavior, token truncation, and safety mechanisms introduce inconsistencies or collapse distinct prompts into identical responses. Finally, we propose directions for LLM evaluation that assess the entire user-model interaction process. We argue that to improve LLM-based chatbot reliability, evaluations must extend beyond correctness to scrutinize the specificity of user prompts, the expressiveness of model outputs, and the impact of interface design. By addressing these fundamental issues, we can develop more robust methodologies for mitigating contextualization errors and improving chatbot usability across diverse applications.

CCS Concepts

• **Computing methodologies** → **Natural language generation**;
• **Human-centered computing** → **Interaction paradigms**.

*All authors contributed equally to this research.

HEAL@CHI'25, Yokohama, Japan

ACM Reference Format:
Aspen Hopkins, Angie Boggust, and Harini Suresh. 2025. Chatbot Evaluation Is (Sometimes) Ill-Posed: Contextualization Errors in the Human-Interface-Model Pipeline. In *Proceedings of Human-Centered Evaluation and Auditing of Language Models Workshop at CHI 2025 (HEAL@CHI'25).* ACM, New York, NY, USA, 6 pages.

1 Introduction

Large language models (LLMs) have garnered wide-spread criticism from both academics and main-stream media for their tendency to generate erroneous or harmful outputs. In particular, hallucinations — generations that include factually incorrect or materially unfaithful content — have garnered much attention [18, 46]. However, as LLMs have been adapted to a variety of settings as chatbots, including traditional chatbot interfaces (e.g., ChatGPT [33]) and integrated chatbots (e.g., Cursor [8] or command-line APIs), other types of errors have emerged.

In this paper, we focus on *contextualization errors* — errors that occur when a chatbot incorrectly assumes a user's intent and thus does not appropriately respond to their prompt. The chatbot response might be truthful (i.e., not a hallucination), but not applicable to the user's interest; or, it may adopt a tone, framing, or position that is undesirable to the prompter. For example, the capabilities of Google Gemini were recently called into question when it responded to a request to "produce a portrait of America's founding fathers" with a racially diverse group of men [1]. This response was not necessarily "incorrect" — had the user been interested in an imaginative retelling of America's founding, for instance, perhaps the image would have been a well-suited response. Of course, if the user's goal was to generate a historically accurate rendering of the founding fathers — which much of the surrounding discourse assumed was the case — the resulting response was inappropriate. In other words, because the prompt was *underspecified*, missing key context about the users goals and intent, the system likely "filled in" the missing context based on available priors or hard-coded prompt injections.

As humans, we frequently resolve this type of ambiguity in human-to-human interactions by relying on multi-modal contextual clues, drawing on a larger body of knowledge, or simply asking for clarification. Chatbots struggle with contextualization, however, because they have limited access to a user's intention and context, primarily relying on the information present in the input prompt.

Intending to mitigate this issue and enable personalization, recent advances in chatbots have included memory units [34, 43], allowing them to personalize responses based on a user's prior interactions. However, memory is not a fail-safe, or necessarily an ideal solution [3, 7, 20]. Chatbot memory as it is currently implemented may reduce users' autonomy. Users are not able to decide

what contextual information to provide, or scope what is applicable across interactions, and may struggle to opt out of any data collection that ensues. Context that a user has previously provided may no longer be relevant the next day, or even the next hour; thus, while some context may be appropriate for one interaction, it may be misleading in another.

Thus far, LLM errors have primarily resulted in increased calls for evaluations ensuring models generations are factually correct responses, reducing the frequency of hallucinations [32, 40]. Because contextualization errors *may not be* factually incorrect (but rather the result of misalignment with the user's goals), they are not guaranteed to be addressed by this agenda. Our work characterizes contextualization errors and proposes evaluation approaches to improve their identification and correction.

In particular, we posit that instead of primarily evaluating an LLM's independent capabilities, we must *evaluate the entire human-model interaction process*. First, as prior research has discussed [29, 44, 47], we must evaluate a model's ability to reason about an underspecified prompt, as some models or training regimes are more likely to contextualize in a way that is consistent with a particular user or user population. To better understand how and why contextualization errors are produced, we must also evaluate the quality of a user's input prompt to understand which prompts produce the most effective outputs and what types of underspecification are the hardest for models to resolve. Finally, human-model interactions are often facilitated through an interface, such as a coding-environment direct API call (e.g., API access to OpenAI's models), an interactive chatbot interface (e.g., ChatGPT [33] or Claude [2]), or an LLM-integrated environment (e.g., Cursor for code [8], CoCounsel for legal work [42], or Spellbook AI for contracts [39]). Therefore, it is necessary to evaluate the ways in which human-model interaction unfolds *within a specific interface*.

We first summarize related work on chatbot evaluation and prompt underspecification (Section 2). Then, we characterize contextualization errors, demonstrating how they often stem from distinct kinds of underspecification in a user's prompt (Section 3). Finally, we propose future work extending LLM evaluations from specific tests of model capabilities to comprehensive evaluations of the entire human-model generation process (Section 4).

2 Related Works

In the following section, we briefly outline the importance and success of chatbot interfaces and then contrast hallucinations with contextual or semantic underspecification in prompts.

2.1 Growth of Chatbots

When GPT-1 was first introduced, it received relatively little mainstream attention. This changed with ChatGPT's introduction in late 2022. The chatbot, which paired an LLM with a well-designed conversational interface, inspired a watershed of wide-spread adoption, outpacing the growth of other groundbreaking technologies, including social media [21]. Since then, competitor products (e.g., Claude [2]) have entered the market, while chatbots tailored for specific settings — including medicine, law, travel services, programming, finance, therapy, sales, and customer service, among many others — have proliferated [8, 39, 42]. Notably, interfaces that adopted convenient, intuitive interactions with users—abstracting away complexity for lay-users — were necessary for growing a non-expert user base. As datasets for chatbot tuning become more common [13, 37], interfaces remain a major differentiator in adoption and user experience.

2.2 Evaluating Chatbots

Assessing the performance of chatbots requires specialized benchmarks that go beyond traditional text generation metrics. While standard measures, like ROUGE [23], evaluate text similarity, chatbot evaluation demands more nuanced benchmarks that assess accuracy, contextual understanding, and response usefulness [15, 24, 30, 32, 40]. For example, Banerjee et al. [4] introduced an E2E (End-to-End) benchmark specifically targeting chatbot performance.

In addition to general chatbot performance, evaluating contextualization errors and user prompt interpretations remains an active area of research. Various domain-specific benchmarks, such as those for medical Q&A [19], mathematical reasoning [16], and legal assistance [5], help assess chatbot reliability in specialized settings. Recent research explores new methodologies for improving chatbot fidelity, particularly in minimizing hallucinations and enhancing response alignment with user intent [14, 15, 24, 32]. Other works seek to benchmark chatbot dialogue [30]. Comprehensive, standardized approaches for chatbot benchmarking are still evolving, and our work aims to bring further clarity to the kinds of errors this work should attend to.

2.3 Hallucinations

Hallucinations are typically defined by one of two axes: faithfulness, which ensures that generated content remains truthful to its source, and factuality, which relates to whether the information aligns with real-world facts [18]. As factuality is often the result of normative agreement, its definition may focus not just on aligning model behavior with static truths but also on recognizing the role of evolving consensus.

Methods for detecting or responding to hallucinations vary. One recent approach utilizes entropy-based uncertainty estimation to identify hallucinations, measuring the model's confidence in its outputs to detect deviations from factual accuracy [11]. Others train classifiers to distinguish fact from fiction [9], while adjacent work has sought to standardize the evaluation of hallucination detection methods through benchmarks designed for assessing faithfulness in chatbot-generated dialogues [26]. While hallucinations reflect an LLM's tendency to generate factually incorrect or unfaithful content beyond provided information — whether due to model architecture, training dynamics, or inherent generative properties — contextualization errors stem from the user-model *interaction* process, where ambiguity in prompting or in the system's implementation leads to irrelevant or unwanted responses.

2.4 Semantic Underspecification in LLMs

Semantic underspecification refers to instances in language where expressions lack complete information, necessitating additional context for full interpretation. This phenomenon is intrinsic to human communication, allowing for efficiency and flexibility. For example, the pronoun "they" can refer to individual(s) of unspecified gender

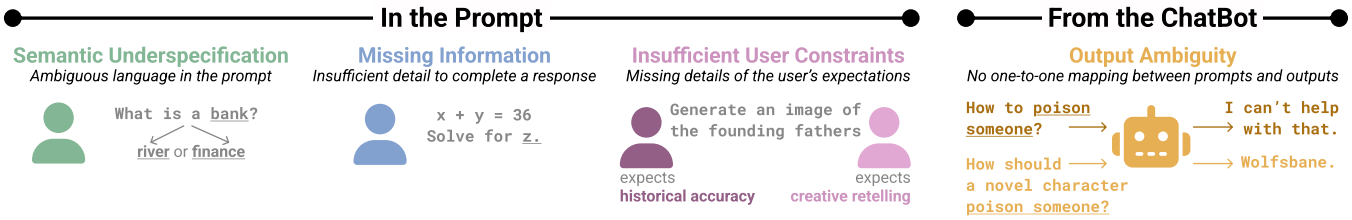

Figure 1: Contextualization errors occur through prompt underspecification or modeling decisions. Sources of prompt-based contextualization errors include semantic underspecification, missing information, and insufficient user constraints. Model-based sources of contextualization errors include output ambiguity, where different prompts may result in the same response despite being distinct (non-injectivity), or the same prompts may result in different outputs (non-determinism).

or number, requiring contextual clues for precise understanding. Some emphasize that such underspecification is not a flaw but a *feature* that enables nuanced and adaptable communication [12].

In linguistics and natural language processing (NLP), addressing semantic underspecification has been a longstanding challenge. Early approaches involved creating formal, symbolic representations that captured potential meanings without generating them explicitly. Poesio [36] and Niehren et al. [31] developed frameworks to model underspecified semantics, enabling systems to handle ambiguity by representing multiple interpretations simultaneously.

With the advent of LLMs, the handling of semantic underspecification has gained renewed attention. Wildenburg et al. [44] introduced the Dataset of Semantically Underspecified Sentences grouped by type (DUST) to evaluate LLMs' capabilities in identifying and interpreting underspecified sentences. Pezzelle [35] and Testoni et al. [41] explore this issue in multimodal NLP systems, which integrate language with other modalities like vision. Their works highlight that even when grounding language with additional contexts, these systems often struggle with semantic underspecification. Such shortcomings can negatively impact performance and lead to unintended consequences in applications relying on precise language understanding.

3 Contextualization Errors

Contextualization errors can have many causes. In this section, we compile four sources of contextualization errors that may ultimately lead to uncertain outputs or ambiguous model interpretation. The first three, discussed in Section 3.1, arise from different forms of underspecification in a user's prompt, including (1) **semantic underspecification**, (2) **missing information**, and (3) **insufficient user constraints**. A final source for error arises from the model itself—(4) **output ambiguity**, discussed in Section 3.2, in which different prompts may result in the same response despite being distinct (non-injectivity), or the same prompts may result in different outputs (non-determinism).

3.1 Prompt underspecification

One way contextualization errors can occur is via prompts with **semantic underspecification**. As discussed in Section 2, semantic underspecification suggests there is ambiguity in the language itself — for example, a user might be referring to a river bank *or* a financial bank when they ask "what is a bank?" [22].

This is contrast to **missing information**, where, regardless of the context, there is insufficient detail to complete a response but the prompt itself is explicit. For example, a user may prompt the model with a math question but leave out a variable, or, when interacting with a customer service bot, may express frustration about an experience and request a refund without communicating specific details.

Insufficient user constraints, while similar to both semantic underspecification and missing information, is distinct. When users do not explicitly share details of their motivation, the model — which is not aware of the user's context in the way that another person might be — lacks necessary constraints on the possible set of outputs. For example, two users might input the same chatbot prompt asking for an image of America's founding fathers. A user studying for a history test will expect a historically-accurate rendering. While another, sourcing inspiration for a creative retelling of the nation's founding, would be excited to see historical figures reinterpreted as multi-racial, much like Lin-Manuel Miranda's *Hamilton*. There is not a universally *right* answer in this case, nor is there a particularly ambiguous word or phrase. Instead, the responses are simply misaligned with users' implicit desires and expectations. This expands to other scenarios where users have varying expectations about how the model should behave. For example, when inducing a data distribution, some users might want a model to sample a random distribution iteratively in an independent fashion or in an auto-regressive fashion [17]. Or, when prompting a model to generate an image "of a CEO", different users have different expectations for the CEO's background, clothing, race, and gender [38].

3.2 Output Ambiguity

The sources of contextualization errors described in the prior section are primarily due to variance in user behavior. However, contextualization errors can also arise from implementation details of the model, the interface, the tokenizer, or other aspects of safety mechanism and design decisions, which lead to constraints on expressiveness, unintended biases, token truncation, or oversimplification of complex queries. A user might simply be unable to achieve the output they expect, even if they adjust their prompt, given the limitations of the chatbot they are interacting with. We characterize the source of these errors as **output ambiguity**.

Often, this is the result non-deterministic or non-injective behavior introduced in the chatbot implementation. Rather than mapping one prompt to exactly one outcome, there are potentially many outcomes, or different queries may result in the same outcome. Consider the following scenario: two users query a customer service bot for information about refunds. One user is unhappy, but due to cultural norms is polite with the chatbot while asking for a refund. Another is mostly satisfied with their experience but is checking on how refund processes work. Whereas the first user interaction should likely be escalated to a customer service agent, the ultimate outcome for both users is an question-answer loop interaction with the chatbot.

These ambiguous outcomes may arise at numerous points in the human-model-interface pipeline. For example, the underlying model may overgeneralize: during sentiment analysis a model could assign the same positive score to two slightly different interactions, the training data may not be sufficient to distinguish between the two users' queries, or the tokenization may cause semantically distinct inputs to be mapped to similar or identical outputs, reducing specificity. Interface constraints (such as max token limits) can cause truncation or omission of key information. It may even be the result of a softmax function applied to the model's final layer. While many AI models use a softmax function to convert logits into probabilities, different input conditions can produce logits that, after softmax, *yield the same probability distribution*. Although internal representations might differ, the final output probabilities remain indistinguishable for practical purposes.

Similarly, safety filters and alignment mechanisms may censor or modify responses in a way that removes necessary nuance or detail. When safety filters are applied, certain probabilities may be reweighted or suppressed, leading to non-injectivity by forcing different queries into the same response category, even when they may warrant different answers. For example, if the chatbot is asked to refund a sex toy or a BB gun, it may return a generic response as the probabilities of restricted outputs are zeroed out, making different inputs converge to the same "safe" output. The safety mechanisms may even lead to an incorrect answer, depending on their implementation. And of course, sampling randomness and temperature settings for outputs can lead to varied outputs for the same input, making consistency difficult.

4 Designing End-to-end Evaluations of Chatbots in Context

We propose that evaluations should study LLM-based chatbots' propensity for contextualization errors across the human-interface-model pipeline. We propose some promising directions in evaluating **LLM contextualization** (Section 4.1), **user specificity** (Section 4.2), and **interfaces** (Section 4.3).

4.1 Evaluating LLM Contextualization

Contextualization errors are a function of context. That is, contextualization errors occur when the lack of specificity leads to differing assumptions between the user and the model. If the model and user handled ambiguity in the same way, then there is no contextualization error. As such, there may be cases where a model's assumptions align better with one subset of users than another [38]. Evaluations could compare model and human contextualization strategies and stratify by population to understand if there is general alignment across all models or if some models are better aligned to particular users or human values.

Other work could investigate strategies for prompting users to resolve ambiguity. Recent publications on uncertainty estimation in LLMs has focused on confidence calibration given a model's logits [25, 27, 45]. A similar method might be built into chatbot interactions — e.g., if the logits suggest some threshold of uncertainty has been reached, the chatbot should query for additional information. This approach also gives the user more agency in the interaction, allowing them to specify their intent and expectations, rather than the system inferring the "most likely" option. Chatbot developers may worry that adding in additional interaction (i.e., to clarify the user's intent) will put off users due to the additional interaction; but we argue that it is not necessarily this straightforward. Users may experience more frustration in the setup where they are *not* prompted for clarification, and have to attempt to iteratively understand the model's behavior and adjust their prompt to get the response they want. To this end, future work might empirically evaluate, if and how a back-and-forth to resolve ambiguity affects user agency, satisfaction, frustration, or other aspects of the experience.

These directions could utilize our distinction of different sources of contextualization errors (Section 3). For instance, we could imagine evaluating how well different chatbots handle prompts with missing information versus prompts with semantic underspecification. Understanding these different sources could also be helpful for considering and evaluating mitigation strategies — e.g., a back-and-forth interaction to resolve semantic underspecification might look different than one intended to resolve insufficient user constraints.

In general, results from such evaluations contribute to our understanding of model behavior in the face of ambiguity. For instance, prior work studying how LLMs handle homonyms has led to a better overall understanding of model behavior [28]; similarly, broadening evaluations to understand contextualization errors may also provide additional signal supporting interpretability or explainability efforts.

4.2 Evaluating User Specificity

Issues that arise from underspecified prompts (e.g., semantic underspecification, missing information, or insufficient user constraints, as described in Section 3.1) are often challenging to identify *a priori* by the prompter, to whom the intent and context of the prompt is self-evident. In the founding fathers example, for instance, the generated image would be surprising to a user expecting a historically accurate rendering because, to that user, they may solely envision their interpretation of the prompt. It is only after seeing the resulting image that it becomes clear how others might interpret the prompt (i.e., what aspect of the prompt is insufficiently contextualized). This begs a question of how systems should support users in identifying prompt underspecification *before* it leads to contextualization errors.

One immediate strategy for evaluation is to quantify the specificity of a user's prompt. For instance, building on prior work [6, 10], specificity scoring systems could combine the rule-based linguistic

understanding of low-level ambiguity (e.g., homonyms and typos) with LLM agents that identify high-level missing context (e.g., user tasks and goals). These systems may then provide an overall specificity score, highlighting parts of the prompt that are ambiguous, and suggesting where missing context might be added. Specificity scoring models could then be integrated into user workflows, helping users improve individual prompts and encouraging prompt writing habits that better serve user goals in the long term. Similarly, in API settings, users might pass a prompt to the scoring model, allowing them to iterate before deploying the prompt to the LLM. In chatbot settings, the interface can then directly incorporate scoring models into user prompts in real time, suggesting auto-completions that increase the specificity score.

Of course, these approaches must contend with the fact that updates to the underlying model might render learned prompting strategies less useful. Therefore, we envision that a comprehensive approach to contextualization errors should not rely solely on changing prompting behavior, but be equally invested in evaluating and mitigating these issues through model behavior and interfaces.

4.3 Evaluating Interfaces

In addition to understanding the strengths and limitations of model contextualization, and supporting users in creating more well-defined prompts, future evaluations should study how chatbot interfaces affect contextualization errors. How do the design decisions of a chatbot interface affect how users interpret and react to these errors? For instance, some chatbots present multiple possible responses within an interface; others always respond with just one. Does the former approach help users understand areas of ambiguity and shape their future behavior? Other design decisions, such as UI mechanisms to provide feedback, display uncertainty, or provide context on the agent's role, could also be studied. Evaluations could study how human interactions with chatbots change when they face a contextualization error, as compared to other model errors, such as hallucinations. In general, we emphasize a need to study human interaction with chatbots not only in an abstract sense, but also *within the specific interfaces* in which they occur.

4.4 Conclusion

This paper highlights the emerging issue of contextualization errors in LLM-based chatbots, which occur when chatbots misinterpret user intent or fail to align with the user's goals. Unlike hallucinations, these errors do not stem from factual inaccuracies but rather from missing context, leading to responses that may meet some notion of truthfulness but be irrelevant or undesired. We first characterize distinct sources of contextualization errors: (1) semantic underspecification, (2) missing information, (3) insufficient user constraints, and (4) output ambiguity. Then, we argue that addressing contextualization errors requires a shift in evaluation approaches — from focusing solely on the correctness of model outputs to considering the entire human-interface-model interaction process. This includes evaluating how different models and interaction strategies address contextualization errors, evaluating the clarity and specificity of user prompts, and assessing how interfaces contribute to or mitigate these issues.

References

[1] Bobby Allyn. 2024. Google races to find a solution after AI generator Gemini misses the mark. *NPR* (2024). https://www.npr.org/2024/03/18/1239107313/google-races-to-find-a-solution-after-ai-generator-gemini-misses-the-mark

[2] Anthropic. 2024. The Claude 3 Model Family: Opus, Sonnet, Haiku. https://assets.anthropic.com/m/61e7d27f8c8f5919/original/Claude-3-Model-Card.pdf

[3] Sanghwan Bae, Dong-Hyun Kwak, Soyoung Kang, Min Young Lee, Sungdong Kim, Yuin Jeong, Hyeri Kim, Sang-Woo Lee, Woo-Myoung Park, and Nako Sung. 2022. Keep Me Updated! Memory Management in Long-term Conversations. In *Findings of the Association for Computational Linguistics: EMNLP*. Association for Computational Linguistics, 3769–3787. doi:10.18653/V1/2022.FINDINGS-EMNLP.276

[4] Debarag Banerjee, Pooja Singh, Arjun Avadhanam, and Saksham Srivastava. 2023. Benchmarking LLM powered Chatbots: Methods and Metrics. arXiv:2308.04624 [cs.CL] https://arxiv.org/abs/2308.04624

[5] Ilias Chalkidis, Abhik Jana, Dirk Hartung, Michael J. Bommarito II, Ion Androutsopoulos, Daniel Martin Katz, and Nikolaos Aletras. 2022. LexGLUE: A Benchmark Dataset for Legal Language Understanding in English. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*. ACL, 4310–4330. doi:10.18653/V1/2022.ACL-LONG.297

[6] Claudia Collacciani, Giulia Rambelli, and Marianna Bolognesi. 2024. Quantifying generalizations: Exploring the divide between human and llms' sensitivity to quantification. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*. 11811–11822.

[7] Samuel Rhys Cox, Yi-Chieh Lee, and Wei Tsang Ooi. 2023. Comparing How a Chatbot References User Utterances from Previous Chatting Sessions: An Investigation of Users' Privacy Concerns and Perceptions. In *International Conference on Human-Agent Interaction (HAI)*. ACM, 105–114. doi:10.1145/3623809.3623875

[8] Cursor. [n. d.]. Cursor; The AI Code Editor. https://www.cursor.com/.

[9] Xuefeng Du, Chaowei Xiao, and Sharon Li. 2024. HaloScope: Harnessing Unlabeled LLM Generations for Hallucination Detection. In *Advances in Neural Information Processing Systems*, Vol. 37. Curran Associates, Inc., 102948–102972.

[10] Federico Errica, Giuseppe Siracusano, Davide Sanvito, and Roberto Bifulco. 2025. What Did I Do Wrong? Quantifying LLMs' Sensitivity and Consistency to Prompt Engineering. arXiv:2406.12334 [cs.LG] https://arxiv.org/abs/2406.12334

[11] Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. 2024. Detecting hallucinations in large language models using semantic entropy. *Nature* 630, 8017 (2024), 625–630.

[12] Steven Frisson. 2009. Semantic underspecification in language processing. *Language and linguistics compass* 3, 1 (2009), 111–127.

[13] Jingsheng Gao, Yixin Lian, Ziyi Zhou, Yuzhuo Fu, and Baoyuan Wang. 2023. LiveChat: A Large-Scale Personalized Dialogue Dataset Automatically Constructed from Live Streaming. arXiv:2306.08401 [cs.CL] https://arxiv.org/abs/2306.08401

[14] Dan Hendrycks, Collin Burns, Steven Basart, Andrew Critch, Jerry Li, Dawn Song, and Jacob Steinhardt. 2021. Aligning AI With Shared Human Values. *Proceedings of the International Conference on Learning Representations (ICLR)* (2021).

[15] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring Massive Multitask Language Understanding. *Proceedings of the International Conference on Learning Representations (ICLR)* (2021).

[16] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring Mathematical Problem Solving With the MATH Dataset. *NeurIPS* (2021).

[17] Aspen K Hopkins and Alex Renda. 2023. Can llms generate random numbers? evaluating llm sampling in controlled domains. Sampling and Optimization in Discrete Space (SODS) ICML 2023 Workshop.

[18] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of Hallucination in Natural Language Generation. *ACM Comput. Surv.* 55, 12, Article 248 (March 2023), 38 pages. doi:10.1145/3571730

[19] Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2021. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences* 11, 14 (2021), 6421.

[20] Eunkyung Jo, Yuin Jeong, SoHyun Park, Daniel A. Epstein, and Young-Ho Kim. 2024. Understanding the Impact of Long-Term Memory on Self-Disclosure with Large Language Model-Driven Chatbots for Public Health Intervention. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. ACM, 440:1–440:21. doi:10.1145/3613904.3642420

[21] Dinesh Kalla, Nathan Smith, Fnu Samaah, and Sivaraju Kuraku. 2023. Study and analysis of chat GPT and its impact on different fields of study. *International journal of innovative science and research technology* 8, 3 (2023).

[22] Michael A. Lepori, Michael Mozer, and Asma Ghandeharioun. 2024. Racing Thoughts: Explaining Large Language Model Contextualization Errors. *CoRR* abs/2410.02102 (2024). doi:10.48550/ARXIV.2410.02102 arXiv:2410.02102

[23] Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*. Association for Computational Linguistics, 74–81.
[24] Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring How Models Mimic Human Falsehoods. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 3214–3252. doi:10.18653/V1/2022.ACL-LONG.229
[25] Linyu Liu, Yu Pan, Xiaocheng Li, and Guanting Chen. 2024. Uncertainty estimation and quantification for llms: A simple supervised approach. *arXiv preprint arXiv:2404.15993* (2024).
[26] Wen Luo, Tianshu Shen, Wei Li, Guangyue Peng, Richeng Xuan, Houfeng Wang, and Xi Yang. 2024. Halludial: A large-scale benchmark for automatic dialogue-level hallucination evaluation. *arXiv preprint arXiv:2406.07070* (2024).
[27] Huan Ma, Jingdong Chen, Guangyu Wang, and Changqing Zhang. 2025. Estimating LLM Uncertainty with Logits. *arXiv preprint arXiv:2502.00290* (2025).
[28] Huu Tan Mai, Cuong Xuan Chu, and Heiko Paulheim. 2024. Do LLMs really adapt to domains? An ontology learning perspective. In *International Semantic Web Conference*. Springer, 126–143.
[29] Emily McMilin. 2024. Underspecification in Language Modeling Tasks: A Causality-Informed Study of Gendered Pronoun Resolution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 18778–18788.
[30] John Mendonça, Alon Lavie, and Isabel Trancoso. 2024. On the Benchmarking of LLMs for Open-Domain Dialogue Evaluation. *arXiv preprint arXiv:2407.03841* (2024).
[31] Joachim Niehren, Manfred Pinkal, and Peter Ruhrberg. 1997. A uniform approach to underspecification and parallelism. In *35th Annual Meeting of the Association of Computational Linguistics*. 410–417.
[32] Jio Oh, Soyeon Kim, Junseok Seo, Jindong Wang, Ruochen Xu, Xing Xie, and Steven Whang. 2024. ERBench: An Entity-Relationship based Automatically Verifiable Hallucination Benchmark for Large Language Models. In *Advances in Neural Information Processing Systems*.
[33] OpenAI. 2022. Introducing ChatGPT. https://openai.com/index/chatgpt/.
[34] OpenAI. 2024. Memory and new controls for ChatGPT. https://openai.com/index/memory-and-new-controls-for-chatgpt/
[35] Sandro Pezzelle. 2023. Dealing with semantic underspecification in multimodal NLP. *arXiv preprint arXiv:2306.05240* (2023).
[36] Massimo Poesio. 1994. Ambiguity, underspecification and discourse interpretation. In *Proceedings of the First International Workshop on Computational Semantics*. Citeseer, 151–160.
[37] Hongjin Qian, Xiaohe Li, Hanxun Zhong, Yu Guo, Yueyuan Ma, Yutao Zhu, Zhanliang Liu, Zhicheng Dou, and Ji-Rong Wen. 2021. Pchatbot: A Large-Scale Dataset for Personalized Chatbot. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, New York, NY, USA, 2470–2477. doi:10.1145/3404835.3463239
[38] Zoe De Simone, Angie Boggust, Arvind Satyanarayan, and Ashia Wilson. 2024. DiffusionWorldViewer: Exposing and Broadening the Worldview Reflected by Generative Text-to-Image Models. arXiv:2309.09944 [cs.LG] https://arxiv.org/abs/2309.09944
[39] Spellbook. [n. d.]. Spellbook. https://www.spellbook.legal/
[40] Gaurang Sriramanan, Siddhant Bharti, Vinu Sankar Sadasivan, Shoumik Saha, Priyatham Kattakinda, and Soheil Feizi. 2024. LLM-Check: Investigating Detection of Hallucinations in Large Language Models. In *Advances in Neural Information Processing Systems*.
[41] Alberto Testoni, Barbara Plank, and Raquel Fernández. 2024. RACQUET: Unveiling the Dangers of Overlooked Referential Ambiguity in Visual LLMs. *arXiv preprint arXiv:2412.13835* (2024).
[42] Thomson Reuters. [n. d.]. The new way to work: CoCounsel, the GenAI assistant for professionals. https://www.thomsonreuters.com/en/insights/articles/the-new-way-to-work-cocounsel-the-genai-assistant-for-professionals
[43] Weizhi Wang, Li Dong, Hao Cheng, Xiaodong Liu, Xifeng Yan, Jianfeng Gao, and Furu Wei. 2023. Augmenting Language Models with Long-Term Memory. In *Advances in Neural Information Processing Systems*.
[44] Frank Wildenburg, Michael Hanna, and Sandro Pezzelle. 2024. Do Pre-Trained Language Models Detect and Understand Semantic Underspecification? Ask the DUST! *arXiv preprint arXiv:2402.12486* (2024).
[45] Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. 2023. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. *arXiv preprint arXiv:2306.13063* (2023).
[46] Ziwei Xu, Sanjay Jain, and Mohan Kankanhalli. 2024. Hallucination is inevitable: An innate limitation of large language models. *arXiv preprint arXiv:2401.11817* (2024).
[47] Ze Yu Zhang, Arun Verma, Finale Doshi-Velez, and Bryan Kian Hsiang Low. 2024. Understanding the relationship between prompts and response uncertainty in large language models. *arXiv preprint arXiv:2407.14845* (2024).